

Create a DevOps Pipeline

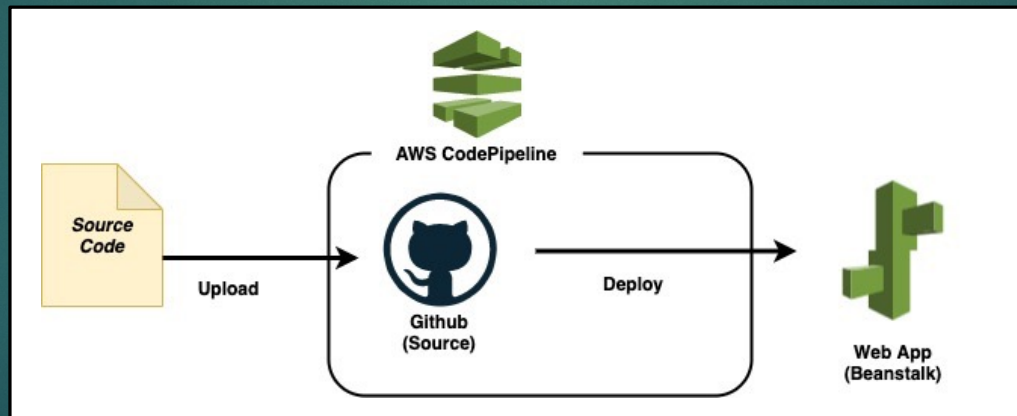
SWEN 514/614: Engineering Cloud Software Systems

Department of Software Engineering
Rochester Institute of Technology



Overview

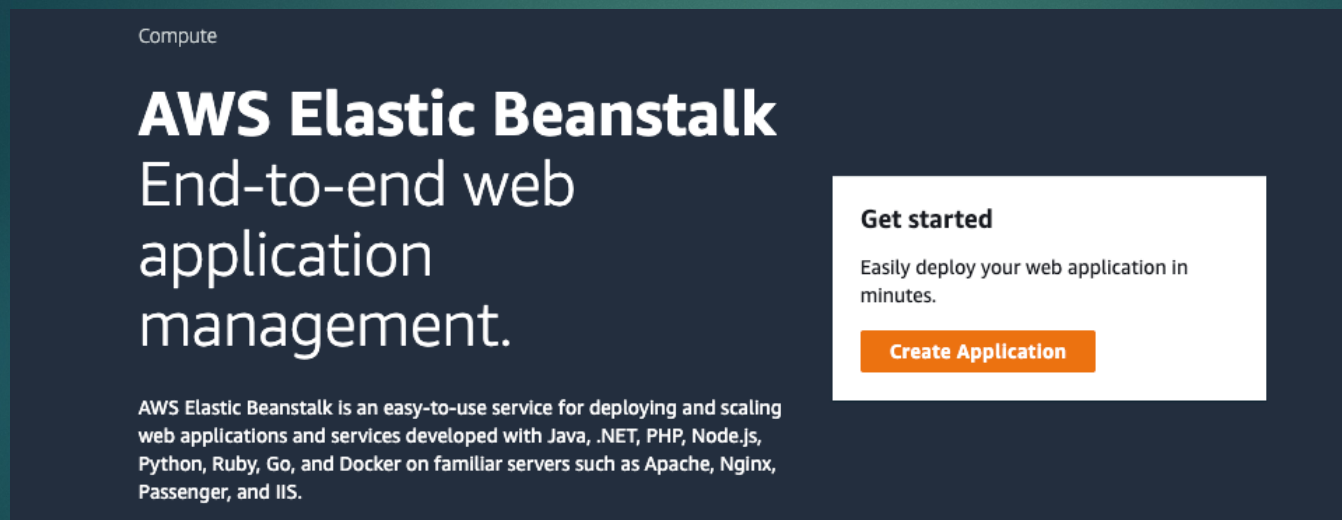
- ▶ In this activity, you create a Continuous Deployment pipeline that deploys a live sample web application
- ▶ This will use AWS CodePipeline to deploy code from your Github account to an Elastic Beanstalk NodeJS application
- ▶ Below is an illustration of the process



- ▶ 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”



The screenshot shows the AWS Elastic Beanstalk console page. At the top left, the word "Compute" is visible. The main heading is "AWS Elastic Beanstalk" in large white font, followed by the subtitle "End-to-end web application management." in a slightly smaller white font. Below this, a paragraph of text describes the service: "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." To the right of this text is a white box with the heading "Get started" and the text "Easily deploy your web application in minutes." Below this text is an orange button labeled "Create Application".

- ▶ 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

Cancel **Next**

Create an app in Elastic Beanstalk

- ▶ Under “Service role”, select “Create and use new service role”
 - ▶ Note: If you need to run this step again, the service role will be available so select “Use an existing service role” the next time since the role already is available
- ▶ 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 “View permission details” 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

☒ Create and use new service role
☐ Use an existing service role

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

aws-elasticbeanstalk-service-role

[View permission details](#)

