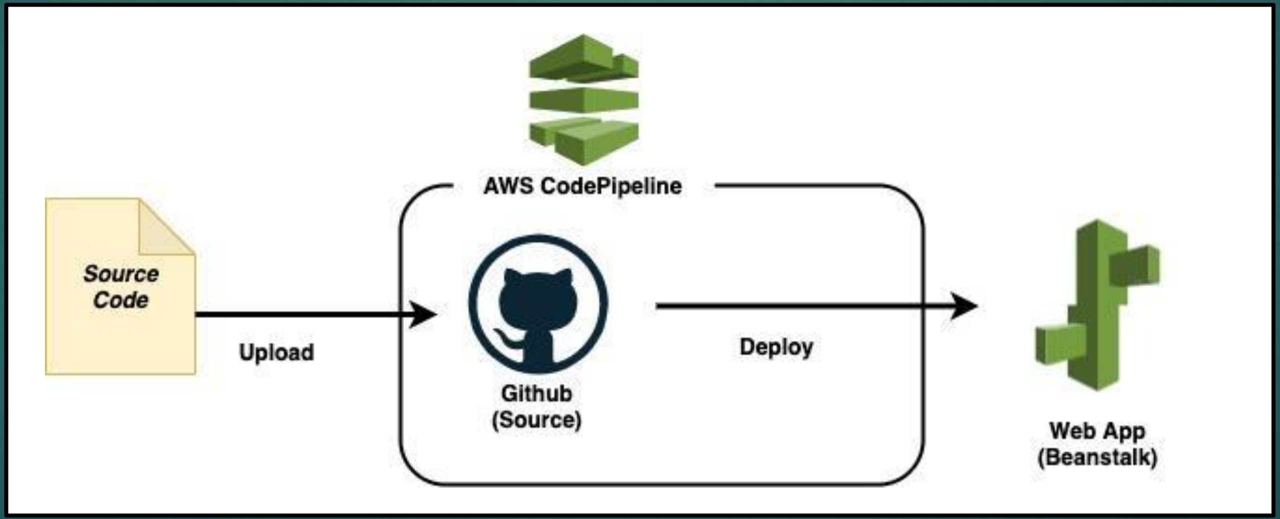
Create a DevOps Pipeline

SWEN 514-614: Engineering Cloud Software Systems

Department of Software Engineering
Rochester Institute of Technology

Overview

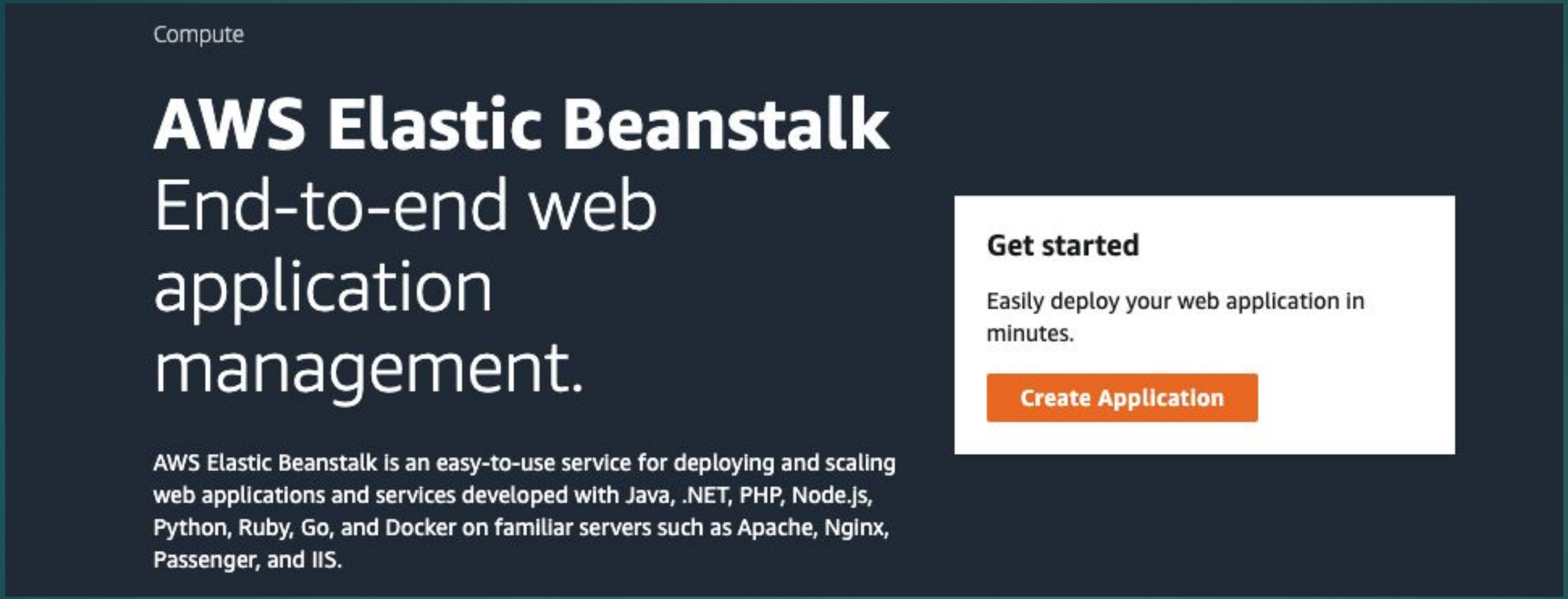
- In this activity, you create a Continuous Deployment pipeline that deploys a live sample web application
- This will use AWS CodePipeline to deploy code from your Github account to an Elastic Beanstalk NodeJS application
- Below is an illustration of the process



- There are 2 deliverables for this activity, which is worth 2 points

Create an app in Elastic Beanstalk

- Go to the AWS console and select "Elastic Beanstalk"



Compute

AWS Elastic Beanstalk

End-to-end web application management.

AWS 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.

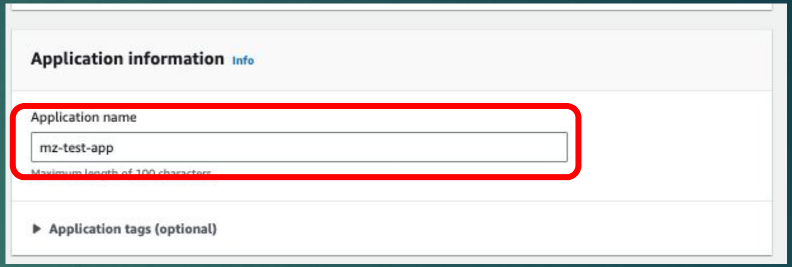
Get started

Easily deploy your web application in minutes.

- Select the "Create Application" button

Create an app in Elastic Beanstalk

- Under “Application information” enter a name for “Application name”
- Under “Platform” select Node.js and keep all the defaults
- Click “Next”



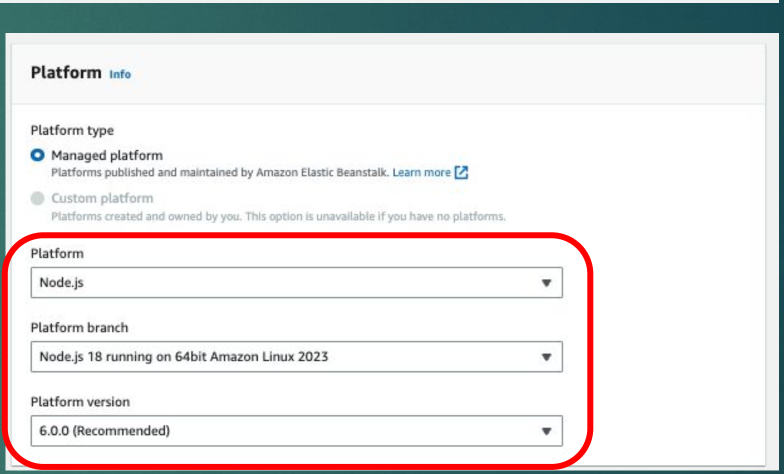
Application information Info

Application name

mz-test-app

Maximum length of 100 characters.

▸ Application tags (optional)



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

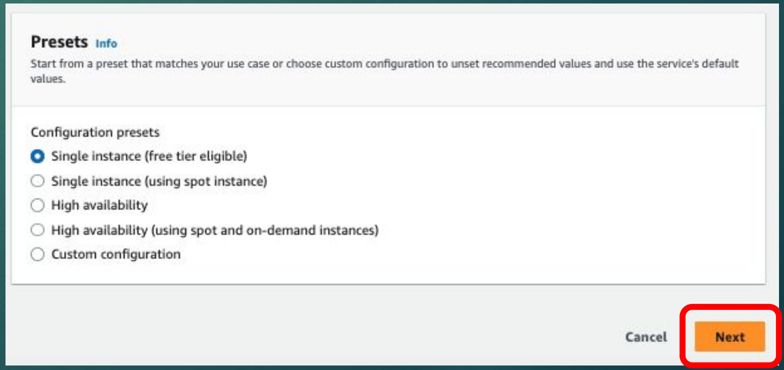
Node.js ▾

Platform branch

Node.js 18 running on 64bit Amazon Linux 2023 ▾

Platform version

6.0.0 (Recommended) ▾



Presets Info

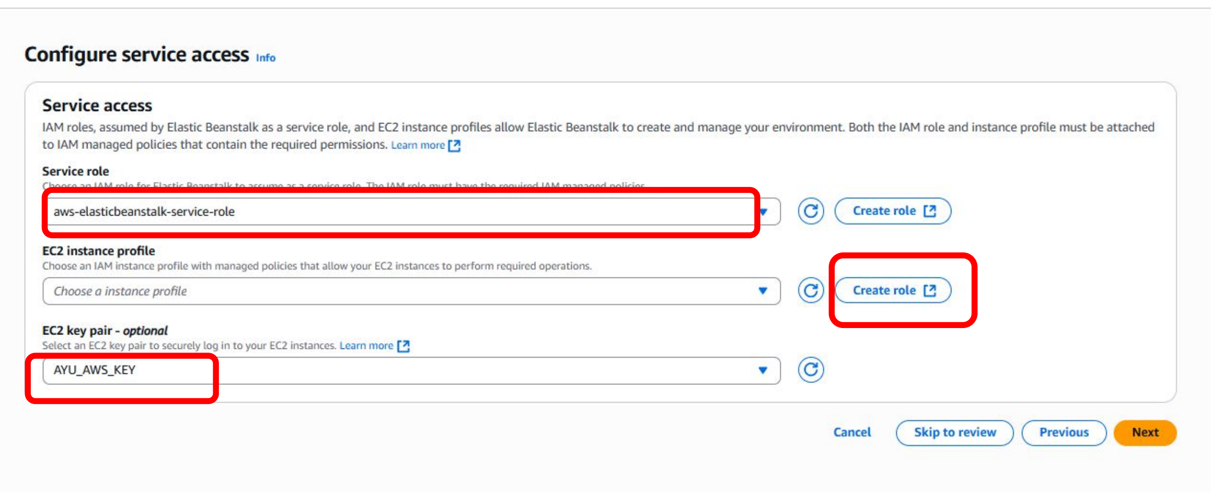
Start from a preset that matches your use case or choose custom configuration to unset recommended values and use the service's default values.

Configuration presets

● Single instance (free tier eligible)
○ Single instance (using spot instance)
○ High availability
○ High availability (using spot and on-demand instances)
○ Custom configuration

Create an app in Elastic Beanstalk

- Under “Service role”, select the default “aws-elasticbeanstalk-service-role”
- Under “EC2 key pair”, select your key pair
- Under EC2 instance profile, we need to create an IAM instance profile that will allow EC2 to communicate with other AWS services
  - Click the “Create role” button



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

Choose an IAM role for Elastic Beanstalk to assume as a service role. The IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 instance profile

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

Choose a instance profile

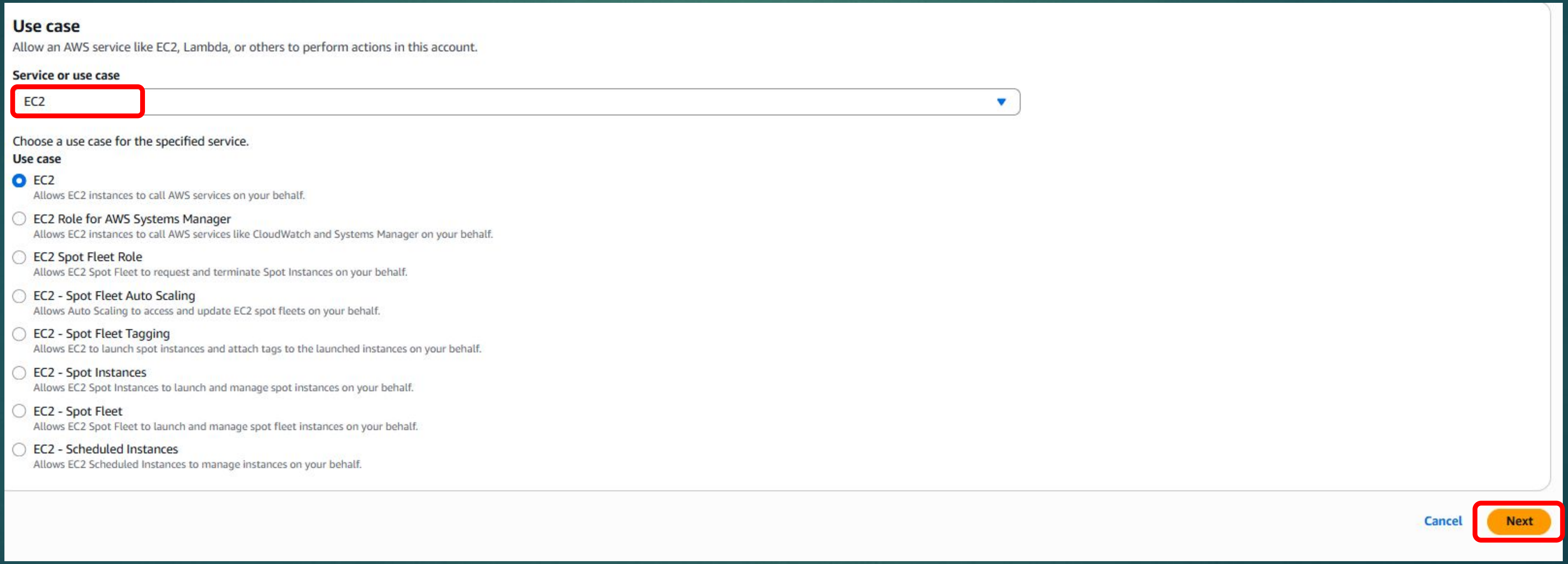
EC2 key pair - *optional*

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

AYU_AWS_KEY

Create IAM Role

- You'll be redirected to the IAM console.
- Select EC2 as the Use Case in the dropdown menu and click “Next” :



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.

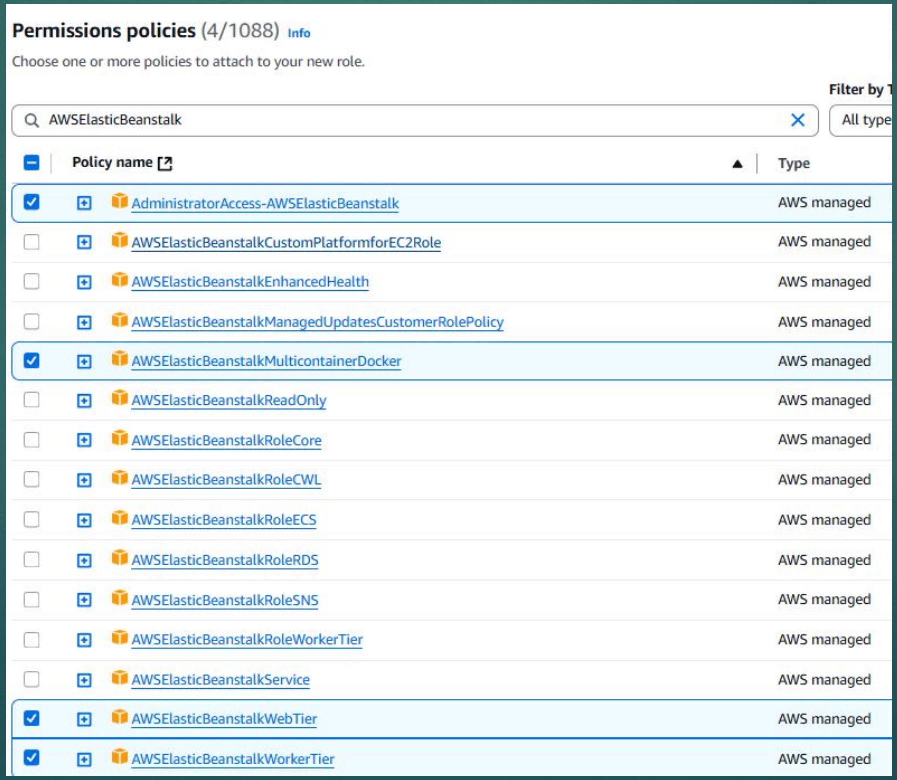
○ EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

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

○ EC2 - Scheduled Instances
Allows EC2 Scheduled Instances to manage instances on your behalf.

