aws CERTIFIED Developer Associate

aws CERTIFIED Solutions Architect Associate

aws CERTIFIED SysOps Administrator Associate

lab

lab title

AWS Continuous Integration and Delivery (CI/CD) V1.01

Course title

BackSpace Academy
AWS Certified Associate

BackSpace

Table of Contents

Contents

About the Lab

Please note that not all AWS services are supported in all regions. Please use the US-East-1 (North Virginia) region for this lab.

These lab notes are to support the hands on instructional videos of the AWS Deployment Services section of the AWS Certified Associate Course.

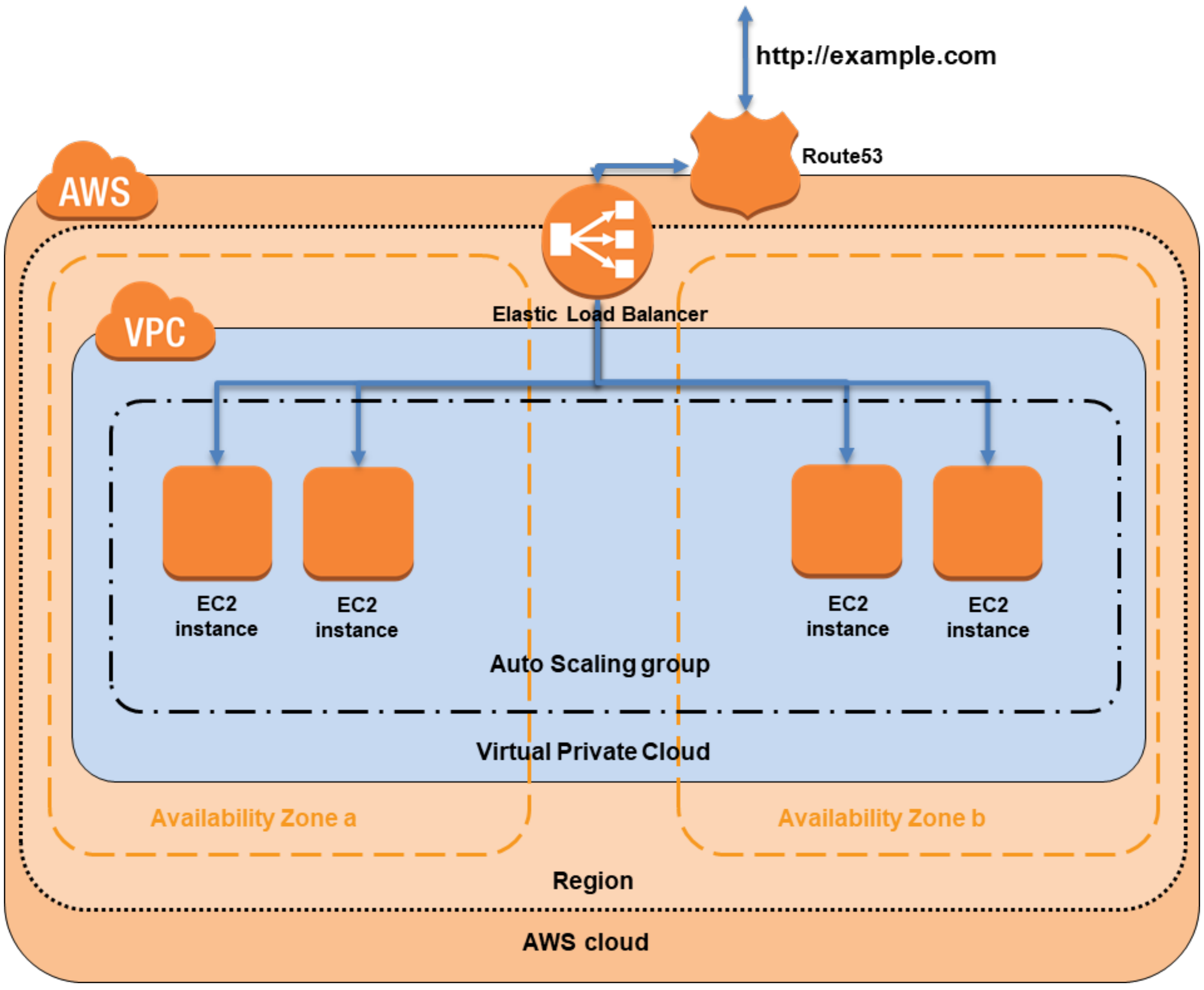
Please note that AWS services change on a weekly basis and it is extremely important you check the version number on this document to ensure you have the lastest version with any updates or corrections.

NodeJS Application Highly Available and Fault Tolerant Architecture

In this section, we will use the Elastic Beanstalk Service to create a highly available and fault tolerant architecture for deploying our application.