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

MZ-AWS-RIT

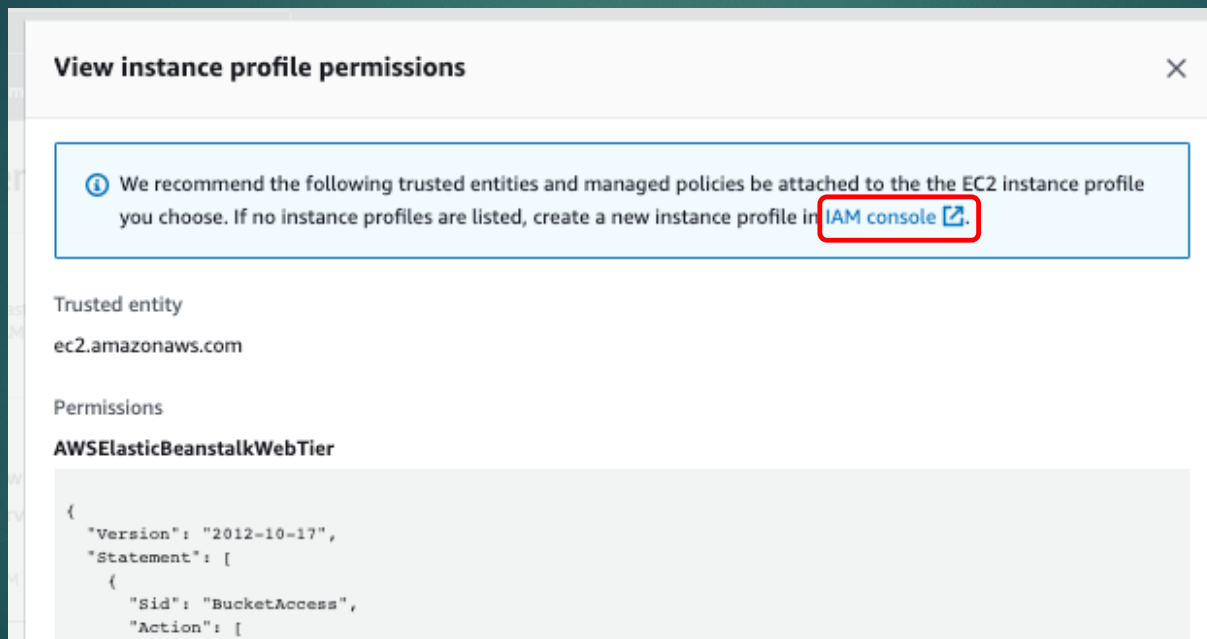
EC2 instance profile
Choose an IAM instance profile with managed policies that allow your EC2 instances to perform required operations.

[View permission details](#)

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

Create an app in Elastic Beanstalk

- ▶ We need to create a new instance profile to attach to the EC2 instance
- ▶ To do this, click the “IAM console”



IAM Dashboard

- ▶ On the IAM Dashboard, we are going to create a new instance profile (role)
- ▶ Click the “Roles” on the left

The screenshot shows the AWS IAM Dashboard. On the left, the navigation menu is visible with the 'Roles' option highlighted by a red rectangle. The main dashboard area displays 'Security recommendations' and 'IAM resources'.

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

- User groups
- Users
- Roles**
- Policies
- Identity providers
- Account settings

Access reports

- Access analyzer
- Archive rules
- Analyzers

IAM Dashboard

Security recommendations 1

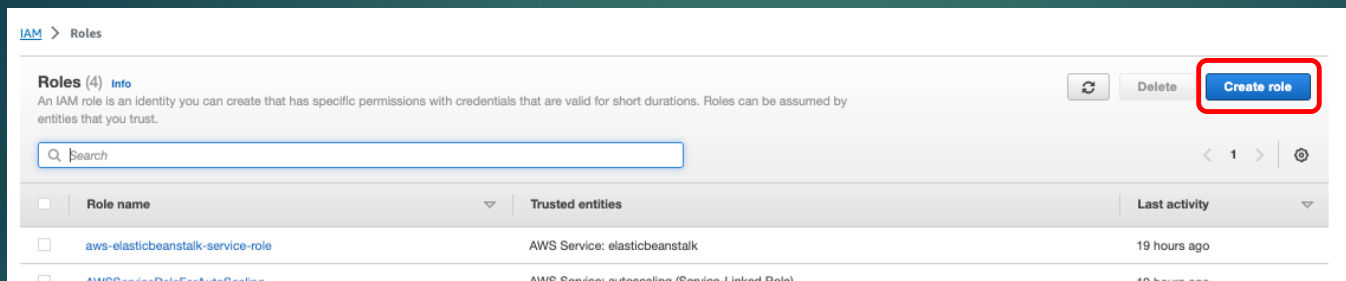
- Add MFA for root user**
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account. **Add MFA**
- Root user has no active access keys**
Using access keys attached to an IAM user instead of the root user improves security.