Create IAM Role

- In the search box, enter "AWSElasticBeanstalk" and hit Enter
- A subset of permissions is returned
- Select the these 4 permissions (see below)
- Click "Next"



Permissions policies (4/1088) Info

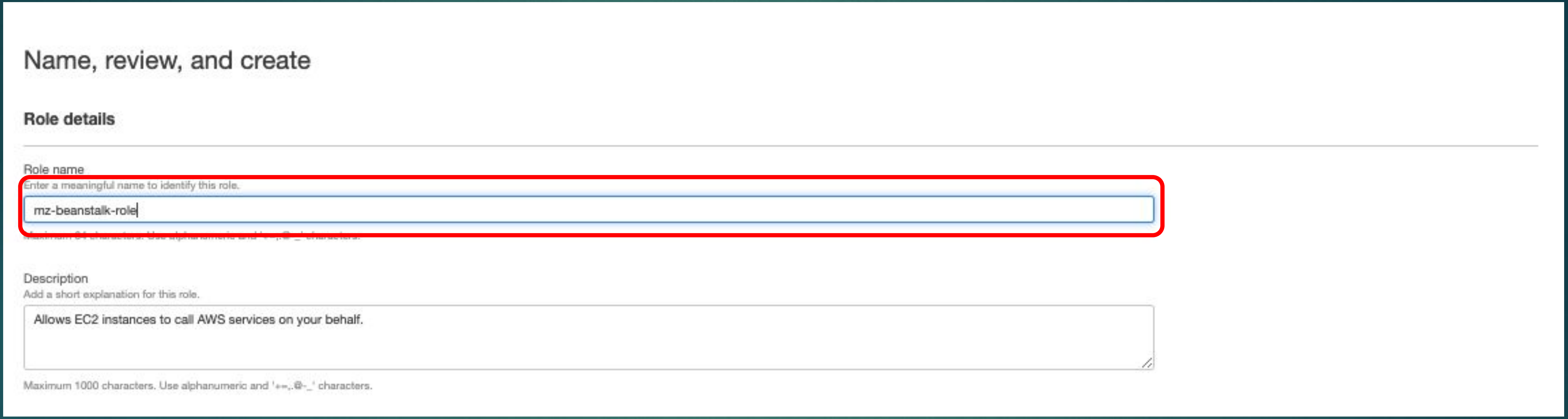
Choose one or more policies to attach to your new role.

AWSElasticBeanstalk

⊟	Policy name	Type
☑	AdministratorAccess-AWSElasticBeanstalk	AWS managed
☐	AWSElasticBeanstalkCustomPlatformforEC2Role	AWS managed
☐	AWSElasticBeanstalkEnhancedHealth	AWS managed
☐	AWSElasticBeanstalkManagedUpdatesCustomerRolePolicy	AWS managed
☑	AWSElasticBeanstalkMulticontainerDocker	AWS managed
☐	AWSElasticBeanstalkReadOnly	AWS managed
☐	AWSElasticBeanstalkRoleCore	AWS managed
☐	AWSElasticBeanstalkRoleCWL	AWS managed
☐	AWSElasticBeanstalkRoleECS	AWS managed
☐	AWSElasticBeanstalkRoleRDS	AWS managed
☐	AWSElasticBeanstalkRoleSNS	AWS managed
☐	AWSElasticBeanstalkRoleWorkerTier	AWS managed
☐	AWSElasticBeanstalkService	AWS managed
☑	AWSElasticBeanstalkWebTier	AWS managed
☑	AWSElasticBeanstalkWorkerTier	AWS managed

Create IAM Role

- Give your role a name and click “Create role”



Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.

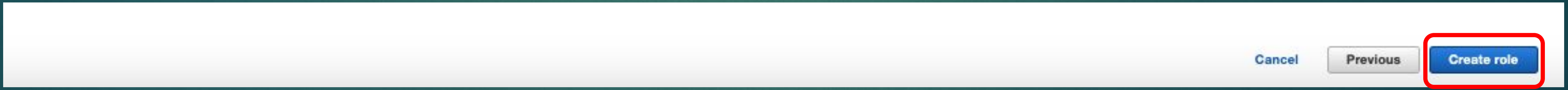
mz-beanstalk-role

Maximum 64 characters. Use alphanumeric and '+=,.@-_' characters.

Description
Add a short explanation for this role.

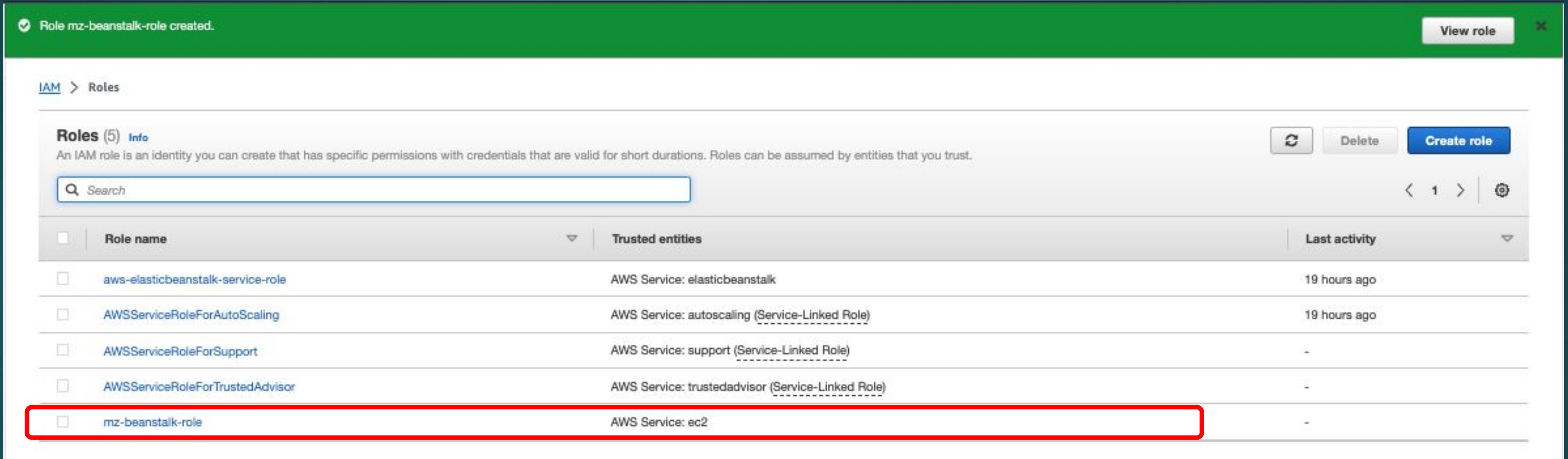
Allows EC2 instances to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.



Create IAM Role

- Verify your role is created and go back the the tab where you were configuring your Beanstalk app



Role mz-beanstalk-role created.

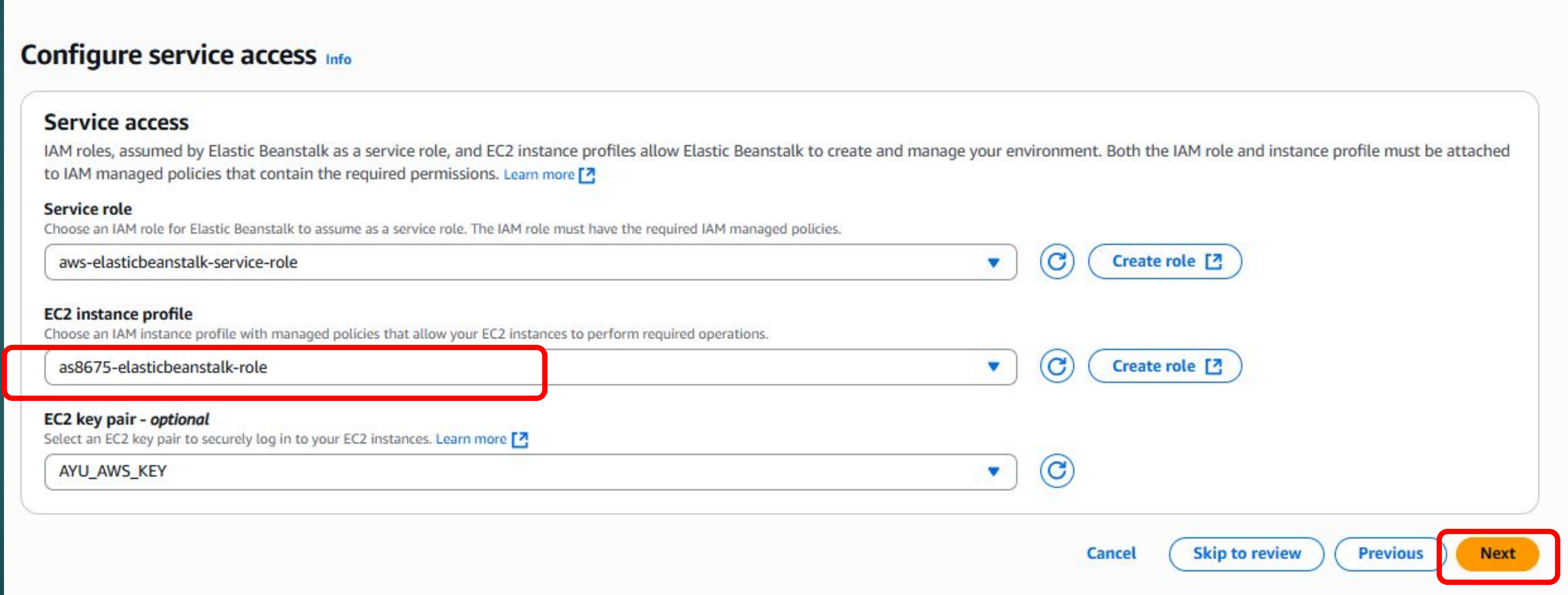
Roles (5) Info

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-service-role	AWS Service: elasticbeanstalk	19 hours ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	19 hours ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
mz-beanstalk-role	AWS Service: ec2	-

Create an app in Elastic Beanstalk

- Your new instance profile should now be available under "EC2 instance profile"
- Select it and click "Next".
- Click "Skip to review" and click "Create"



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

Choose an IAM role for Elastic Beanstalk to assume as a service role. The IAM role must have the required IAM managed policies.

aws-elasticbeanstalk-service-role

EC2 instance profile

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

as8675-elasticbeanstalk-role

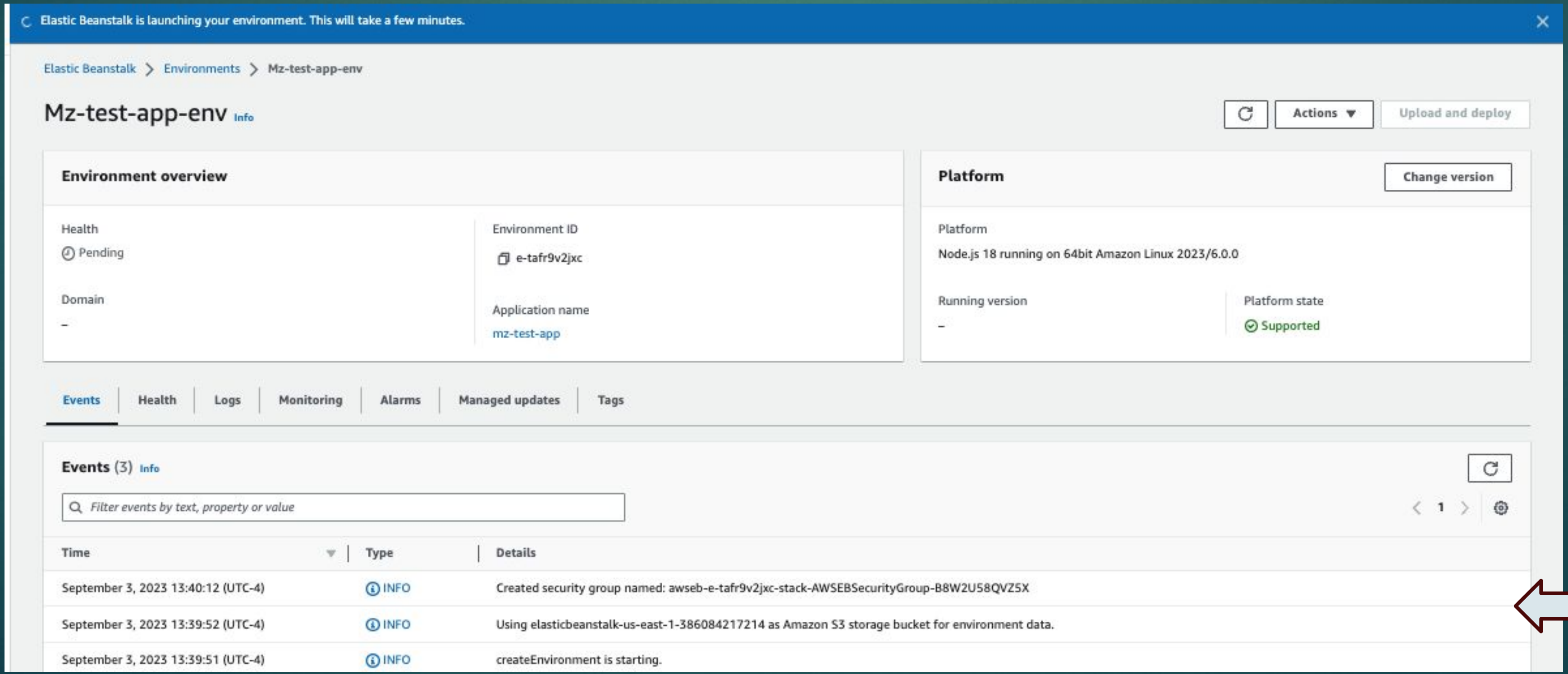
EC2 key pair - *optional*

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

AYU_AWS_KEY

Create an app in Elastic Beanstalk

- Your app is now being created and will take about 3-5 minutes
- Note the steps in “Events” are from a CloudFormation script that’s creating the application
- Feel free to check CloudFormation as it will show several more steps



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

Mz-test-app-env

Environment overview

Health: Pending

Environment ID: e-tafr9v2jxc

Domain: –

Application name: mz-test-app

Platform

Platform: Node.js 18 running on 64bit Amazon Linux 2023/6.0.0

Running version: –

Platform state: Supported

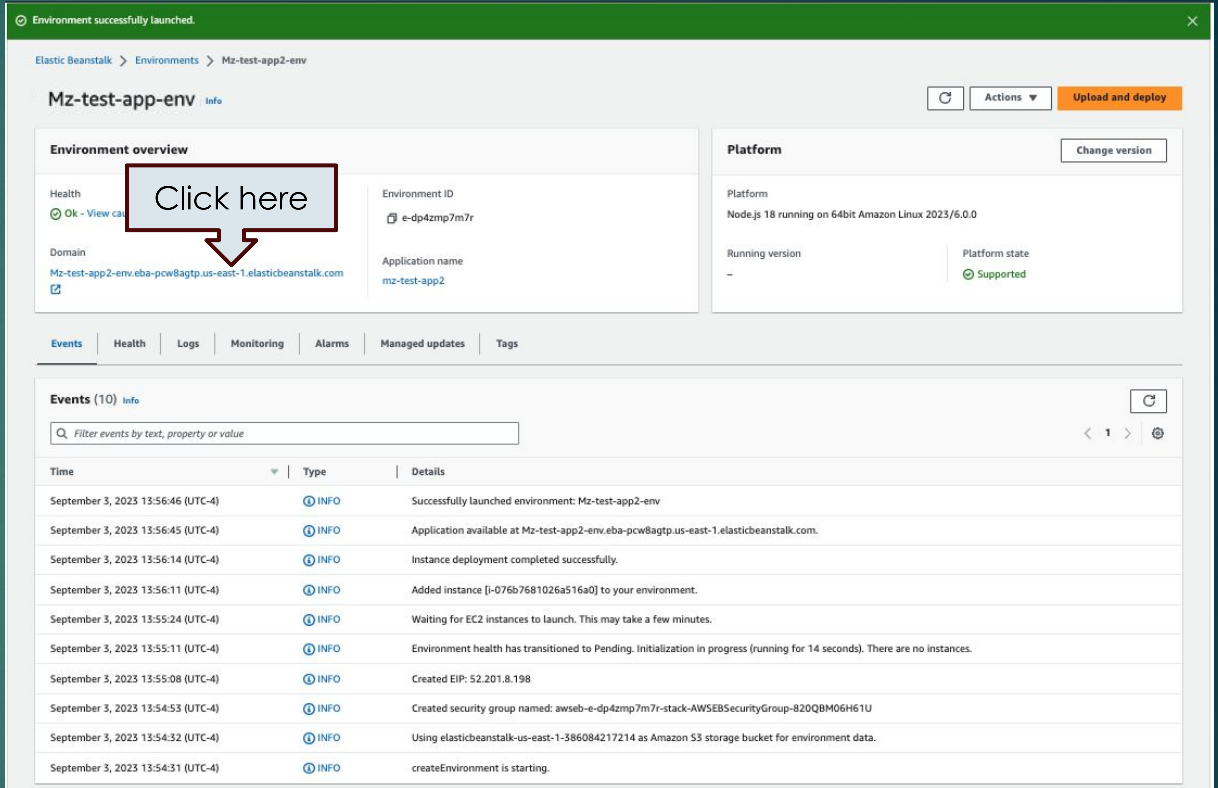
Events (3)

Time	Type	Details
September 3, 2023 13:40:12 (UTC-4)	INFO	Created security group named: awseb-e-tafr9v2jxc-stack-AWSEBSecurityGroup-B8W2U58QVZ5X
September 3, 2023 13:39:52 (UTC-4)	INFO	Using elasticbeanstalk-us-east-1-386084217214 as Amazon S3 storage bucket for environment data.
September 3, 2023 13:39:51 (UTC-4)	INFO	createEnvironment is starting.

CloudFormation steps

Create an app in Elastic Beanstalk

- When completed, you should see a success message
- Click the link under "Domain"



Click here

Environment successfully launched.