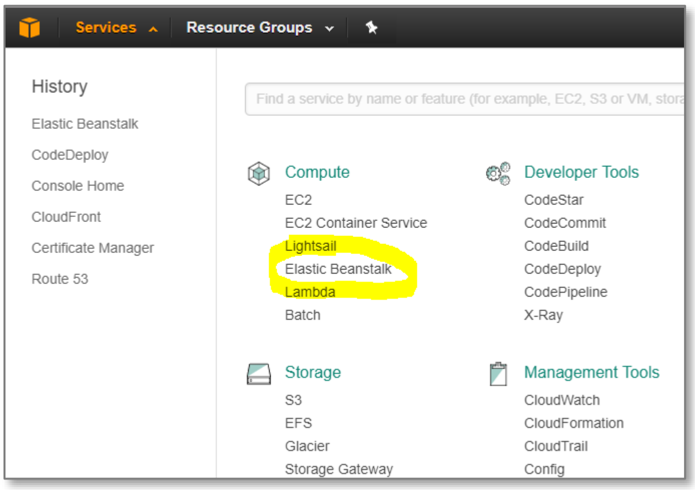
* Please note these notes have been created using the new Elastic Beanstalk console/user interface. You may need to opt-in to use the new user interface.

Create the Architecture

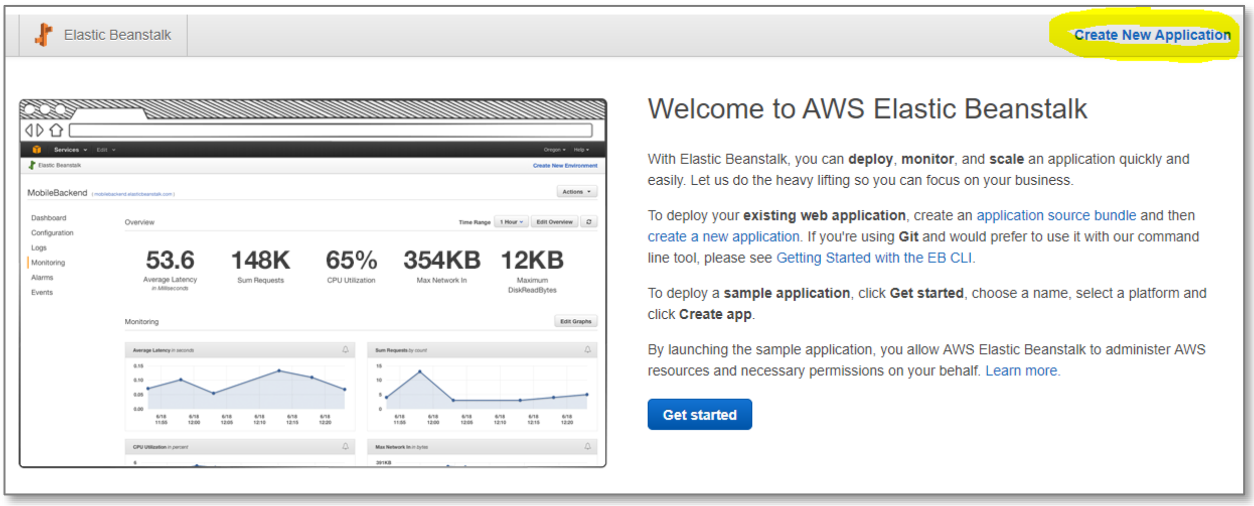


The simplest and quickest way to get our architecture deployed is using the sample NodeJS application.

Make sure you are in US-East (N. Virginia) region. From the AWS console select “Elastic Beanstalk” from the Compute services.



Click “Create New Application”



Welcome to AWS Elastic Beanstalk

With Elastic Beanstalk, you can **deploy**, **monitor**, and **scale** an application quickly and easily. Let us do the heavy lifting so you can focus on your business.

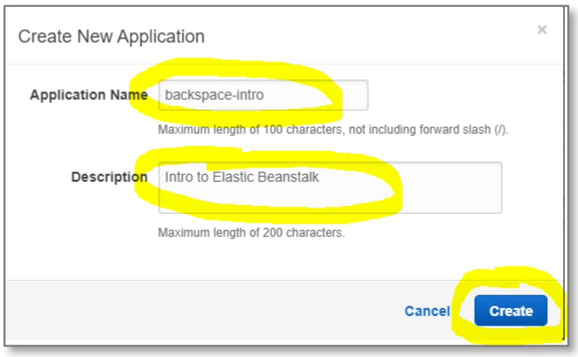
To deploy your **existing web application**, create an application source bundle and then create a new application. If you're using **Git** and would prefer to use it with our command line tool, please see Getting Started with the EB CLI.