IAM resources
Resources in this AWS Account

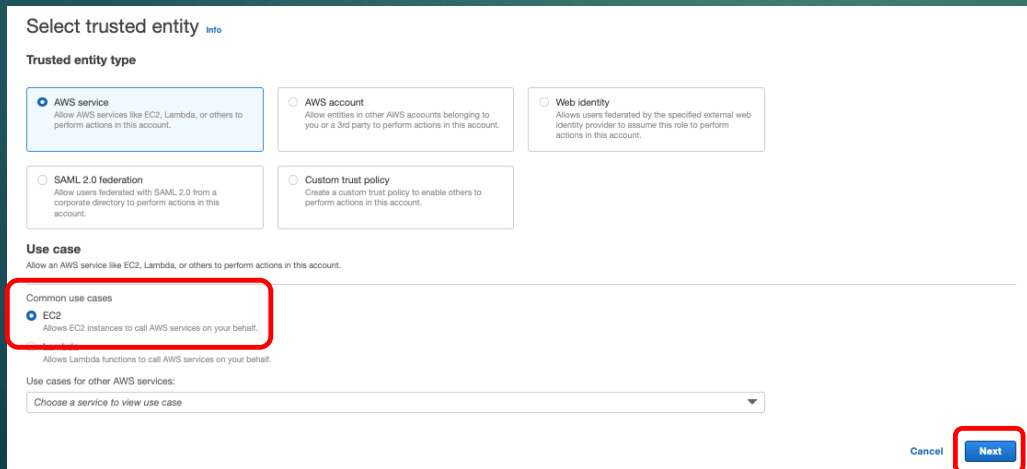
User groups	Users	Roles	Policies	Identity providers
0	0	4	0	0

Create IAM Role

► Select “Create role”



► On the next screen, select “EC2” under “Use case” and click “Next”



Create IAM Role

- ▶ In the search box, enter “AWSElasticBeanstalk” and hit Enter
- ▶ A subset of permissions is returned
- ▶ Select the top 3 permissions (see below)
- ▶ Click “Next”

Add permissions [Info](#)

Permissions policies (Selected 3/875) [Info](#)
Choose one or more policies to attach to your new role.

14 matches

AWSElasticBeanstalk X Clear filters

<input type="checkbox"/>	Policy name ?	Type	Description
<input checked="" type="checkbox"/>	AWSElasticBeanstalkWebTier	AWS m...	Provide the instances in your web server environment access to upload log files to Amazon S3.
<input checked="" type="checkbox"/>	AWSElasticBeanstalkWorkerTier	AWS m...	Provide the instances in your worker environment access to upload log files to Amazon S3, to use Amazon SQS to monitor your app...
<input checked="" type="checkbox"/>	AWSElasticBeanstalkMulticontainerDocker	AWS m...	Provide the instances in your multicontainer Docker environment access to use the Amazon EC2 Container Service to manage cont...
<input type="checkbox"/>	AWSElasticBeanstalkEnhancedHealth	AWS m...	AWS Elastic Beanstalk Service policy for Health Monitoring system

▶ **Set permissions boundary - optional** [Info](#)
Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous **Next**

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

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.

Cancel Previous **Create role**

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. [View role](#)

[IAM](#) > Roles

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.

[Refresh](#) [Delete](#) [Create role](#)

<input type="checkbox"/>	Role name	Trusted entities	Last activity
<input type="checkbox"/>	aws-elasticbeanstalk-service-role	AWS Service: elasticbeanstalk	19 hours ago
<input type="checkbox"/>	AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	19 hours ago
<input type="checkbox"/>	AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
<input type="checkbox"/>	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
<input type="checkbox"/>	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 “Skip to review”
- ▶ On the Review screen, click “Submit” to create your new application

Configure service access [Info](#)

Service access
IAM roles, assumed by Elastic Beanstalk as a service role, and EC2 instance profiles allow Elastic Beanstalk to create and manage your environment. Both the IAM role and instance profile must be attached to IAM managed policies that contain the required permissions. [Learn more](#)

Service role
☒ Create and use new service role
☐ Use an existing service role

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

[View permission details](#)

EC2 key pair
Select an EC2 key pair to securely log in to your EC2 instances. [Learn more](#)

[Refresh](#)

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

[Refresh](#)

[View permission details](#)

Cancel [Skip to review](#) Previous [Next](#)

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.

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

Mz-test-app-env Info

[Refresh](#) [Actions](#) [Upload and deploy](#)

Environment overview

Health	Environment ID
⌚ Pending	📁 e-tafr9v2jxc
Domain	Application name
–	mz-test-app

Platform

[Change version](#)

Platform	
Node.js 18 running on 64bit Amazon Linux 2023/6.0.0	
Running version	Platform state
–	🟢 Supported

[Events](#) [Health](#) [Logs](#) [Monitoring](#) [Alarms](#) [Managed updates](#) [Tags](#)

Events (3) Info

[Refresh](#) [Previous](#) [1](#) [Next](#) [Settings](#)

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”