Elastic Beanstalk > Environments > Mz-test-app2-env

Mz-test-app-env

Environment overview

Health: Ok - View ca

Domain: Mz-test-app2-env.eba-pcw8agtp.us-east-1.elasticbeanstalk.com

Environment ID: e-dp4zmp7m7r

Application name: mz-test-app2

Platform

Platform: Node.js 18 running on 64bit Amazon Linux 2023/6.0.0

Running version: –

Platform state: Supported

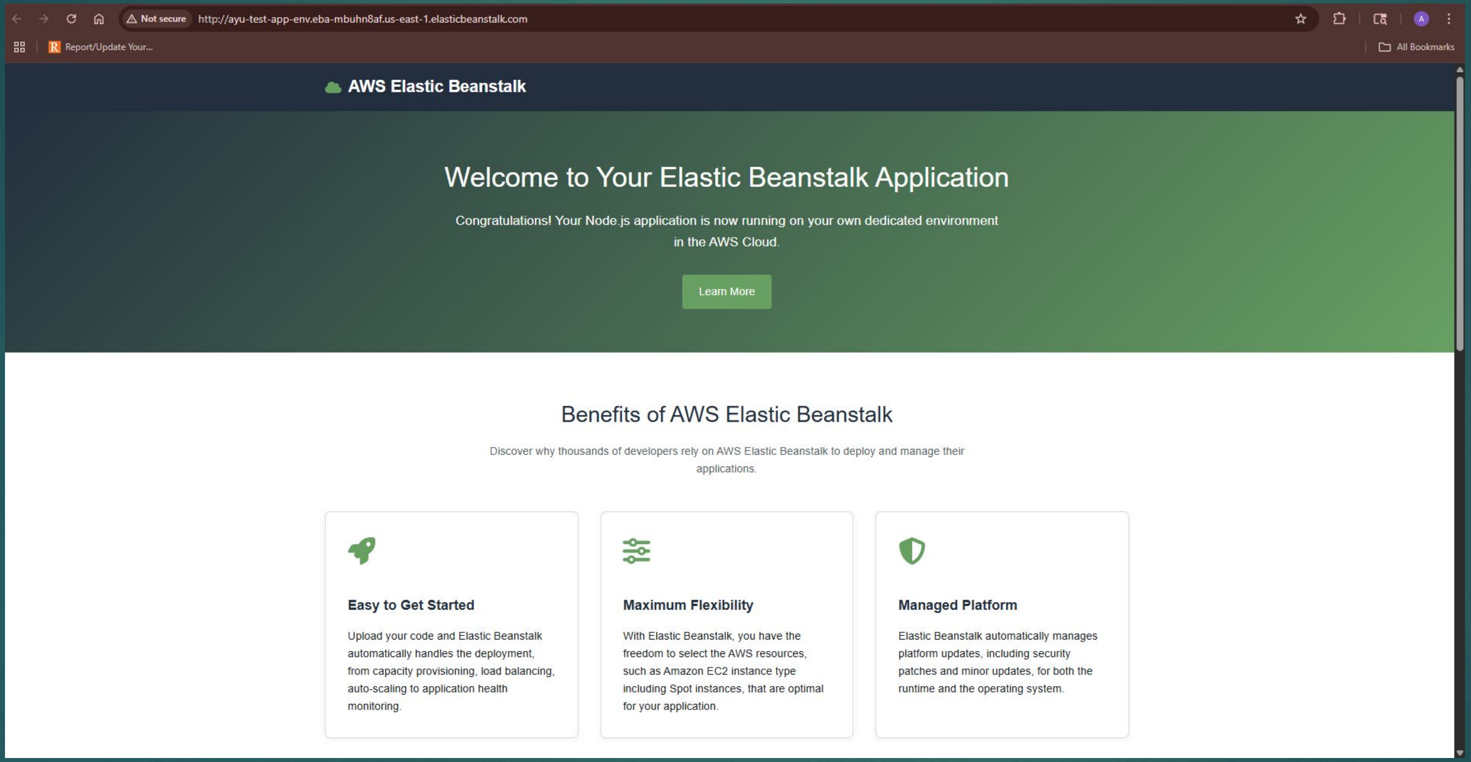
Events (10) Info

Time	Type	Details
September 3, 2023 13:56:46 (UTC-4)	INFO	Successfully launched environment: Mz-test-app2-env
September 3, 2023 13:56:45 (UTC-4)	INFO	Application available at Mz-test-app2-env.eba-pcw8agtp.us-east-1.elasticbeanstalk.com.
September 3, 2023 13:56:14 (UTC-4)	INFO	Instance deployment completed successfully.
September 3, 2023 13:56:11 (UTC-4)	INFO	Added instance [i-076b7681026a516a0] to your environment.
September 3, 2023 13:55:24 (UTC-4)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
September 3, 2023 13:55:11 (UTC-4)	INFO	Environment health has transitioned to Pending. Initialization in progress (running for 14 seconds). There are no instances.
September 3, 2023 13:55:08 (UTC-4)	INFO	Created EIP: 52.201.8.198
September 3, 2023 13:54:53 (UTC-4)	INFO	Created security group named: awseb-e-dp4zmp7m7r-stack-AWSEBSecurityGroup-820QBM06H61U
September 3, 2023 13:54:32 (UTC-4)	INFO	Using elasticbeanstalk-us-east-1-386084217214 as Amazon S3 storage bucket for environment data.
September 3, 2023 13:54:31 (UTC-4)	INFO	createEnvironment is starting.

- Note: If Beanstalk is taking longer than 5 minutes, do the following:
  - Go to CloudFormation and delete the Stack
  - Start over on slide #4

Create an app in Elastic Beanstalk

- The sample application has been deployed and is running


Not secure
http://ayu-test-app-env.eba-mbuhn8af.us-east-1.elasticbeanstalk.com
Report/Update Your...
All Bookmarks

AWS Elastic Beanstalk

Welcome to Your Elastic Beanstalk Application

Congratulations! Your Node.js application is now running on your own dedicated environment in the AWS Cloud.

Benefits of AWS Elastic Beanstalk

Discover why thousands of developers rely on AWS Elastic Beanstalk to deploy and manage their applications.

Easy to Get Started

Upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring.

Maximum Flexibility

With Elastic Beanstalk, you have the freedom to select the AWS resources, such as Amazon EC2 instance type including Spot instances, that are optimal for your application.

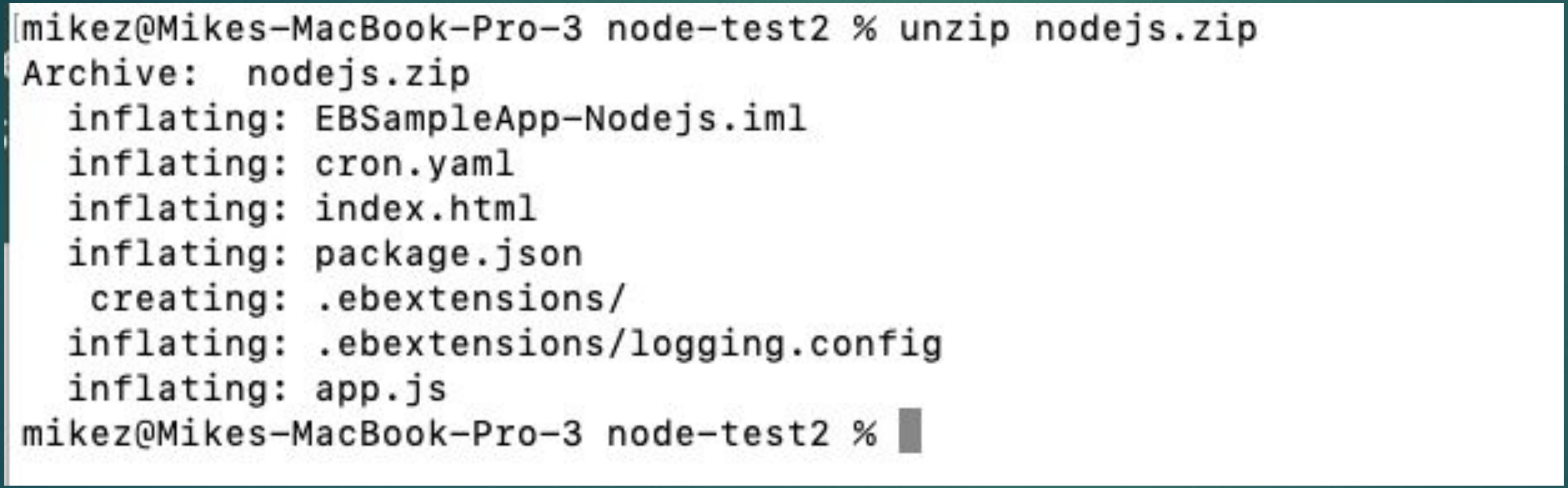
Managed Platform

Elastic Beanstalk automatically manages platform updates, including security patches and minor updates, for both the runtime and the operating system.

- Next, we will connect a source code repository (Github) to the application
- When the repository is modified, this will trigger the application to redeploy

Download source files

- The source for the sample app (nodejs.zip) can be found under Assignments > Activity #14 - Create a DevOps Pipeline
- Download this file to your PC and unzip

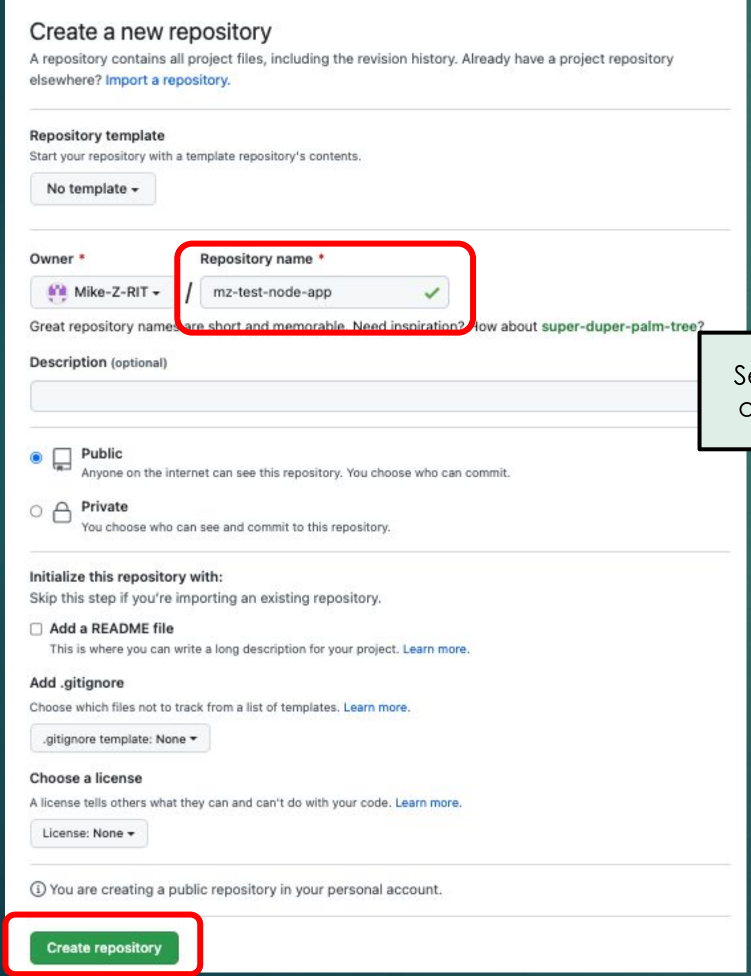


```
mikez@Mikes-MacBook-Pro-3 node-test2 % unzip nodejs.zip
Archive:  nodejs.zip
  inflating: EBSampleApp-Nodejs.iml
  inflating: cron.yaml
  inflating: index.html
  inflating: package.json
   creating: .ebextensions/
  inflating: .ebextensions/logging.config
  inflating: app.js
mikez@Mikes-MacBook-Pro-3 node-test2 %
```

- Next, go to Github to create a new repo

Create a Repo

- Create a new repo in Github and click "Create repository"



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template

Start your repository with a template repository's contents.

No template ▾

Owner * **Repository name** *

Mike-Z-RIT ▾ / mz-test-node-app ✓

Great repository names are short and memorable. Need inspiration? How about **super-duper-palm-tree**?

Description (optional)

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

□ **Add a README file**
This is where you can write a long description for your project. Learn more.

Add .gitignore

Choose which files not to track from a list of templates. Learn more.

.gitignore template: None ▾

Choose a license

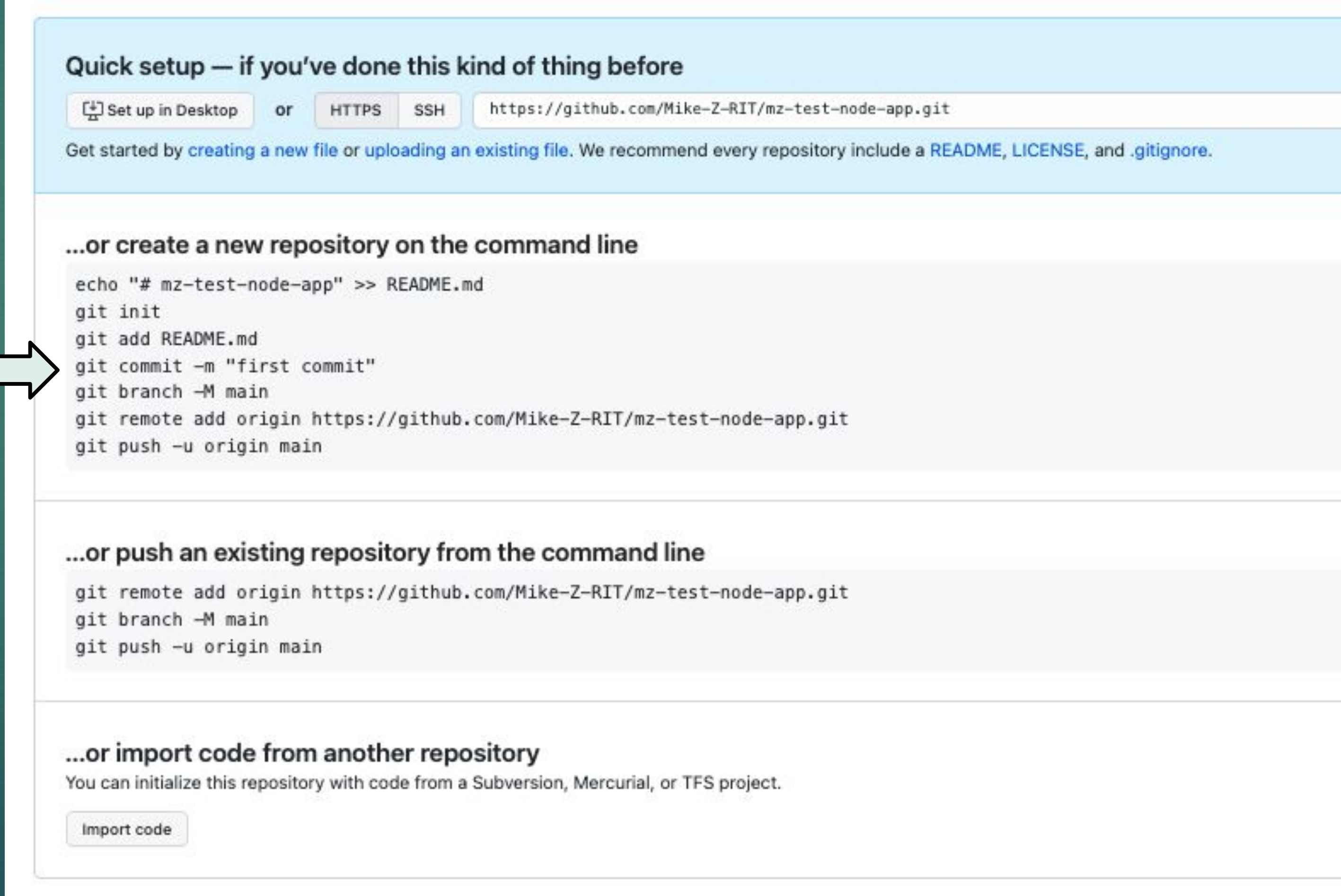
A license tells others what they can and can't do with your code. Learn more.

License: None ▾

ⓘ You are creating a public repository in your personal account.

See example on next slide

- Follow the instructions to import the sample app to your repo



Quick setup — if you've done this kind of thing before

https://github.com/Mike-Z-RIT/mz-test-node-app.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

```
echo "# mz-test-node-app" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Mike-Z-RIT/mz-test-node-app.git
git push -u origin main
```

...or push an existing repository from the command line

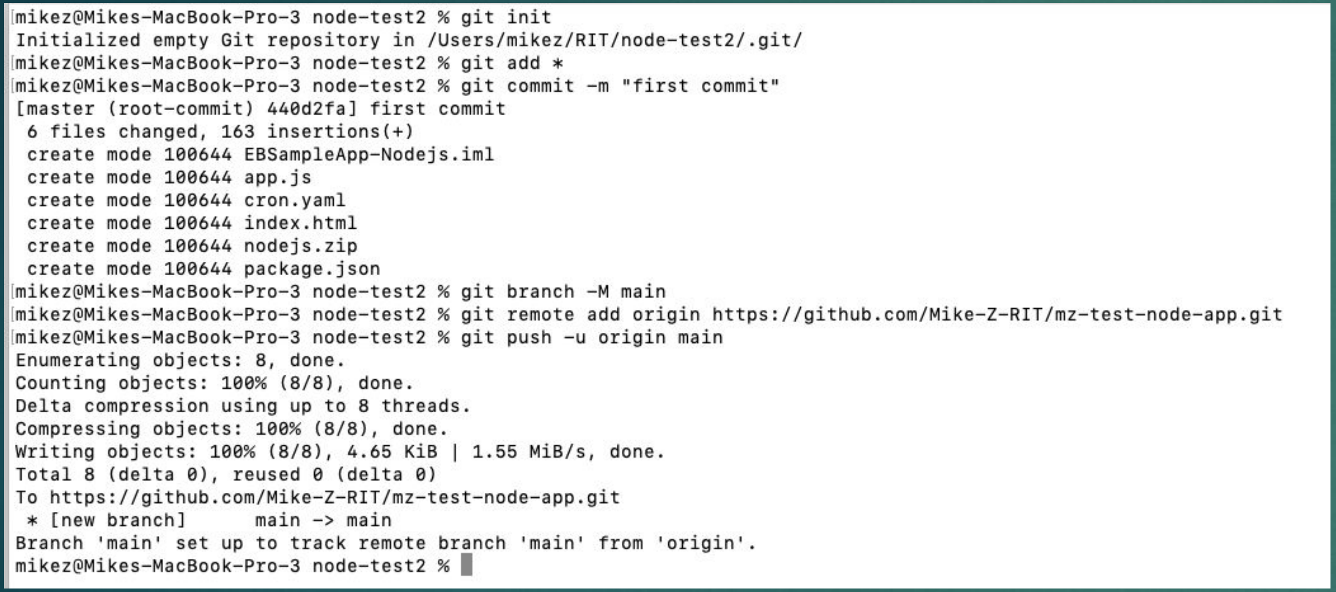
```
git remote add origin https://github.com/Mike-Z-RIT/mz-test-node-app.git
git branch -M main
git push -u origin main
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

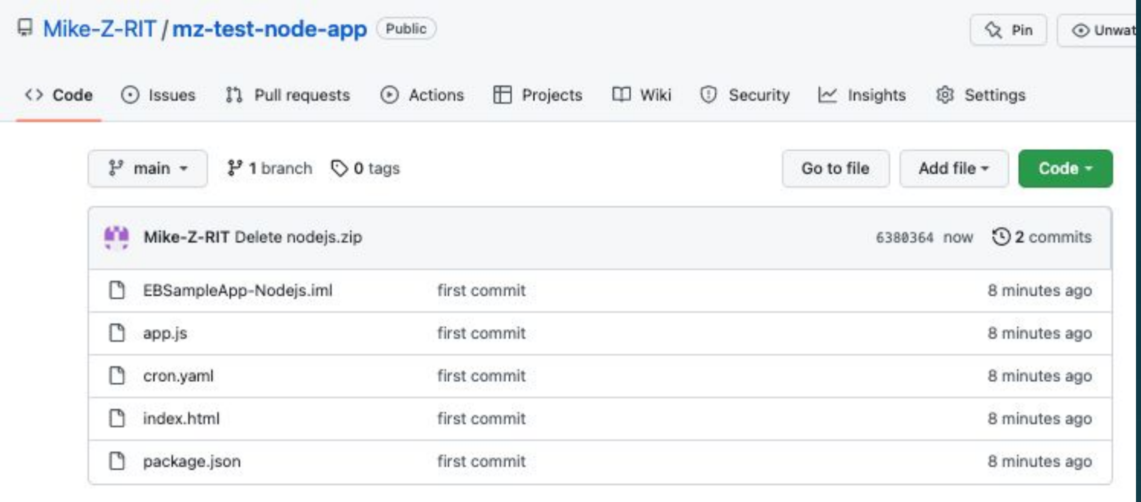
Create a Repo (example)

- Here is an example of pushing the files to Github following the steps from the previous slide



```
mikez@Mikes-MacBook-Pro-3 node-test2 % git init
Initialized empty Git repository in /Users/mikez/RIT/node-test2/.git/
mikez@Mikes-MacBook-Pro-3 node-test2 % git add *
mikez@Mikes-MacBook-Pro-3 node-test2 % git commit -m "first commit"
[master (root-commit) 440d2fa] first commit
 6 files changed, 163 insertions(+)
 create mode 100644 EBSampleApp-Nodejs.iml
 create mode 100644 app.js
 create mode 100644 cron.yaml
 create mode 100644 index.html
 create mode 100644 nodejs.zip
 create mode 100644 package.json
mikez@Mikes-MacBook-Pro-3 node-test2 % git branch -M main
mikez@Mikes-MacBook-Pro-3 node-test2 % git remote add origin https://github.com/Mike-Z-RIT/mz-test-node-app.git
mikez@Mikes-MacBook-Pro-3 node-test2 % git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 8 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 4.65 KiB | 1.55 MiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://github.com/Mike-Z-RIT/mz-test-node-app.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
mikez@Mikes-MacBook-Pro-3 node-test2 %
```

- When complete, verify your repo has the files below

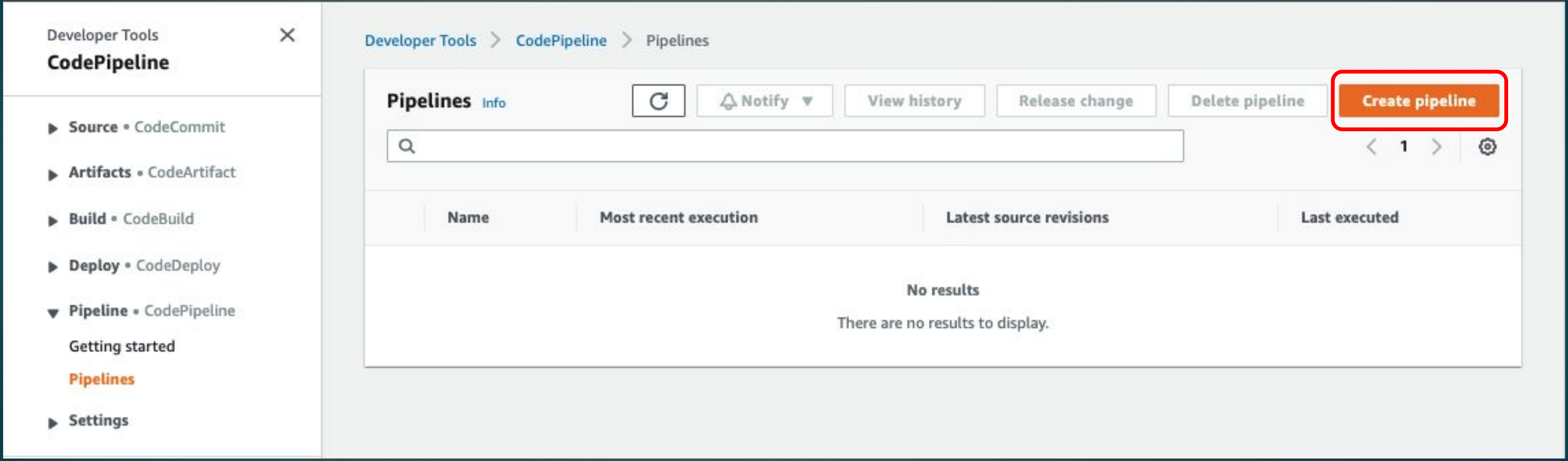


Mike-Z-RIT Delete nodejs.zip   6380364 now   2 commits

EBSampleApp-Nodejs.iml	first commit	8 minutes ago
app.js	first commit	8 minutes ago
cron.yaml	first commit	8 minutes ago
index.html	first commit	8 minutes ago
package.json	first commit	8 minutes ago

Create your Pipeline

- Go to the AWS console and select “CodePipeline”
- Click the “Create pipeline” button



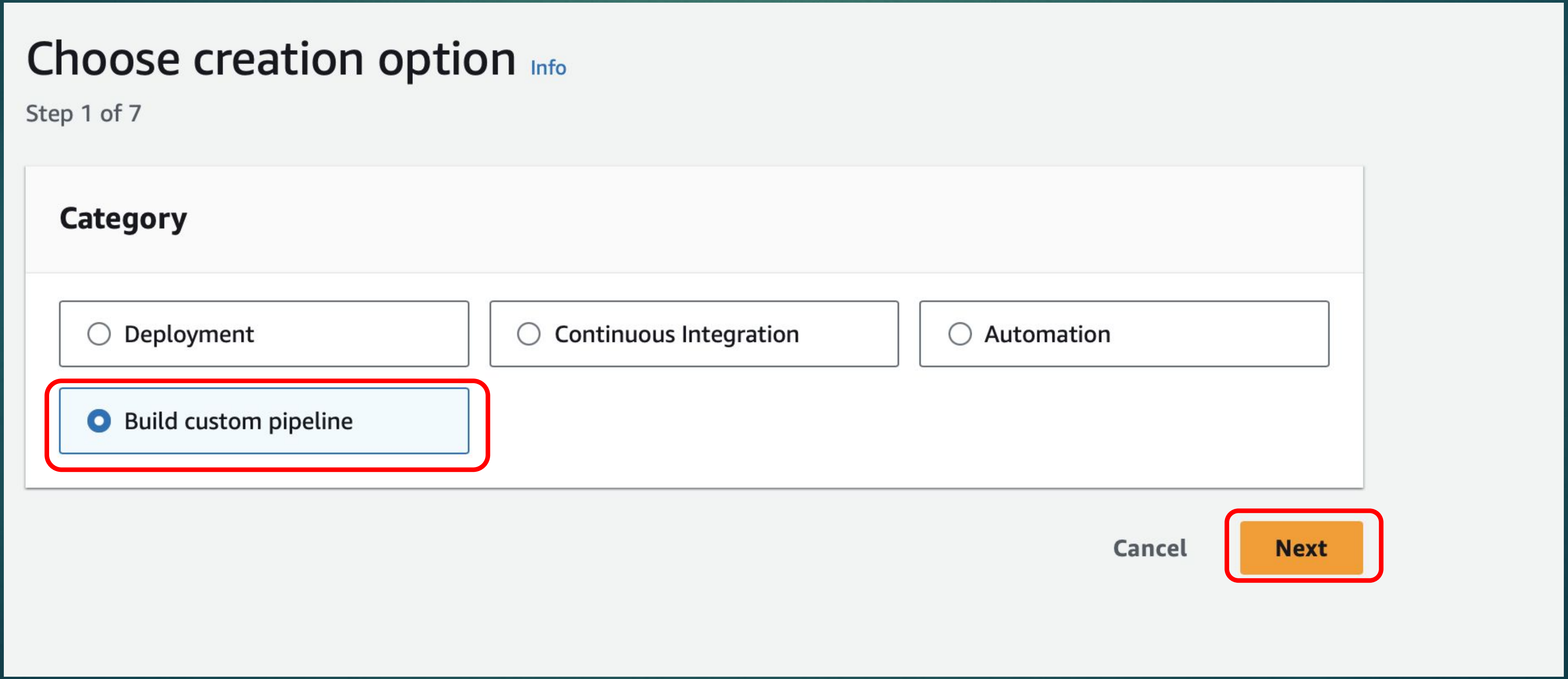
Name	Most recent execution	Latest source revisions	Last executed

No results

There are no results to display.

Create your Pipeline

- Under Choose creation option select “Build Custom Pipeline”
- Click “Next”



Choose creation option

Step 1 of 7

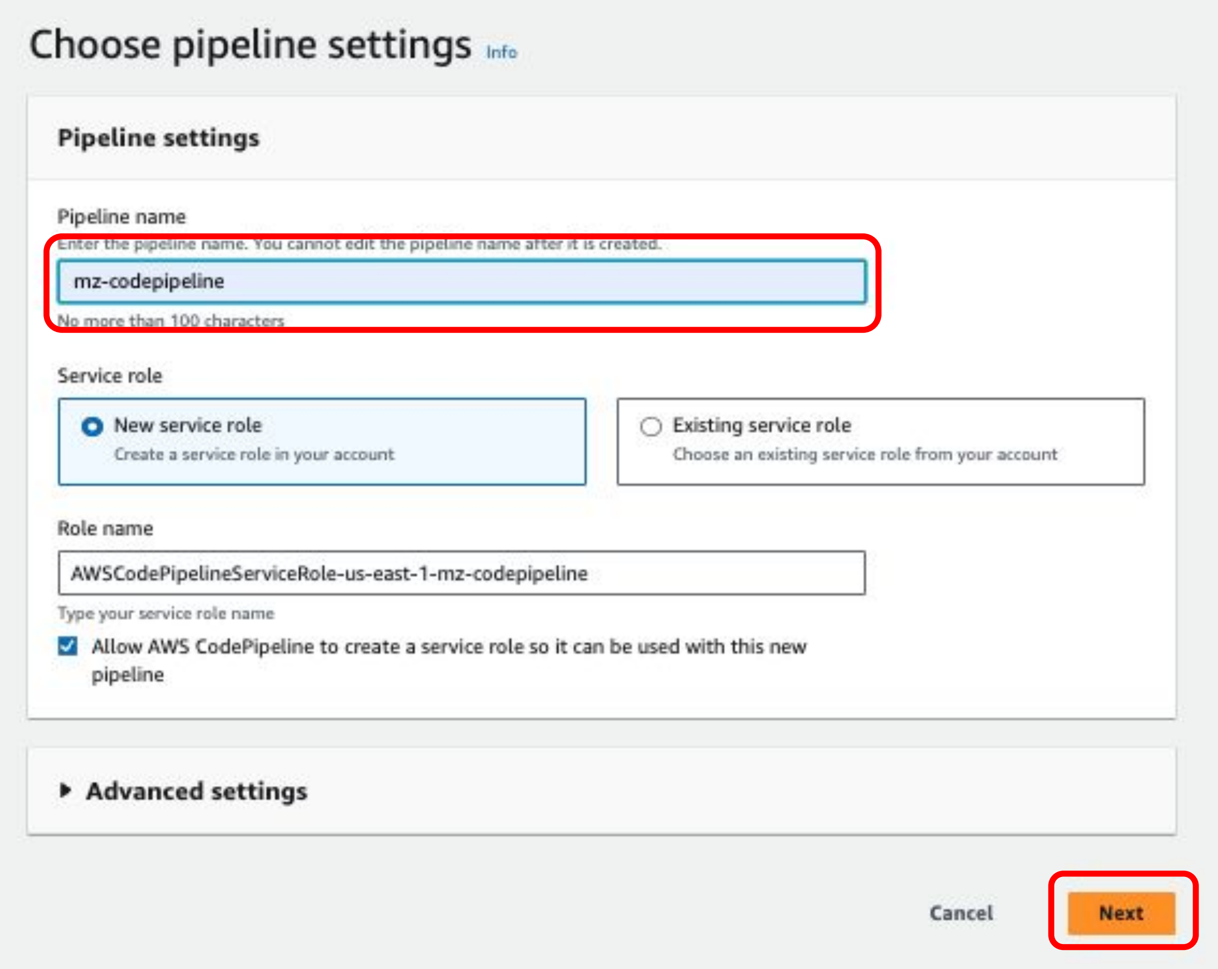
Category

○ Deployment ○ Continuous Integration ○ Automation

● Build custom pipeline

Create your Pipeline

- Enter name for "Pipeline name"
- Click "Next"



Choose pipeline settings

Pipeline settings

Pipeline name

Enter the pipeline name. You cannot edit the pipeline name after it is created.

mz-codepipeline

No more than 100 characters

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-mz-codepipeline

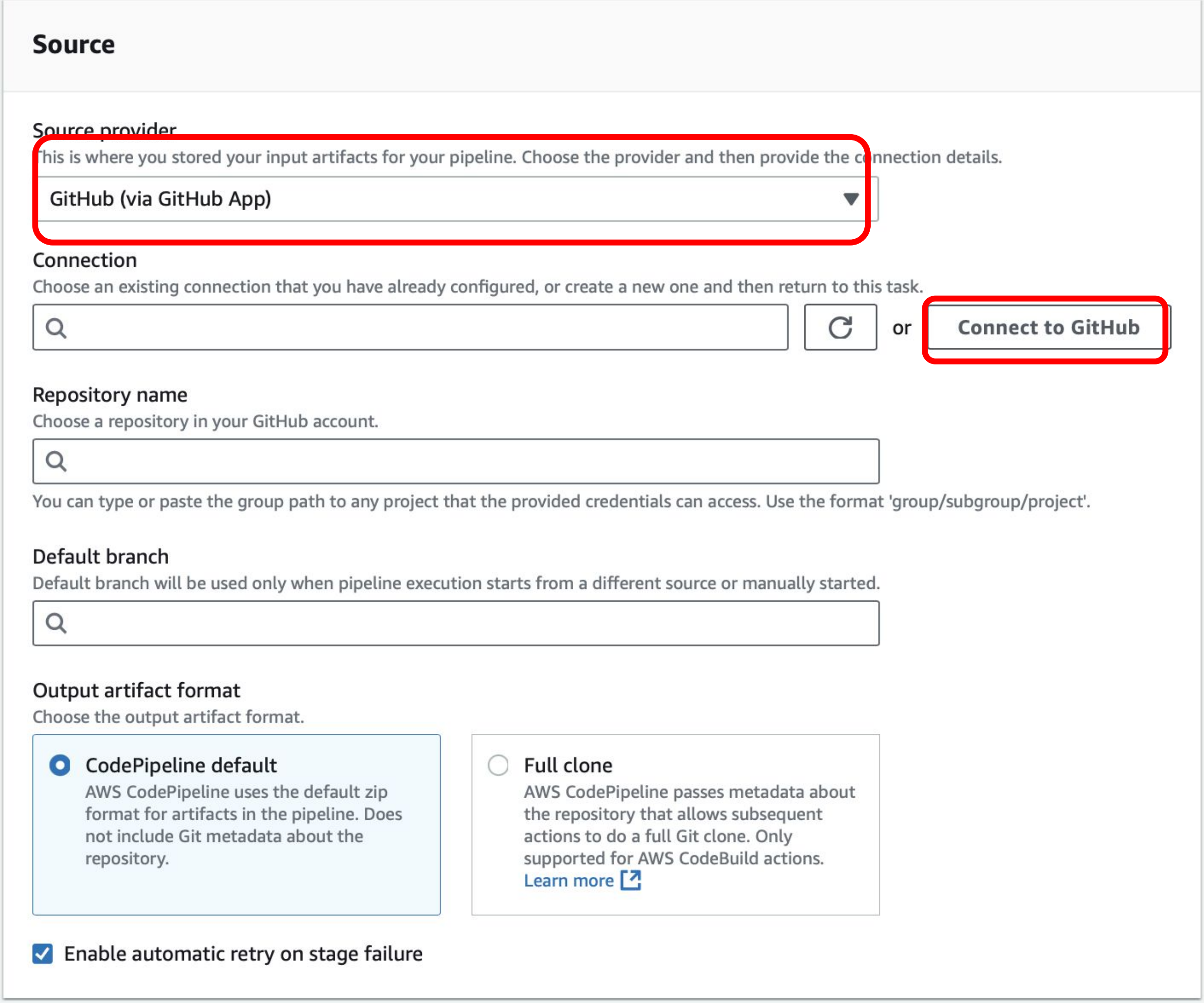
Type your service role name

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

▸ Advanced settings

Pipeline Source

- Under “Source Provider”, Select “Github (via Github APP)”
- Click the “Connect to Github” to link your AWS account to your Github account
- This is a multi-step process but only needs to be done once



Source

Source provider

This is where you stored your input artifacts for your pipeline. Choose the provider and then provide the connection details.

GitHub (via GitHub App) ▼

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.

Output artifact format

Choose the output artifact format.

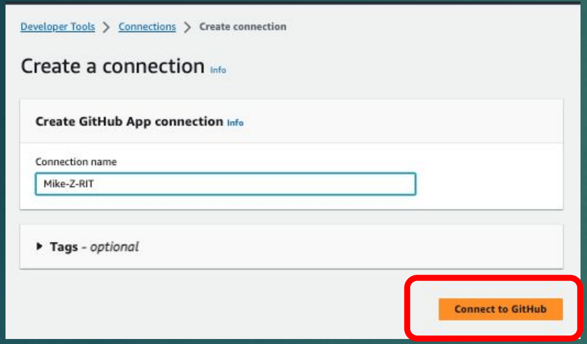
● CodePipeline default
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

○ 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.
Learn more

☑ Enable automatic retry on stage failure

Connect Pipeline to Github

#1 - To connect to Github, provide your Github ID and click “Connect to Github”