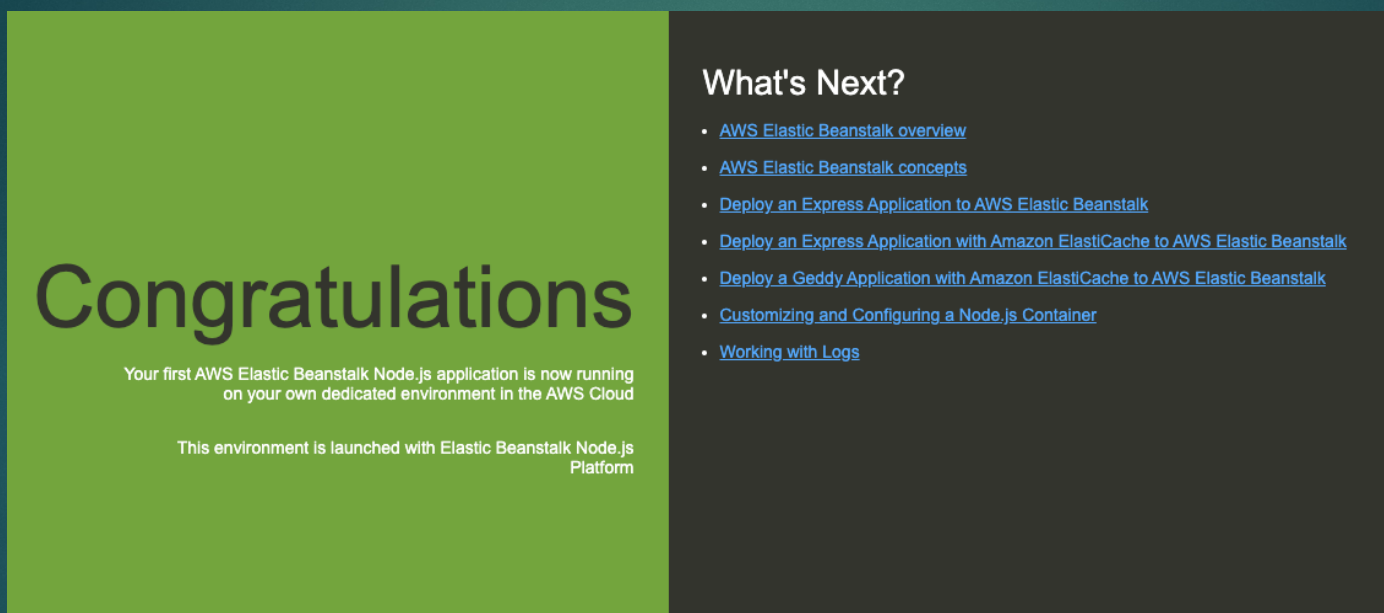
The screenshot shows the AWS Elastic Beanstalk console for an environment named 'Mz-test-app-env'. The 'Environment overview' section displays the health status as 'Ok', the domain link 'Mz-test-app2-env.eba-pcw8agtp.us-east-1.elasticbeanstalk.com', and other details like Environment ID and Application name. A red box with the text 'Click here' and an arrow points to the domain link. The 'Platform' section shows 'Node.js 18 running on 64bit Amazon Linux 2023/6.0.0'. The 'Events' section shows a list of events, including 'Successfully launched environment: Mz-test-app2-env'.

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: aws-eb-e-dp4zmp7m7r-stack-AWSECGroup-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



- ▶ 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.

Create repository

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

Set up in Desktop or **HTTPS** **SSH** <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.

Import code

Create a Repo (example)

- ▶ Here is an example of pushing the files to Github following the steps from the previous slide
- ▶ When complete, verify your repo has the files below

```
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 %
```

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

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

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

The screenshot shows the AWS CodePipeline console interface. On the left is a sidebar with a 'Developer Tools' header and a 'CodePipeline' sub-header. Below this, there are several expandable sections: 'Source • CodeCommit', 'Artifacts • CodeArtifact', 'Build • CodeBuild', 'Deploy • CodeDeploy', and 'Pipeline • CodePipeline'. The 'Pipeline • CodePipeline' section is expanded, showing 'Getting started' and 'Pipelines' (highlighted in orange), along with a 'Settings' link. The main content area has a breadcrumb trail 'Developer Tools > CodePipeline > Pipelines'. Below this, there's a 'Pipelines' section with an 'Info' link, a refresh button, a 'Notify' button with a dropdown arrow, and buttons for 'View history', 'Release change', 'Delete pipeline', and 'Create pipeline' (which is highlighted with a red rectangle). A search bar is also present. Below the buttons is a table with columns: 'Name', 'Most recent execution', 'Latest source revisions', and 'Last executed'. The table is currently empty, displaying 'No results' and 'There are no results to display.'

Developer Tools **CodePipeline**

- ▶ Source • CodeCommit
- ▶ Artifacts • CodeArtifact
- ▶ Build • CodeBuild
- ▶ Deploy • CodeDeploy
- ▼ Pipeline • CodePipeline
 - Getting started
 - Pipelines**
 - Settings

Developer Tools > CodePipeline > Pipelines

Pipelines Info Refresh Notify ▼ View history Release change Delete pipeline **Create pipeline**

Name	Most recent execution	Latest source revisions	Last executed
No results			
There are no results to display.			

Create your Pipeline

- ▶ Enter name for “Pipeline name”
- ▶ Click “Next”

Choose pipeline settings [Info](#)

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**

Cancel **Next**

Pipeline Source

- ▶ Under “Source Provider”, Select “Github (Version 2)”
- ▶ 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

Add source stage [info](#)

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 step.

[Connect to GitHub](#)

Repository name
Choose a repository in your GitHub account.

<account>/<repository-name>

Branch name
Choose a branch of the repository.

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.

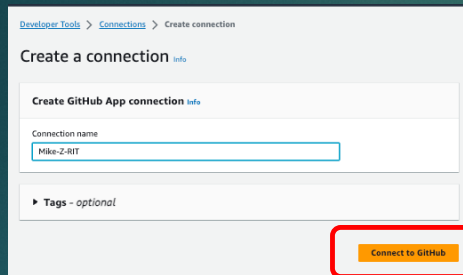
Output artifact format
Choose the output artifact format.

☒ **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.