Developer Tools > Connections > Create connection

Create a connection Info

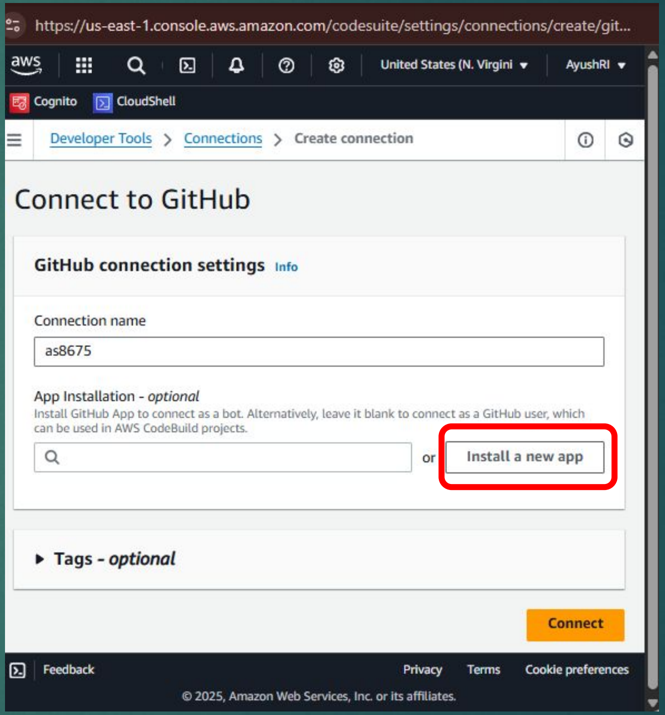
Create GitHub App connection Info

Connection name

Mike-Z-RIT

Tags - *optional*

#2 - Click the “Install a new app”



https://us-east-1.console.aws.amazon.com/codesuite/settings/connections/create/git...

Developer Tools > Connections > Create connection

Connect to GitHub

GitHub connection settings Info

Connection name

as8675

App Installation - *optional*

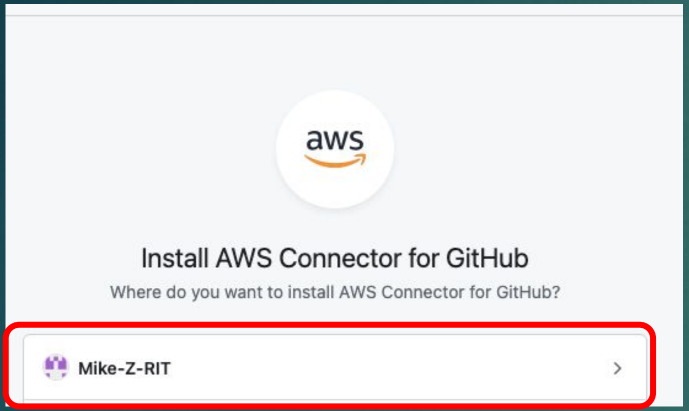
Install GitHub App to connect as a bot. Alternatively, leave it blank to connect as a GitHub user, which can be used in AWS CodeBuild projects.

or

© 2025, Amazon Web Services, Inc. or its affiliates.

Connect Pipeline to Github

#1- Select your Github account

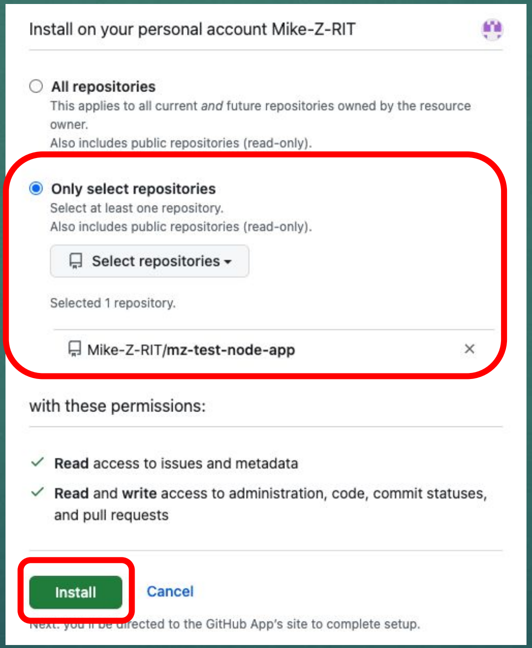


Install AWS Connector for GitHub

Where do you want to install AWS Connector for GitHub?

Mike-Z-RIT

#2 - Select “Only select repositories” and select the repo you just created. Click “Install”



Install on your personal account Mike-Z-RIT

○ **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).

Select repositories ▾

Selected 1 repository.

Mike-Z-RIT/**mz-test-node-app** ×

with these permissions:

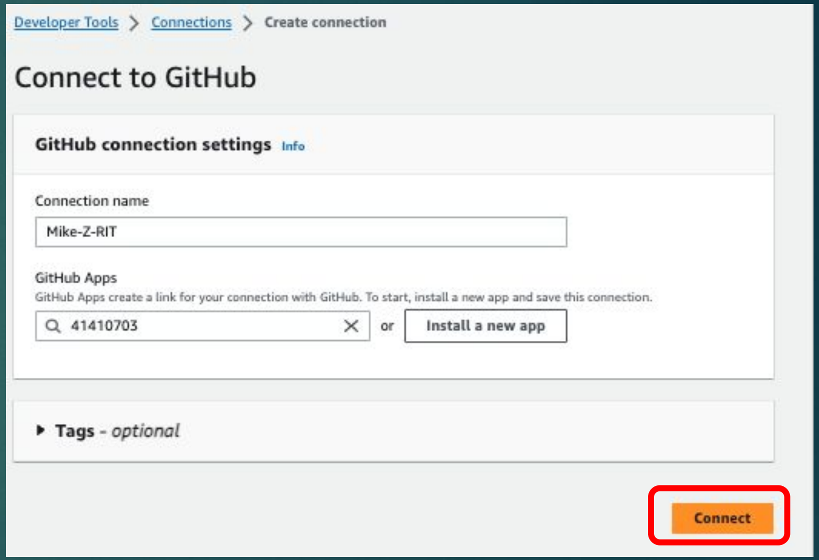
✓ **Read** access to issues and metadata

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

Install Cancel

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

#3 - Click “Connect” to connect AWS to your Github account



Developer Tools > Connections > Create connection

Connect to GitHub

GitHub connection settings Info

Connection name

Mike-Z-RIT

GitHub Apps

GitHub Apps create a link for your connection with GitHub. To start, install a new app and save this connection.

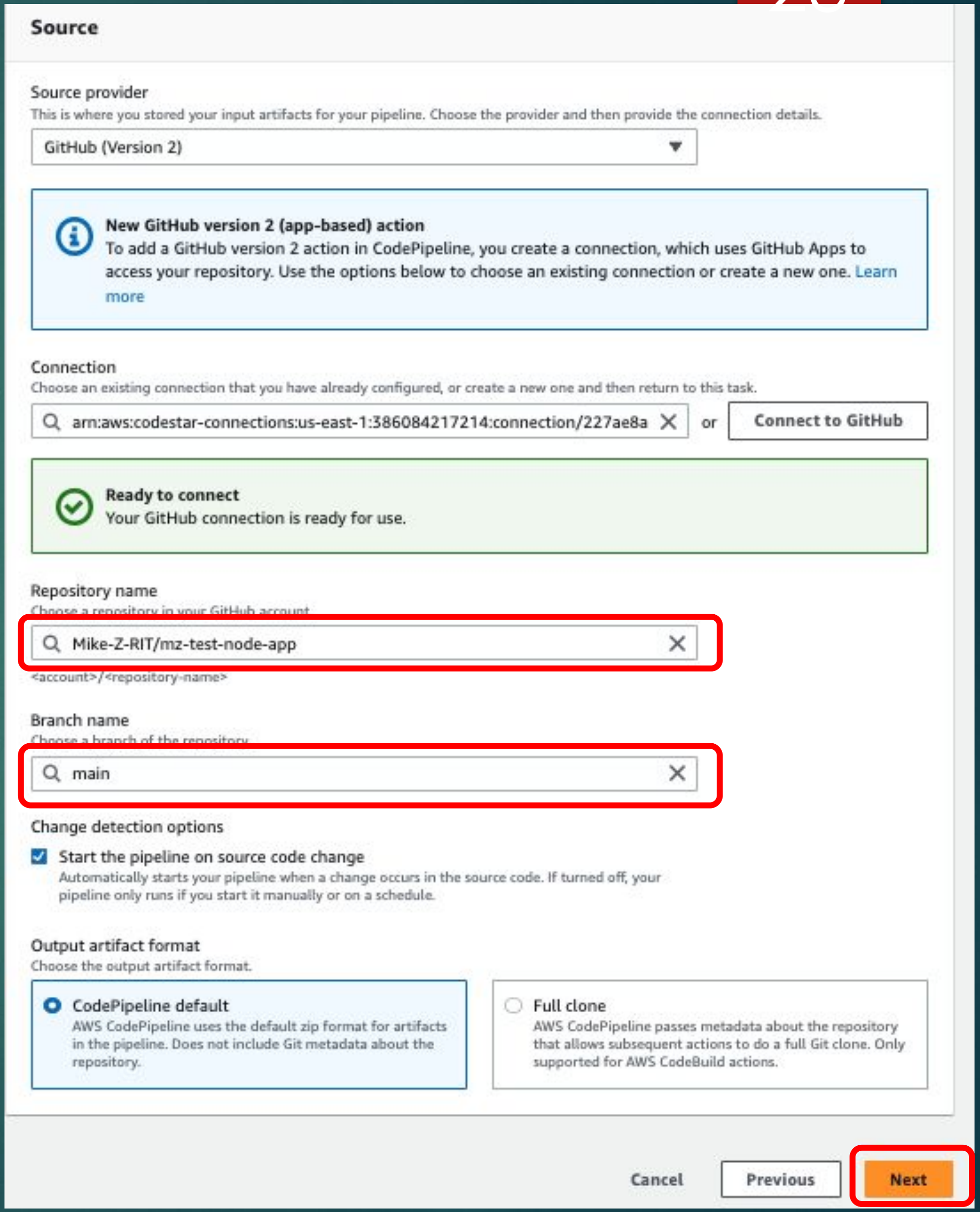
41410703 × or Install a new app

▸ **Tags** - *optional*

Connect

Pipeline Source

- Pipeline is now connected to Github
- Select the repo you previously created
- Select the "main" branch to trigger a deploy
- Your screen should look similar to the right
- Keep Trigger Type as "**No Filter**"
- Click "Next"



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:codestar-connections:us-east-1:386084217214:connection/227ae8a ✕ or Connect to GitHub

Ready to connect

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account

Mike-Z-RIT/mz-test-node-app ✕

<account>/<repository-name>

Branch name

Choose a branch of the repository

main ✕

Change detection options

☑ Start the pipeline on source code change

Automatically starts your pipeline when a change occurs in the source code. If turned off, your pipeline only runs if you start it manually or on a schedule.

Output artifact format

Choose the output artifact format.

● CodePipeline default

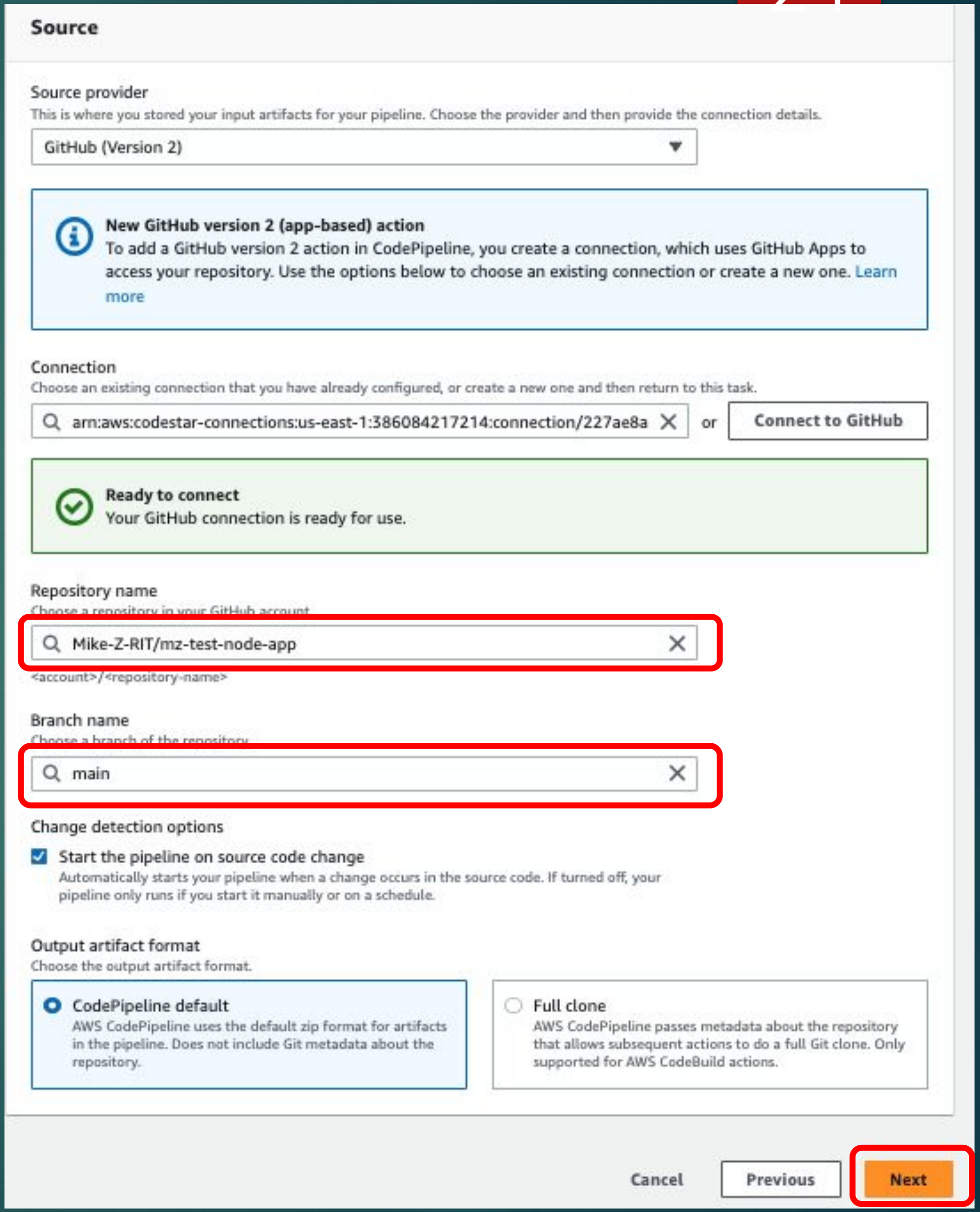
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

○ 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.

Pipeline Source

- Pipeline is now connected to Github
- Select the repo you previously created
- Select the "main" branch to trigger a deploy
- Your screen should look similar to the right
- Click "Next"



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:codestar-connections:us-east-1:386084217214:connection/227ae8a ✕ or Connect to GitHub