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](#)

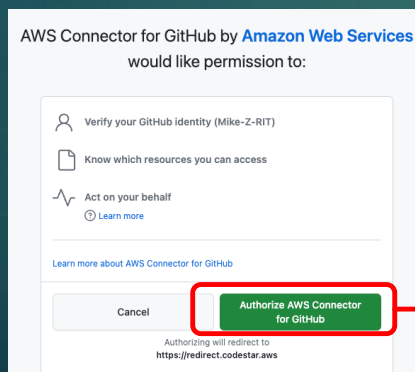
Create GitHub App connection [Info](#)

Connection name
Mike-Z-RIT

Tags - optional

Connect to Github

#2 - Click the “Authorize AWS Connector for Github”



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

- Verify your GitHub Identity (Mike-Z-RIT)
- Know which resources you can access
- Act on your behalf

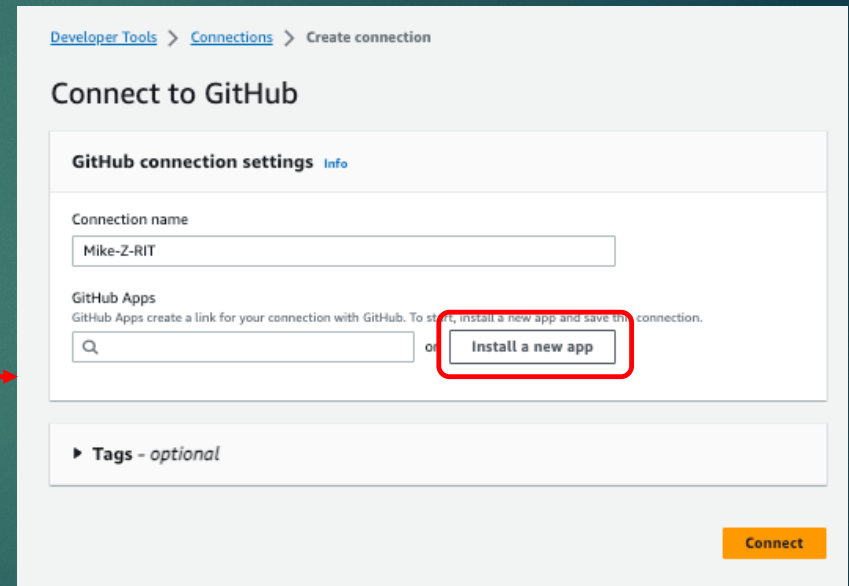
[Learn more](#)

[Learn more about AWS Connector for GitHub](#)

Cancel Authorize AWS Connector for GitHub

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

#3 - Click “Install a new app”



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 the connection.

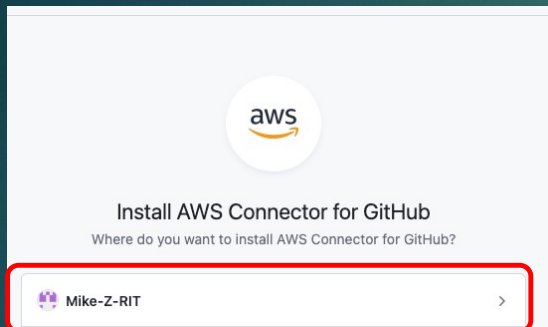
Search or [Install a new app](#)

Tags - optional

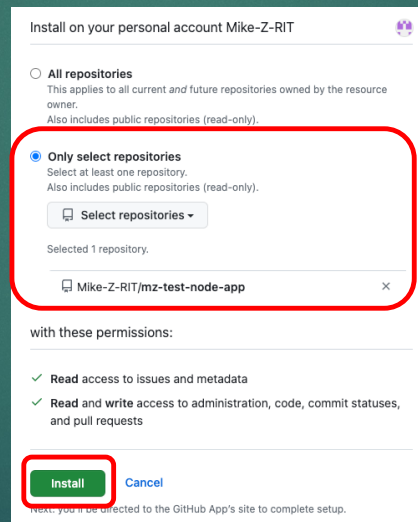
Connect

Connect Pipeline to Github

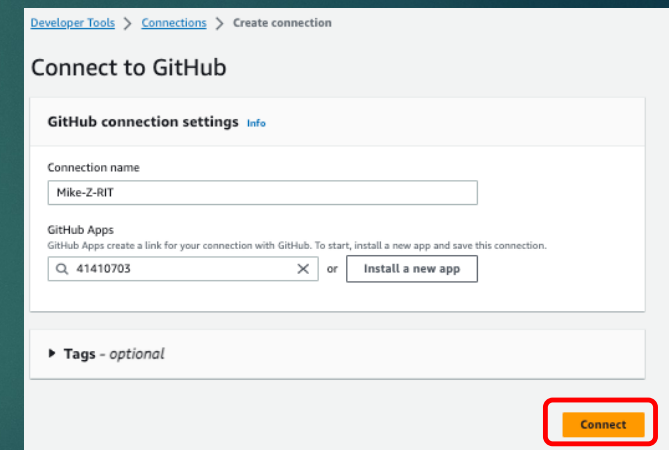
#1- Select your Github account



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



#3 - Click "Connect" to connect AWS to your Github account



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.

am:aws:codestar-connections:us-east-1:386084217214:connection/227ae8a X 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-RT/mz-test-node-app X

Branch name
Choose a branch of the repository

main X

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.

Cancel Previous **Next**

Build Stage

- ▶ Click the “Skip build stage” and confirm by clicking “Skip”

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 Info

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.

Q mz-test-app X

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.