Ready to connect

Your GitHub connection is ready for use.

Repository name

Choose a repository in your GitHub account

Mike-Z-RIT/mz-test-node-app ✕

<account>/<repository-name>

Branch name

Choose a branch of the repository

main ✕

Change detection options

☑ Start the pipeline on source code change

Automatically starts your pipeline when a change occurs in the source code. If turned off, your pipeline only runs if you start it manually or on a schedule.

Output artifact format

Choose the output artifact format.

● CodePipeline default

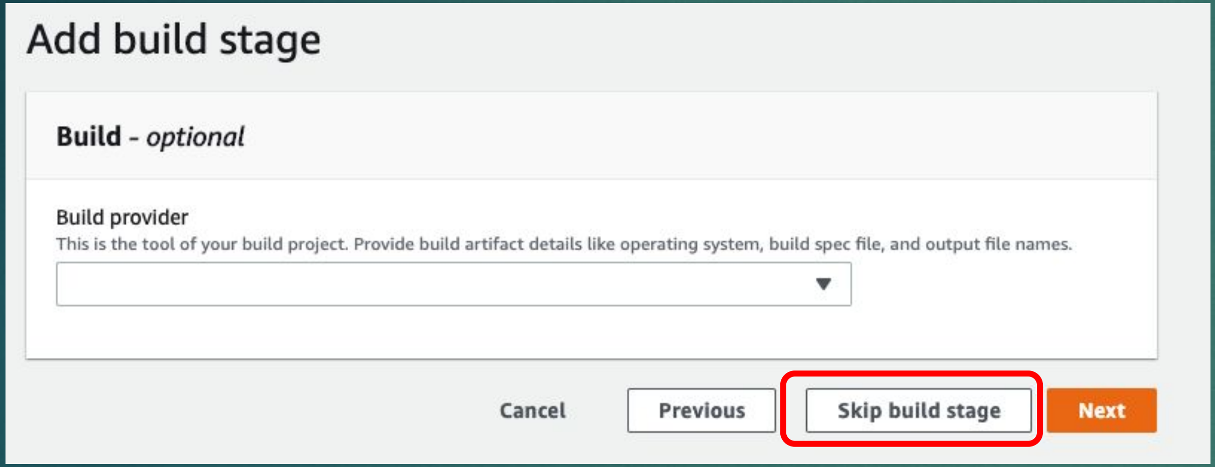
AWS CodePipeline uses the default zip format for artifacts in the pipeline. Does not include Git metadata about the repository.

○ 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.

Build Stage

- Click the “Skip build stage” and confirm by clicking “Skip”
- Then click the “Skip test stage”

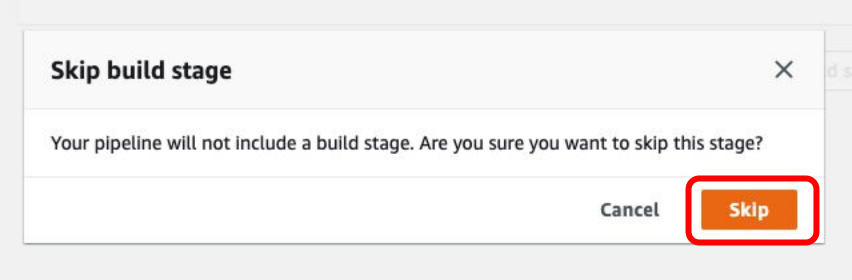


Add build stage

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.

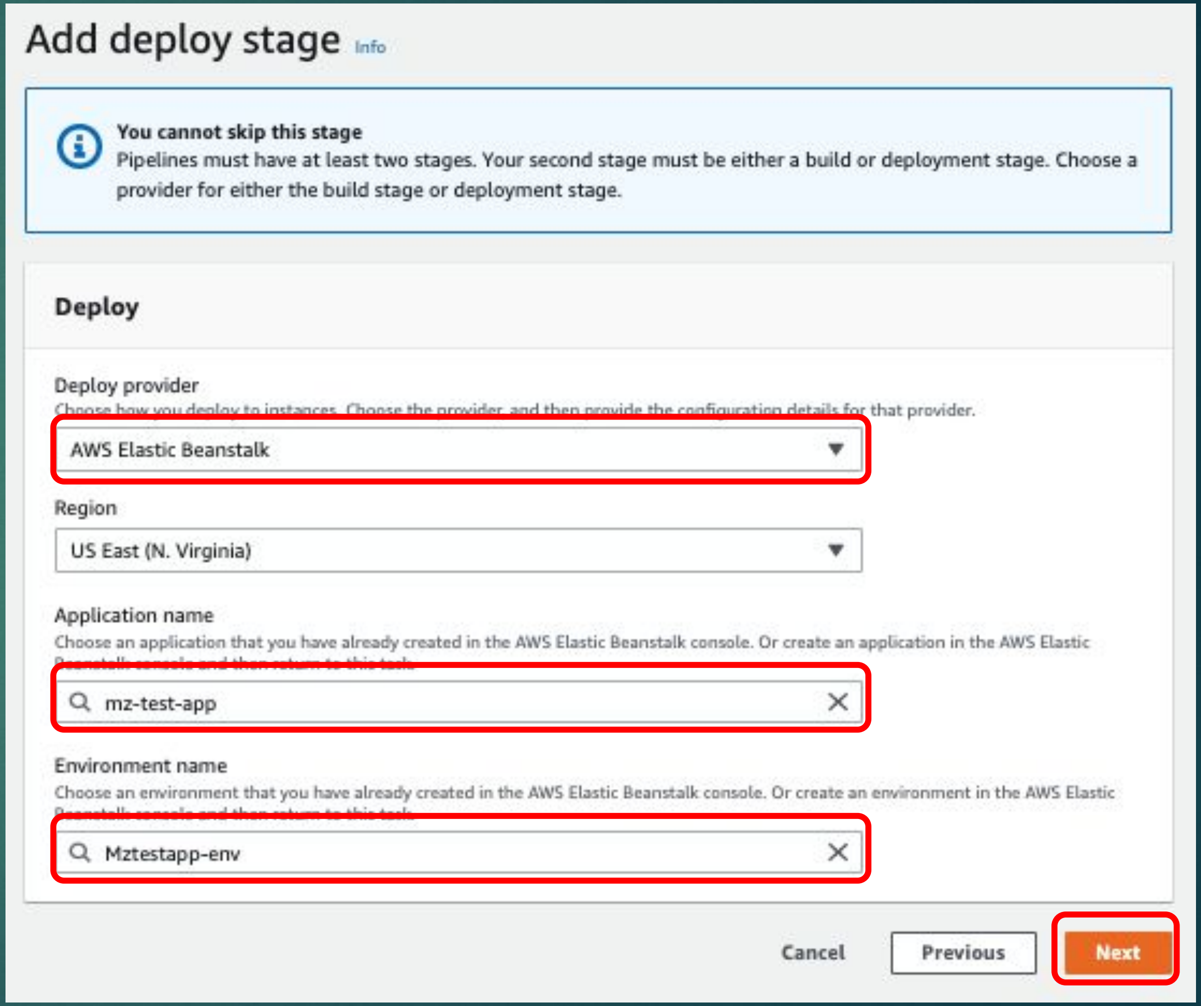


Skip build stage

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

Deploy Stage

- Under “Deploy provider” Select “AWS Elastic Beanstalk”
- Under “Application name” select the application you previously created
- Under “Environment name” select the environment that was previously created
- Click “Next”



Add deploy stage

You cannot skip this stage
Pipelines must have at least two stages. Your second stage must be either a build or deployment stage. Choose a provider for either the build stage or deployment stage.

Deploy

Deploy provider
Choose how you deploy to instances. Choose the provider, and then provide the configuration details for that provider.
AWS Elastic Beanstalk

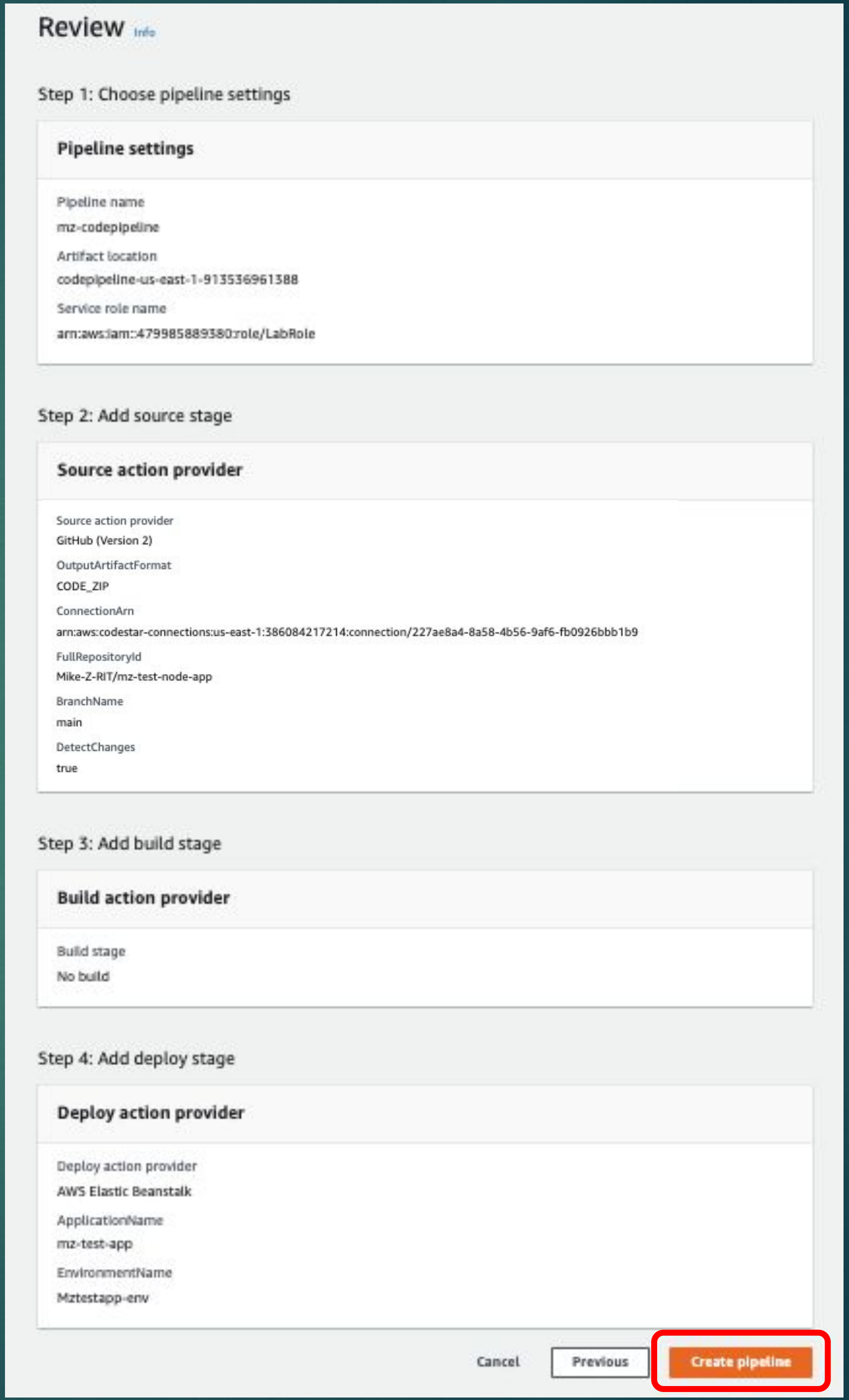
Region
US East (N. Virginia)

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.
mz-test-app

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.
Mztestapp-env

Create your Pipeline

- Click the "Create pipeline"



Review

Step 1: Choose pipeline settings

Pipeline settings

Pipeline name
mz-codepipeline

Artifact location
codepipeline-us-east-1-913536961388

Service role name
arn:aws:iam::479985889380:role/LabRole

Step 2: Add source stage

Source action provider

Source action provider
GitHub (Version 2)

OutputArtifactFormat
CODE_ZIP

ConnectionArn
arn:aws:codestar-connections:us-east-1:386084217214:connection/227ae8a4-8a58-4b56-9af6-fb0926bbb1b9

FullRepositoryId
Mike-Z-RIT/mz-test-node-app

BranchName
main

DetectChanges
true

Step 3: Add build stage

Build action provider

Build stage
No build

Step 4: Add deploy stage

Deploy action provider

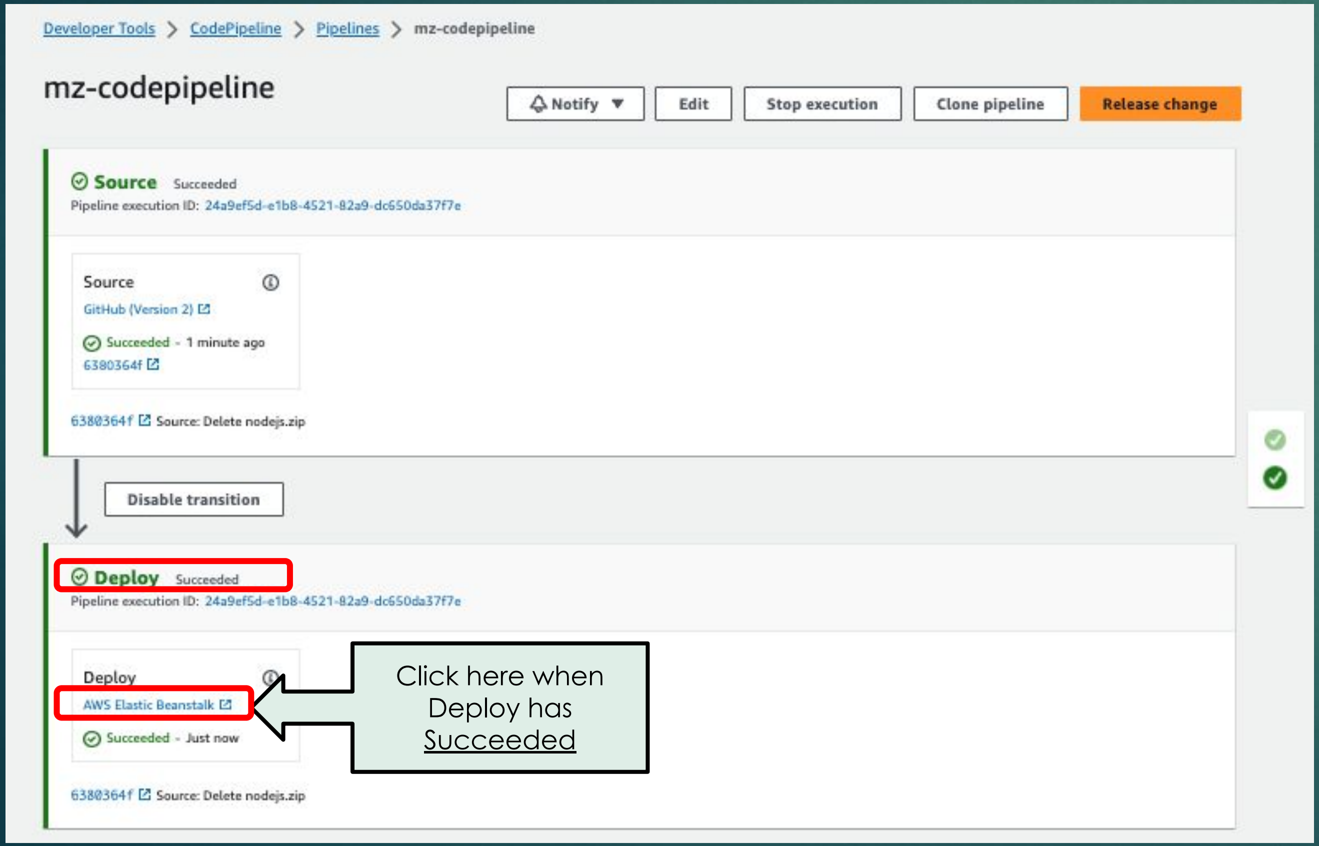
Deploy action provider
AWS Elastic Beanstalk

ApplicationName
mz-test-app

EnvironmentName
Mztestapp-env

Build and Deploy

- This will start the Build and Deploy process



mz-codepipeline

Source Succeeded
Pipeline execution ID: 24a9ef5d-e1b8-4521-82a9-dc650da37f7e

Source
GitHub (Version 2)
Succeeded - 1 minute ago
6380364f

6380364f Source: Delete nodejs.zip

Deploy Succeeded
Pipeline execution ID: 24a9ef5d-e1b8-4521-82a9-dc650da37f7e

Deploy
AWS Elastic Beanstalk
Succeeded - Just now

6380364f Source: Delete nodejs.zip

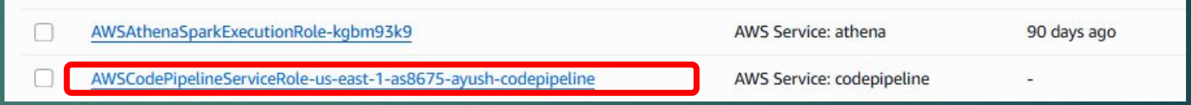
Click here when Deploy has Succeeded

This gets automatically triggered every time you modify the code in Github and commit to the master

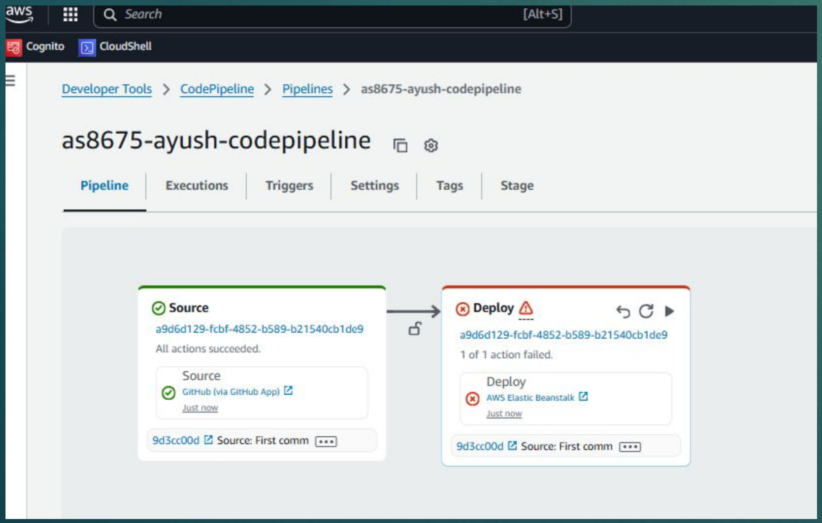
Once the source has been detected, the deploy process will deploy the updated code to your Node application

Build and Deploy

- If you see "elasticbeanstalk:CreateApplicationVersion permission missing", go to IAM → Roles → attach the policy to AWSCodePipelineServiceRole-<your-pipeline-name>
- Search for and attach the AdministratorAccess-AWSElasticBeanstalk policy.
- Go back to your pipeline and refresh and it should work.



□	AWSAthenaSparkExecutionRole-kgbm93k9	AWS Service: athena	90 days ago
□	AWSCodePipelineServiceRole-us-east-1-as8675-ayush-codepipeline	AWS Service: codepipeline	-

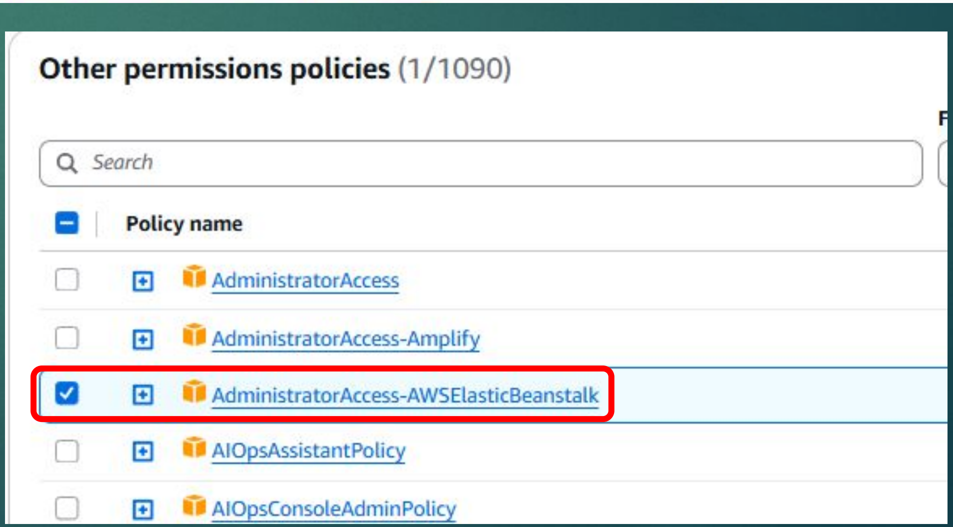


Developer Tools > CodePipeline > Pipelines > as8675-ayush-codepipeline

as8675-ayush-codepipeline

Source
a9d6d129-fcbf-4852-b589-b21540cb1de9
All actions succeeded.
Source
GitHub (via GitHub App)
Just now
9d3cc00d Source: First comm

Deploy
a9d6d129-fcbf-4852-b589-b21540cb1de9
1 of 1 action failed.
Deploy
AWS Elastic Beanstalk
Just now
9d3cc00d Source: First comm

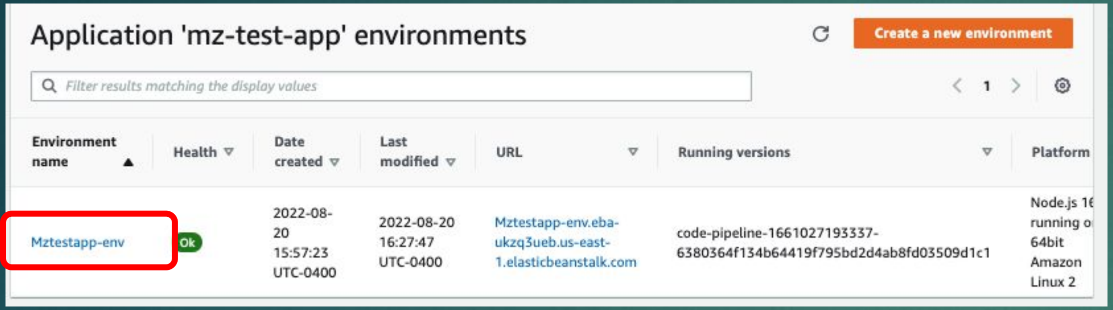


Other permissions policies (1/1090)

	Policy name
□	AdministratorAccess
□	AdministratorAccess-Amplify
☑	AdministratorAccess-AWSElasticBeanstalk
□	AIOpsAssistantPolicy
□	AIOpsConsoleAdminPolicy

Create your Pipeline

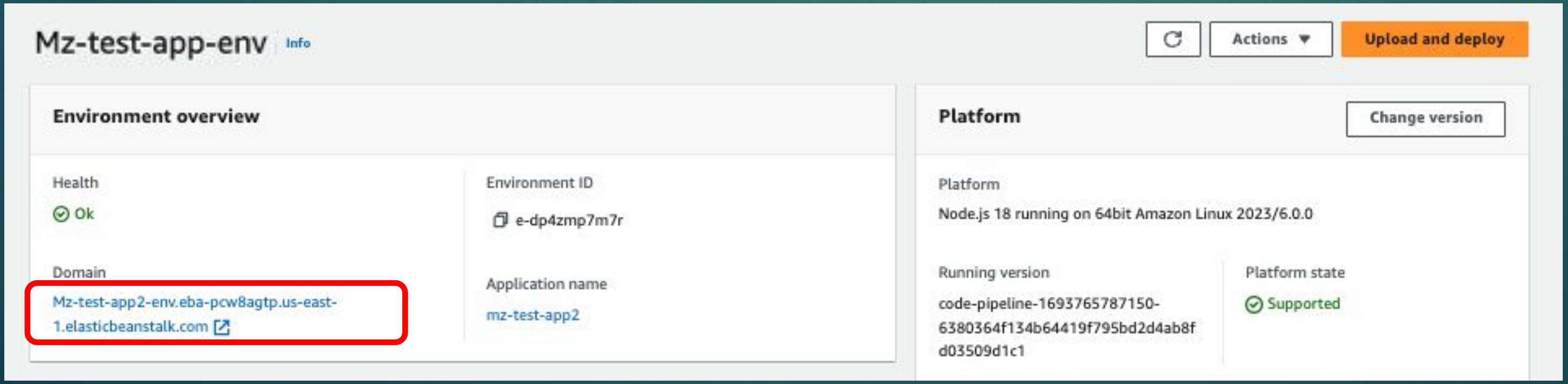
- Click the link for your web application



Application 'mz-test-app' environments

Environment name	Health	Date created	Last modified	URL	Running versions	Platform
Mztestapp-env	Ok	2022-08-20 15:57:23 UTC-0400	2022-08-20 16:27:47 UTC-0400	Mztestapp-env.eba-ukzq3ueb.us-east-1.elasticbeanstalk.com	code-pipeline-1661027193337-6380364f134b64419f795bd2d4ab8fd03509d1c1	Node.js 16 running o 64bit Amazon Linux 2

- On the Dashboard, click the URL



Mz-test-app-env

Environment overview

Health
Ok

Environment ID
e-dp4zmp7m7r

Domain
Mz-test-app2-env.eba-pcw8agtp.us-east-1.elasticbeanstalk.com

Application name
mz-test-app2

Platform

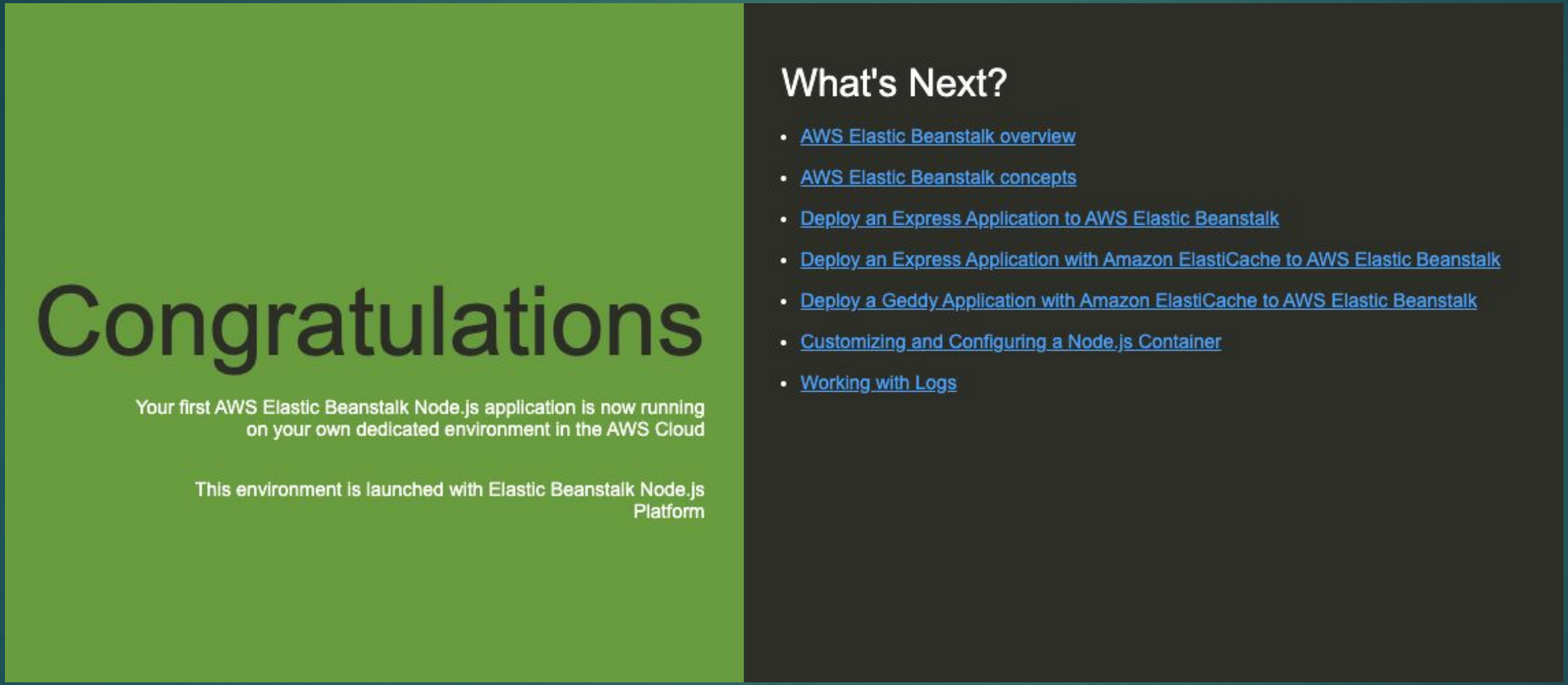
Platform
Node.js 18 running on 64bit Amazon Linux 2023/6.0.0

Running version
code-pipeline-1693765787150-6380364f134b64419f795bd2d4ab8fd03509d1c1

Platform state
Supported

Test your Web Application

- The sample application (from Github) has been deployed and is running



Congratulations

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

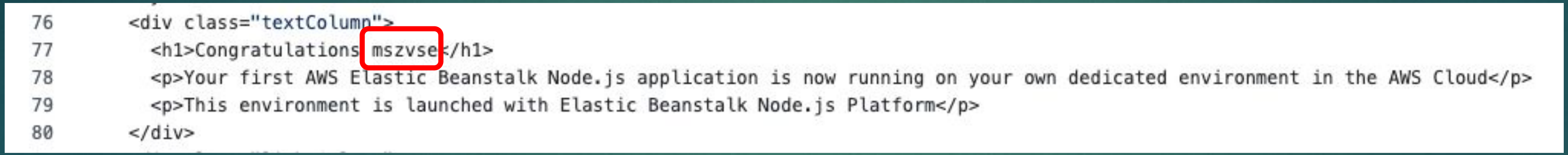
What's Next?

- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy an Express Application to AWS Elastic Beanstalk
- Deploy an Express Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Deploy a Geddy Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

- Congratulations, but you are <u>not</u> done yet
- Now we must make a change to trigger the pipeline to update

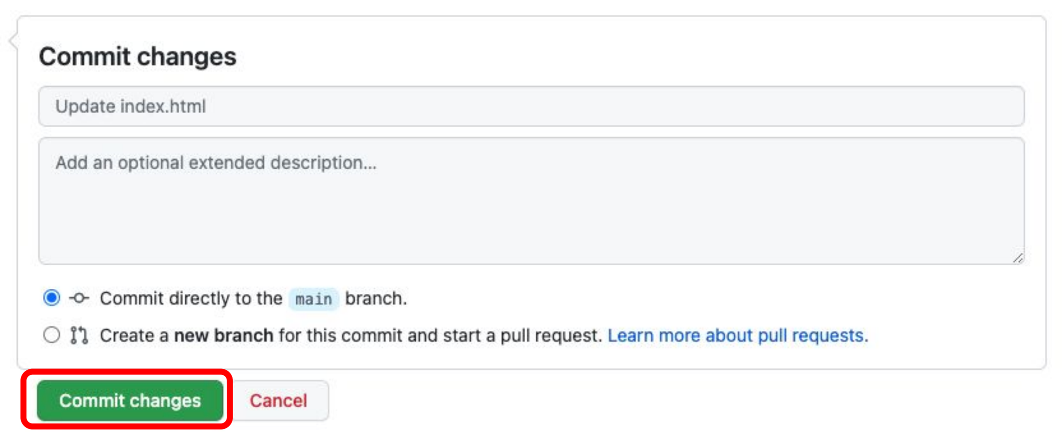
Modify your code and commit

- Next, make a change in code
  - You can either do this on your PC or directly in GitHub
- In the index.html, add your RIT ID in the title after ”Congratulations”