Q Mztestapp-env X

Cancel Previous **Next**

Create your Pipeline

► Click the “Create pipeline”

27

Review [Info](#)

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

Cancel

Previous

Create pipeline

Build and Deploy

- ▶ This will start the Build and Deploy process

Developer Tools > CodePipeline > Pipelines > mz-codepipeline

mz-codepipeline

Notify Edit Stop execution Clone pipeline Release change

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

Source
GitHub (Version 2) [🔗](#)
Succeeded - 1 minute ago
6380364f [🔗](#)
6380364f [🔗](#) Source: Delete nodejs.zip

Disable transition

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

Create your Pipeline

- ▶ Click the link for your web application

Application 'mz-test-app' environments Create a new environment

Filter results matching the display values

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 on 64bit Amazon Linux 2

- ▶ On the Dashboard, click the URL

Mz-test-app-env Info Actions Upload and deploy

Environment overview

Health: Ok

Environment ID: e-dp4zmp7m7r

Domain: [Mz-test-app2-env.eba-pcw8agtp.us-east-1.elasticbeanstalk.com](https://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

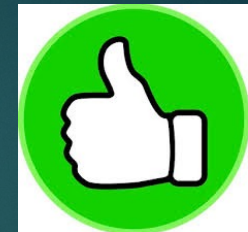
Platform state: Supported

Test your Web Application

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



The screenshot shows the AWS Elastic Beanstalk console. On the left, a green box contains the text 'Congratulations' in large font, followed by 'Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud' and 'This environment is launched with Elastic Beanstalk Node.js Platform'. On the right, a dark grey box titled 'What's Next?' lists several links: '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', and 'Working with Logs'.



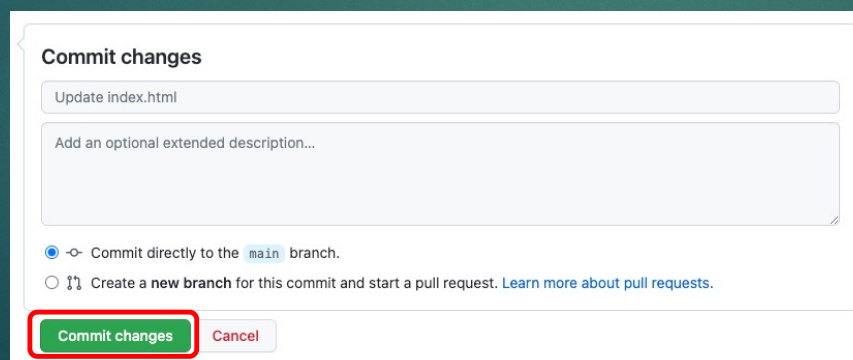
- ▶ Congratulations, but you are not done yet
- ▶ Now we must make a change to trigger the pipeline to update

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"

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

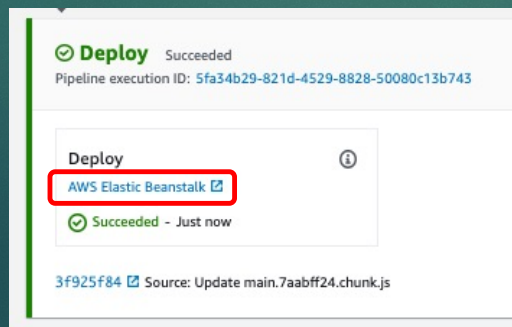
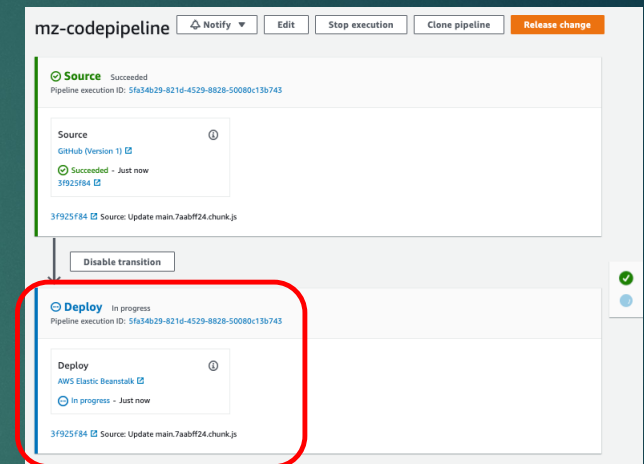
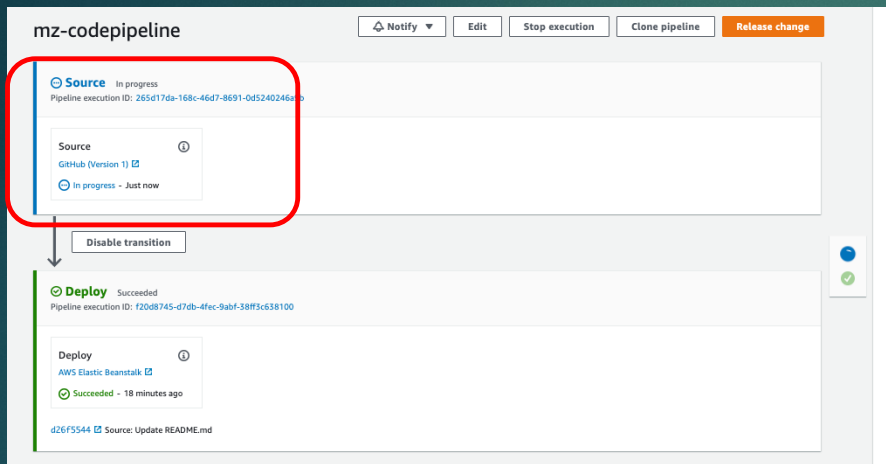
- ▶ Either push your change (from your PC) or click "Commit changes" (below)



The image shows a GitHub 'Commit changes' dialog box. At the top, it says 'Commit changes'. Below that is a text input field containing 'Update index.html'. Underneath is a larger text area for an 'optional extended description'. There are two radio button options: the first is selected and says 'Commit directly to the main branch.', and the second is unselected and says 'Create a new branch for this commit and start a pull request. Learn more about pull requests.' At the bottom, there are two buttons: 'Commit changes' (highlighted with a red box) and 'Cancel'.

Redeploy your code

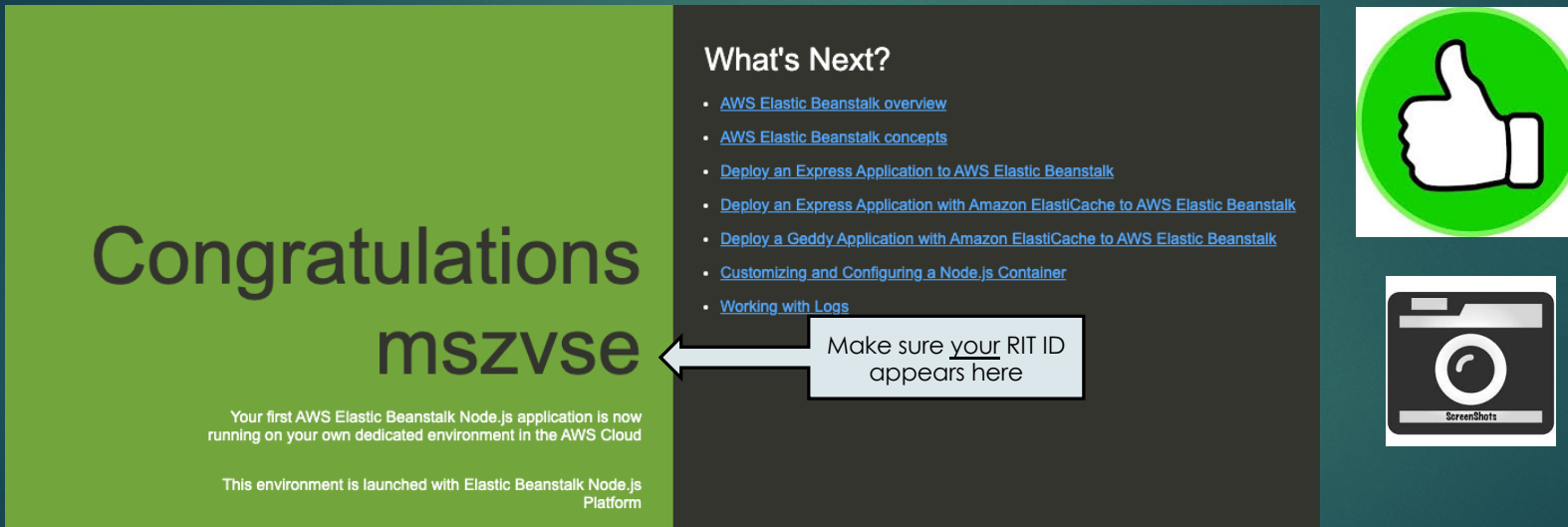
- ▶ When you push your change, the "Source" will indicate "In Progress" followed by "Deploy" stage



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!



The screenshot displays the AWS Elastic Beanstalk console. On the left, a green banner reads "Congratulations mszvsse" and "Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud". Below this, it states "This environment is launched with Elastic Beanstalk Node.js Platform". On the right, a dark grey panel titled "What's Next?" lists several links: "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", and "Working with Logs". A white arrow points from a text box "Make sure your RIT ID appears here" to the "mszvsse" application name. To the right of the console, there are two icons: a green thumbs-up icon and a camera icon labeled "ScreenShots".

- ▶ 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**

The screenshot displays a DevOps pipeline named 'mz-codepipeline'. It consists of three stages, each with a 'Disable transition' button below it. The first stage is 'Source', which is 'Succeeded' and used 'GitHub (Version 1)'. The second stage is 'Manual-Step', which is 'Succeeded' and requires 'Manual approval'. This stage is highlighted with a red rounded rectangle. The third stage is 'Deploy', which is 'Succeeded' and uses 'AWS Elastic Beanstalk'. All stages show a success status with a green checkmark and a timestamp of '1 minute ago'.