```
    <div class="textColumn">
      <h1>Congratulations mszvse</h1>
      <p>Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud</p>
      <p>This environment is launched with Elastic Beanstalk Node.js Platform</p>
    </div>
```

- Either push your change (from your PC) or click “Commit changes” (below)



Commit changes

Update index.html

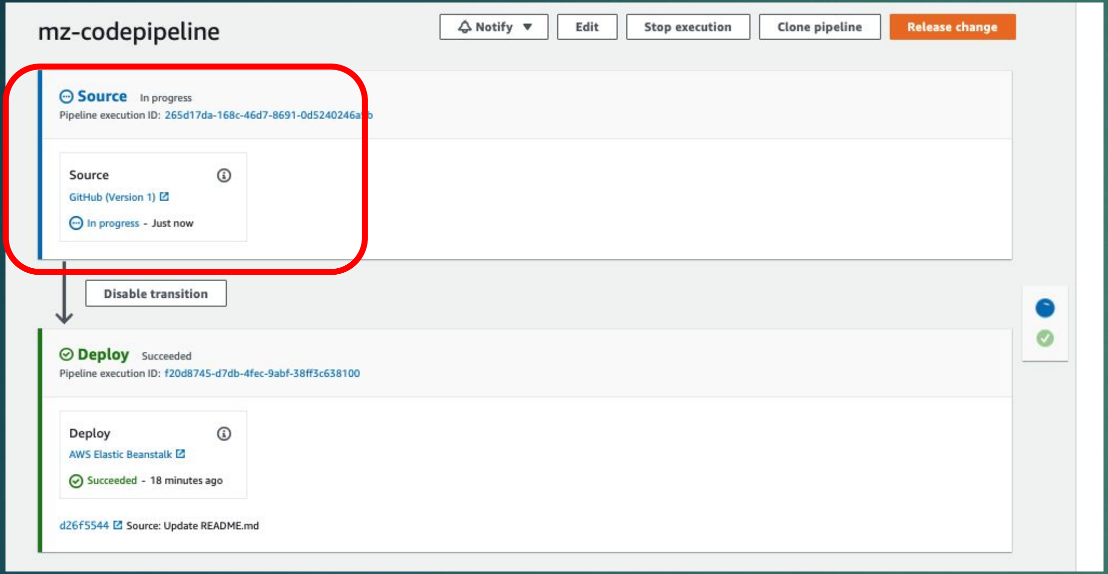
Add an optional extended description...

● Commit directly to the `main` branch.

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

Redeploy your code

- When you push your change, the "Source" will indicate "In Progress" followed by "Deploy" stage



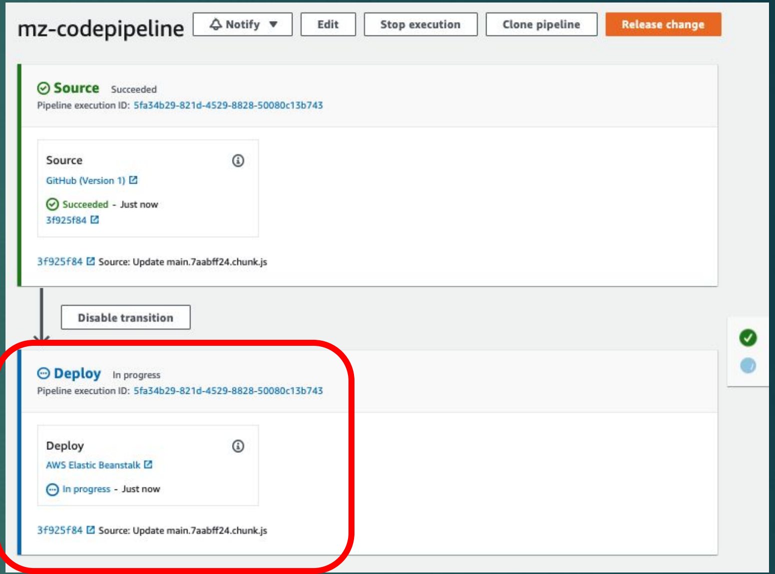
Source In progress
Pipeline execution ID: 265d17da-168c-46d7-8691-0d5240246a[illegible]b

Source
GitHub (Version 1)
In progress - Just now

Deploy Succeeded
Pipeline execution ID: f20d8745-d7db-4fec-9abf-38ff3c638100

Deploy
AWS Elastic Beanstalk
Succeeded - 18 minutes ago

d26f5544 Source: Update README.md



Source Succeeded
Pipeline execution ID: 5fa34b29-821d-4529-8828-50080c13b743

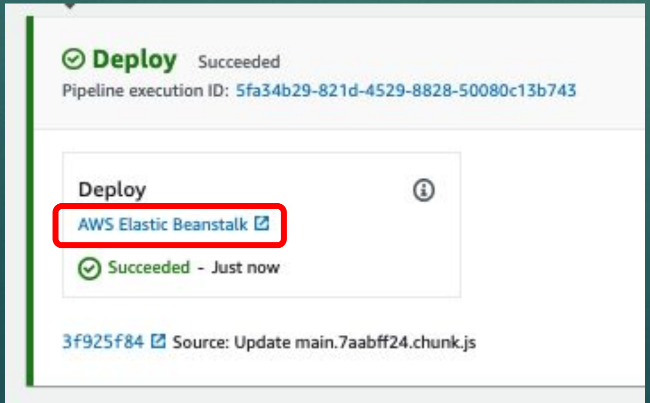
Source
GitHub (Version 1)
Succeeded - Just now
3f925f84

3f925f84 Source: Update main.7aabff24.chunk.js

Deploy In progress
Pipeline execution ID: 5fa34b29-821d-4529-8828-50080c13b743

Deploy
AWS Elastic Beanstalk
In progress - Just now

3f925f84 Source: Update main.7aabff24.chunk.js



Deploy Succeeded
Pipeline execution ID: 5fa34b29-821d-4529-8828-50080c13b743

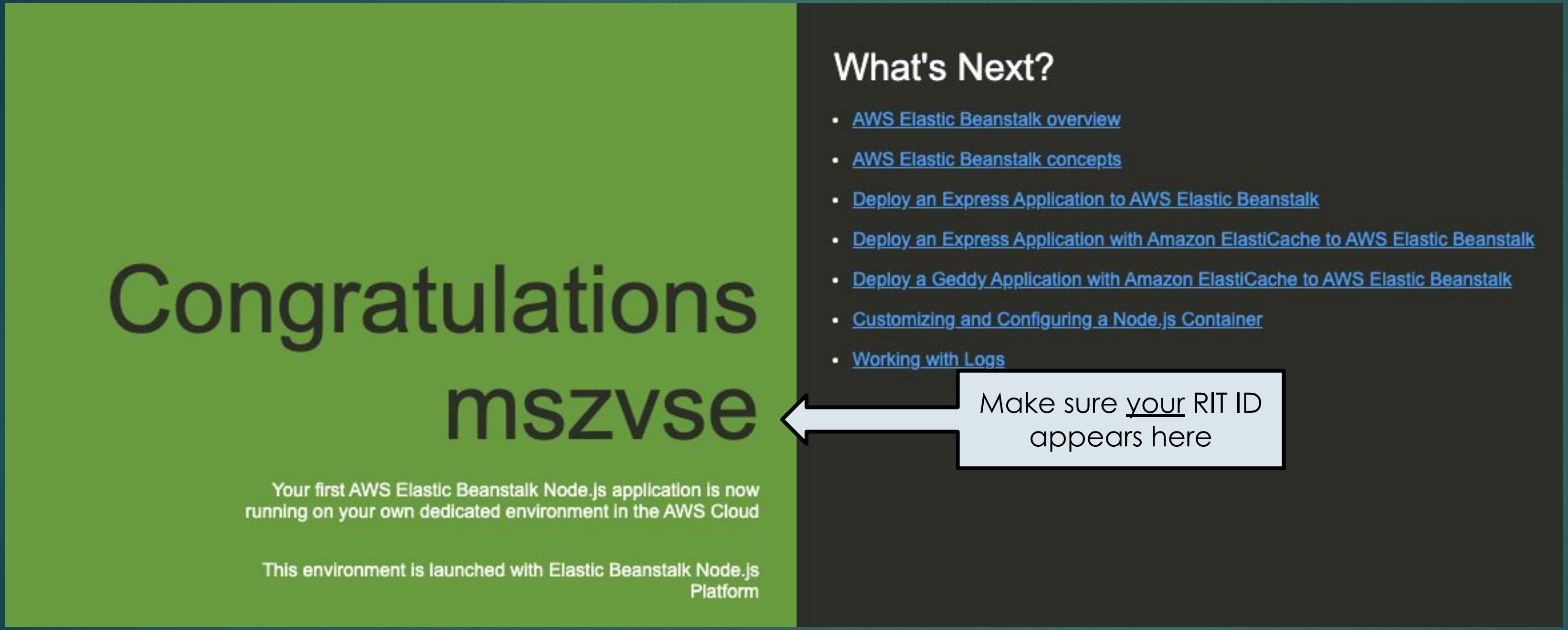
Deploy
AWS Elastic Beanstalk
Succeeded - Just now

3f925f84 Source: Update main.7aabff24.chunk.js

When you see a Succeeded message under Deploy, click the "AWS Elastic Beanstalk" link

Test your Web Application – Deliverable #1

- You are taken to updated web page of your application
- You now have a fully functioning continuous deployment pipeline!



Congratulations
mszvse

Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

This environment is launched with Elastic Beanstalk Node.js Platform

What's Next?

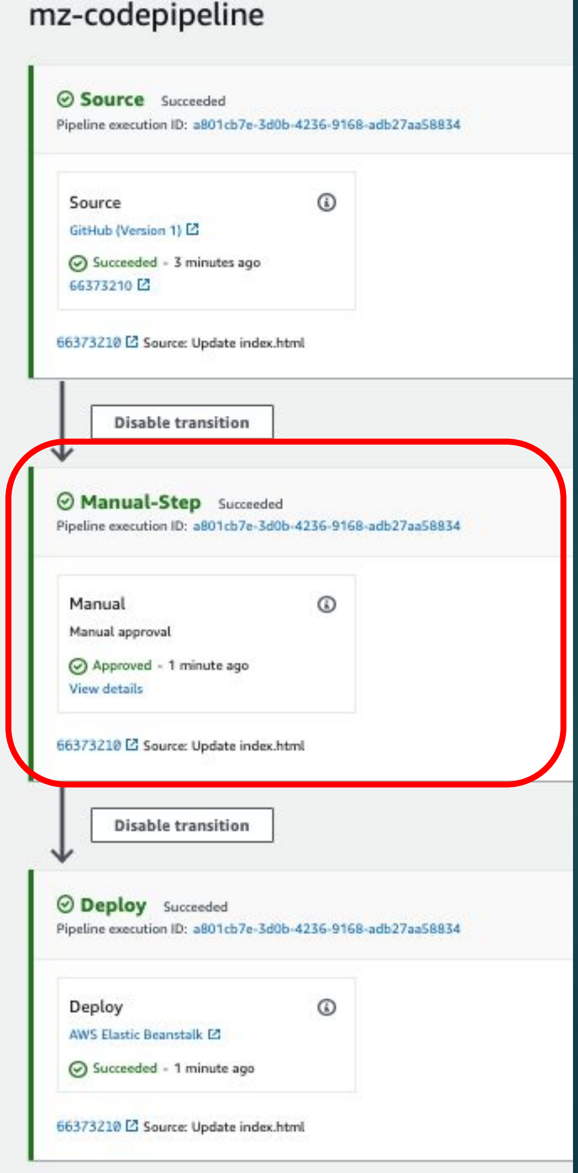
- AWS Elastic Beanstalk overview
- AWS Elastic Beanstalk concepts
- Deploy an Express Application to AWS Elastic Beanstalk
- Deploy an Express Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Deploy a Geddy Application with Amazon ElastiCache to AWS Elastic Beanstalk
- Customizing and Configuring a Node.js Container
- Working with Logs

Make sure your RIT ID appears here

- Take a screenshot of web page with your update and upload to Assignments > Activity #14 - Create a Continuous Delivery Pipeline

Test your Web Application – Deliverable #2

- Next, modify your pipeline to Continuous Delivery
- To do this requires adding a Manual Review Step to your Pipeline
- When you have working, take a screenshot similar to the right
  - Note: You will need to make a code change to re-trigger the pipeline
- Submit screenshot to Assignments > Activity #14 - Create a DevOps Pipeline



mz-codepipeline

Source Succeeded
Pipeline execution ID: a801cb7e-3d0b-4236-9168-adb27aa58834

Source
GitHub (Version 1)
Succeeded - 3 minutes ago
66373210

66373210 Source: Update index.html

Manual-Step Succeeded
Pipeline execution ID: a801cb7e-3d0b-4236-9168-adb27aa58834

Manual
Manual approval
Approved - 1 minute ago
View details

66373210 Source: Update index.html

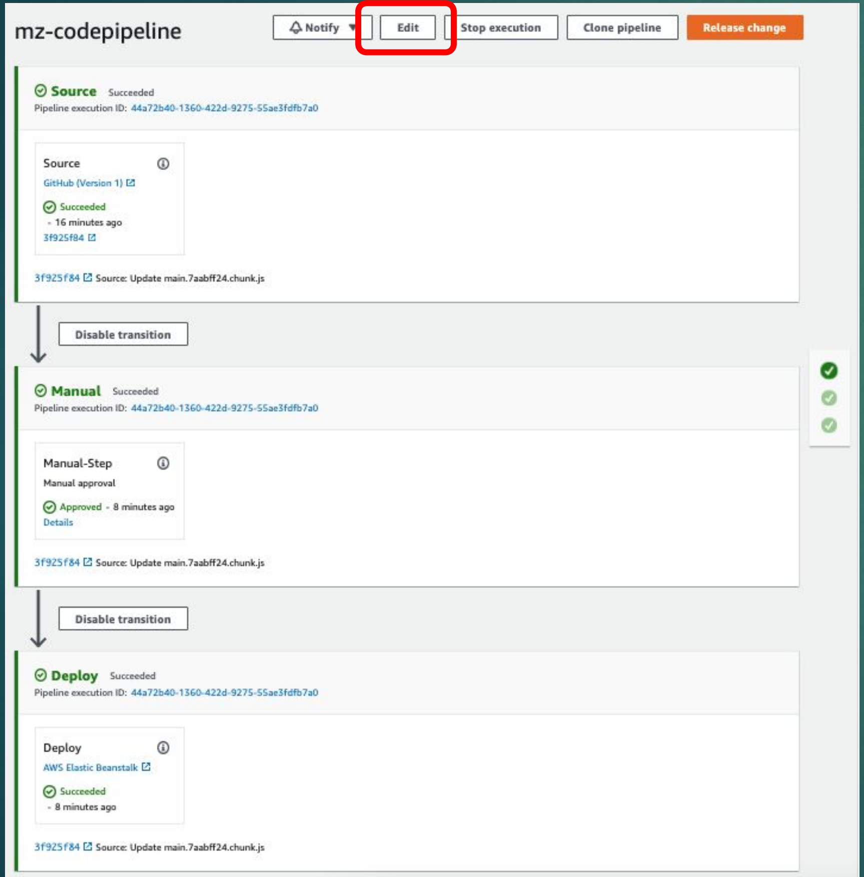
Deploy Succeeded
Pipeline execution ID: a801cb7e-3d0b-4236-9168-adb27aa58834

Deploy
AWS Elastic Beanstalk
Succeeded - 1 minute ago

66373210 Source: Update index.html

Cleanup Pipeline

- Go back to CodePipeline and click “Edit”



mz-codepipeline

✓ Source Succeeded
Pipeline execution ID: 44a72b40-1360-422d-9275-55ae3fdfb7a0

Source
GitHub (Version 1)
✓ Succeeded - 16 minutes ago
3f925f84

3f925f84 Source: Update main.7aabff24.chunk.js

✓ Manual Succeeded
Pipeline execution ID: 44a72b40-1360-422d-9275-55ae3fdfb7a0

Manual-Step
Manual approval
✓ Approved - 8 minutes ago
Details

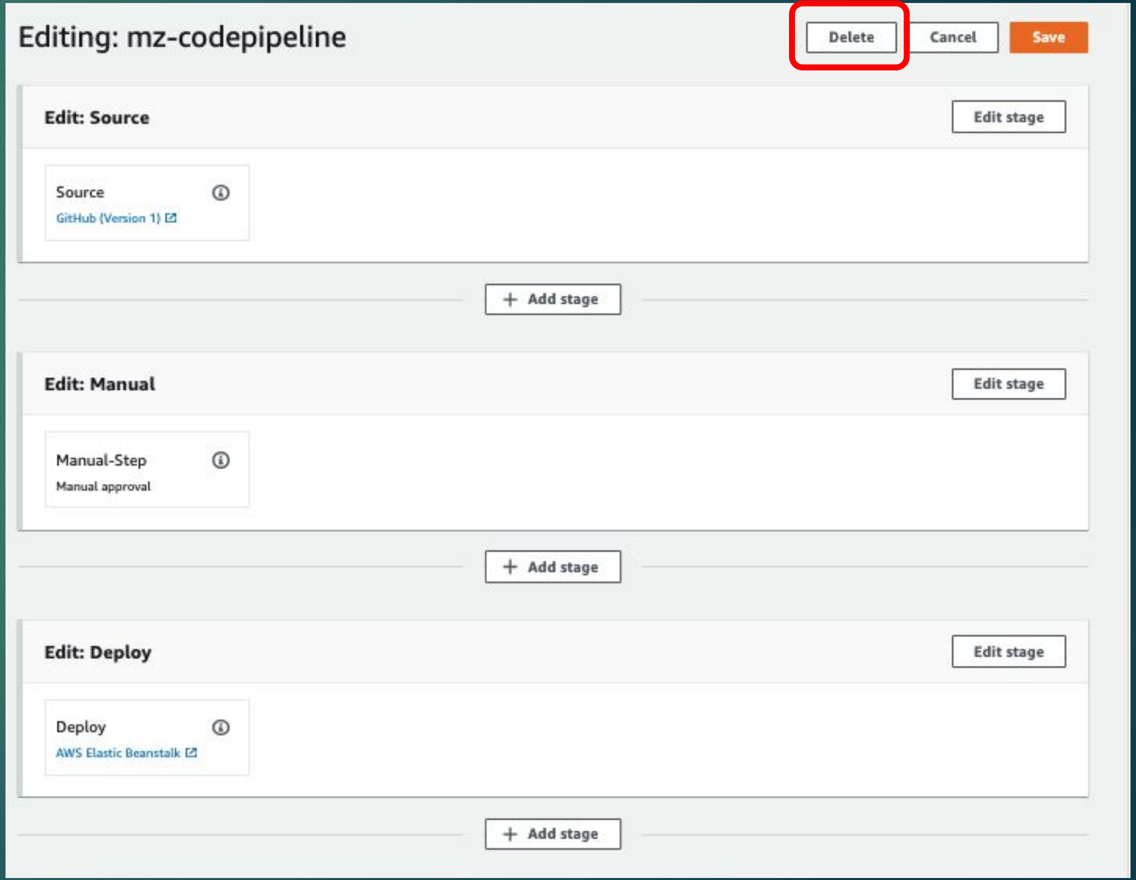
3f925f84 Source: Update main.7aabff24.chunk.js

✓ Deploy Succeeded
Pipeline execution ID: 44a72b40-1360-422d-9275-55ae3fdfb7a0

Deploy
AWS Elastic Beanstalk
✓ Succeeded - 8 minutes ago

3f925f84 Source: Update main.7aabff24.chunk.js

- Click ”Delete” and confirm



Editing: mz-codepipeline

Edit: Source

Source
GitHub (Version 1)

Edit: Manual

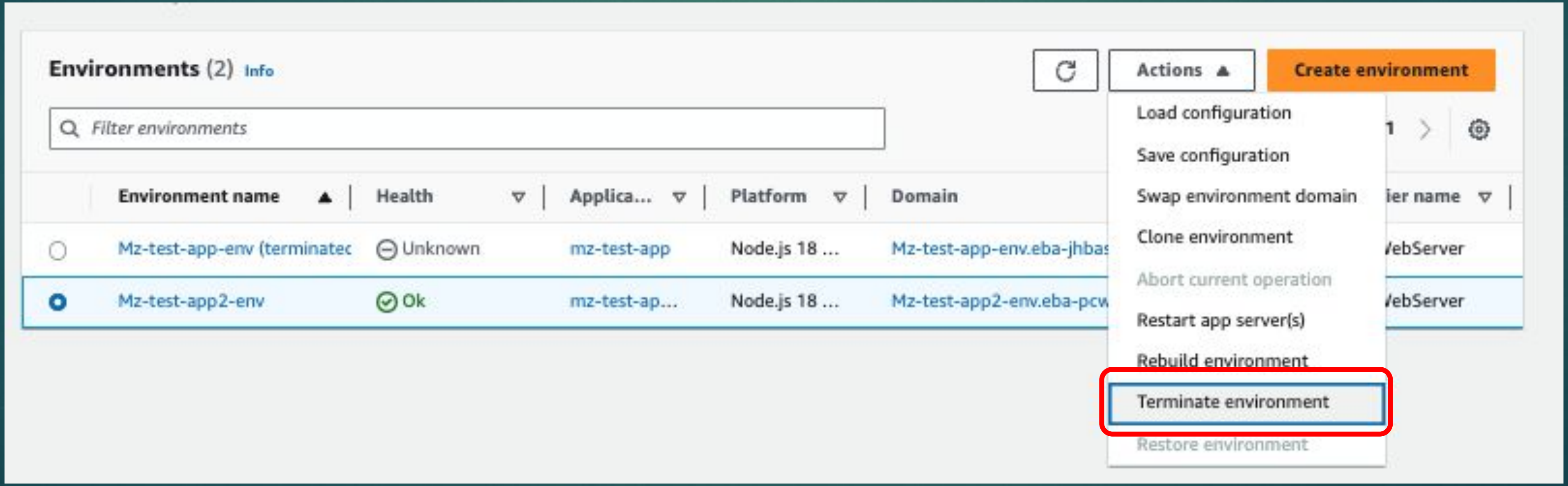
Manual-Step
Manual approval

Edit: Deploy

Deploy
AWS Elastic Beanstalk

Cleanup Beanstalk

- Go to the Beanstalk console under “Environments” and select your environment name
- Select Actions > Terminate Environment
- Verify by typing the name of the environment and click “Terminate”

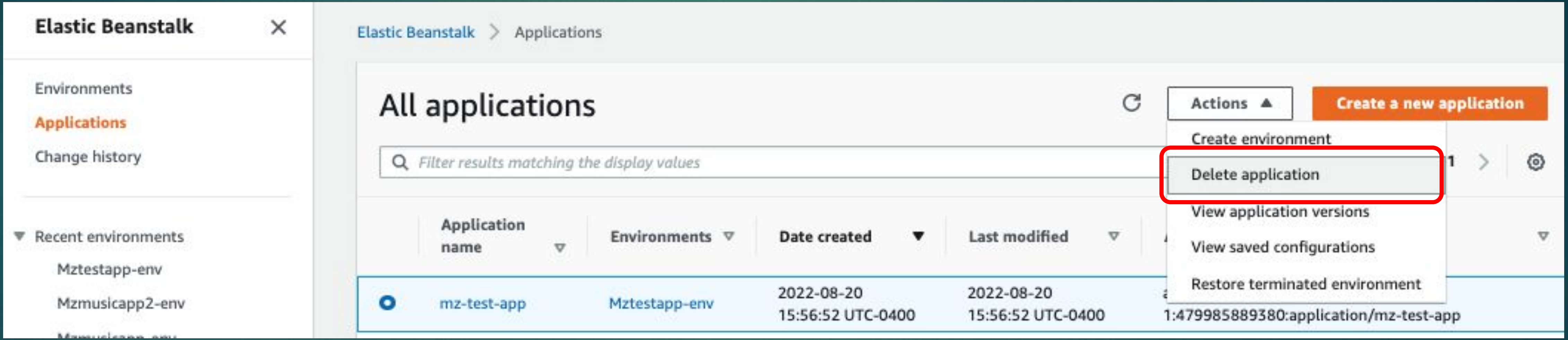


Environments (2) Info

	Environment name ▲	Health	Applica...	Platform	Domain	ier name
○	Mz-test-app-env (terminatec	⊖ Unknown	mz-test-app	Node.js 18 ...	Mz-test-app-env.eba-jhbas	/ebServer
●	Mz-test-app2-env	✓ Ok	mz-test-ap...	Node.js 18 ...	Mz-test-app2-env.eba-pcw	/ebServer

Cleanup Beanstalk

- Under “Applications” select your application name
- Select “Actions > Delete application”
- Verify by typing the name of the application and click “Delete”



All applications

	Application name	Environments	Date created	Last modified	
●	mz-test-app	Mztestapp-env	2022-08-20 15:56:52 UTC-0400	2022-08-20 15:56:52 UTC-0400	1:479985889380:application/mz-test-app

- You are done!