Cleanup Pipeline

- ▶ Go back to CodePipeline and click "Edit"
- ▶ Click "Delete" and confirm

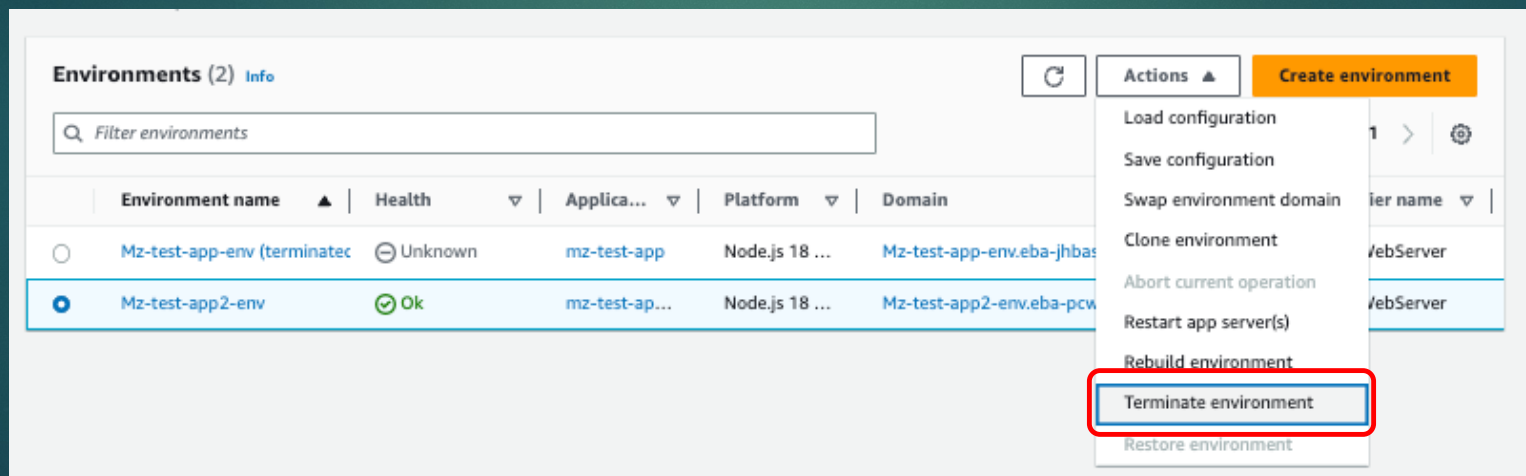
The screenshot shows the AWS CodePipeline console for a pipeline named 'mz-codepipeline'. At the top, there are buttons: 'Notify', 'Edit' (highlighted with a red box), 'Stop execution', 'Clone pipeline', and 'Release change'. Below the buttons, the pipeline execution history is displayed, showing three stages: 'Source', 'Manual', and 'Deploy'. Each stage is marked as 'Succeeded' and includes details like the provider (GitHub, Manual approval, AWS Elastic Beanstalk) and the time taken. A 'Disable transition' button is visible between stages. On the right side, there are three green checkmarks indicating successful execution.



The screenshot shows the 'Editing: mz-codepipeline' screen. At the top, there are buttons: 'Delete' (highlighted with a red box), 'Cancel', and 'Save'. Below the buttons, the pipeline is shown in edit mode with three stages: 'Source', 'Manual', and 'Deploy'. Each stage has an 'Edit stage' button. The 'Source' stage is using 'GitHub (Version 1)' as the provider. The 'Manual' stage is using 'Manual approval'. The 'Deploy' stage is using 'AWS Elastic Beanstalk'. There are '+ Add stage' buttons at the bottom of each stage section.

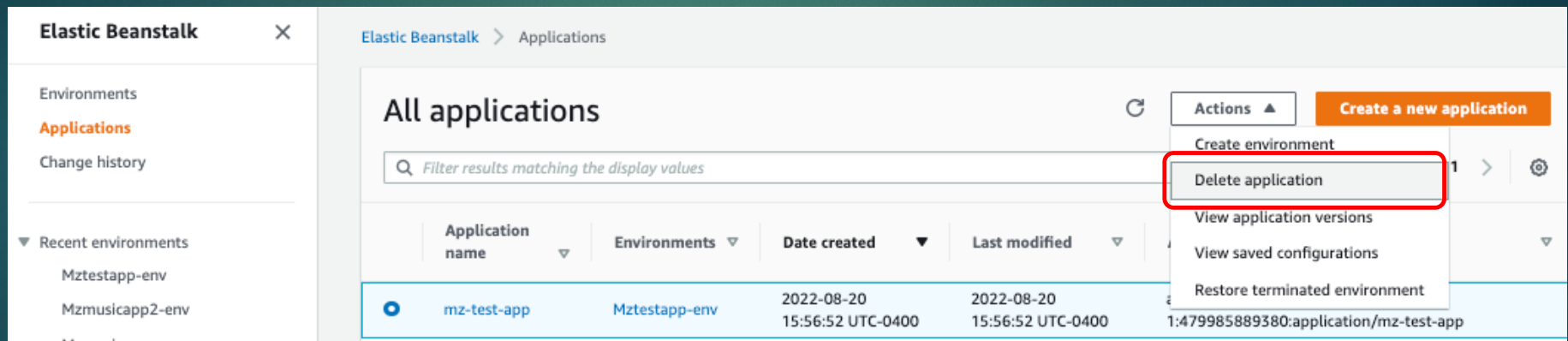
Cleanup Beanstalk

- ▶ 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”



Cleanup Beanstalk

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



The screenshot shows the AWS Elastic Beanstalk console. On the left, the 'Applications' tab is selected in the sidebar. The main area displays 'All applications' with a search bar and a table of applications. The application 'mz-test-app' is selected. The 'Actions' menu is open, and 'Delete application' is highlighted with a red box. Other actions visible include 'Create environment', 'View application versions', 'View saved configurations', and 'Restore terminated environment'.

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!