To deploy a **sample application**, click **Get started**, choose a name, select a platform and click **Create app**.

By launching the sample application, you allow AWS Elastic Beanstalk to administer AWS resources and necessary permissions on your behalf. Learn more.

Give your application a name and description.

Click “Create”



Create New Application

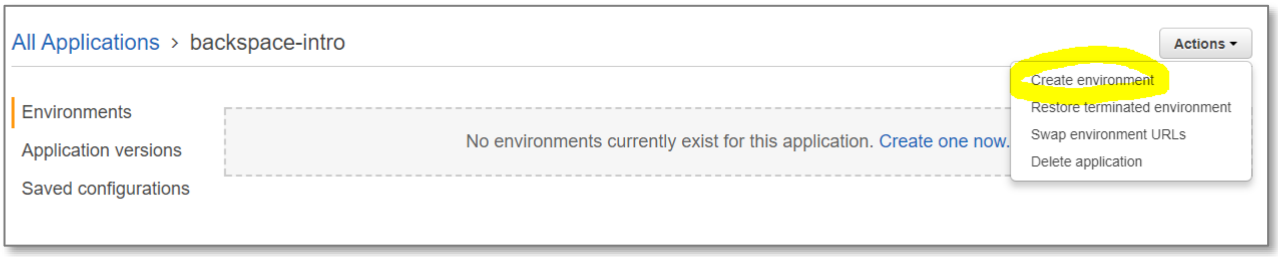
Application Name backspace-intro

Maximum length of 100 characters, not including forward slash (/).

Description Intro to Elastic Beanstalk

Maximum length of 200 characters.

Select “Actions” – “Create Environment”

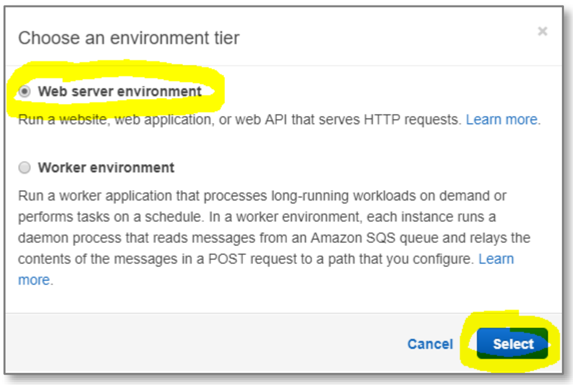

All Applications > backspace-intro

Environments
Application versions
Saved configurations

No environments currently exist for this application. Create one now.

Select “Web server environment”

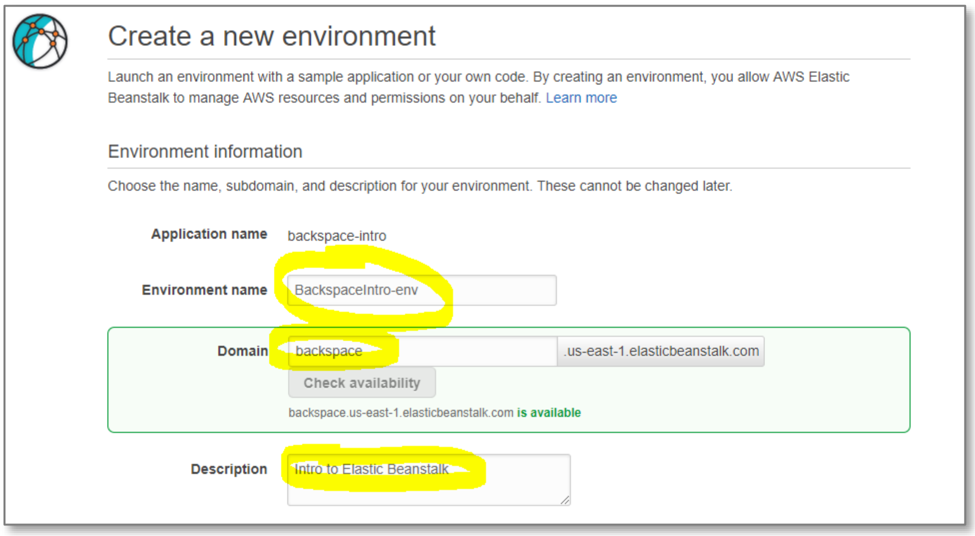
Click “Select”


Choose an environment tier

Web server environment
Run a website, web application, or web API that serves HTTP requests. Learn more.

Worker environment
Run a worker application that processes long-running workloads on demand or performs tasks on a schedule. In a worker environment, each instance runs a daemon process that reads messages from an Amazon SQS queue and relays the contents of the messages in a POST request to a path that you configure. Learn more.

Give your environment a name, domain name and description.


Create a new environment

Launch an environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. Learn more

Environment information

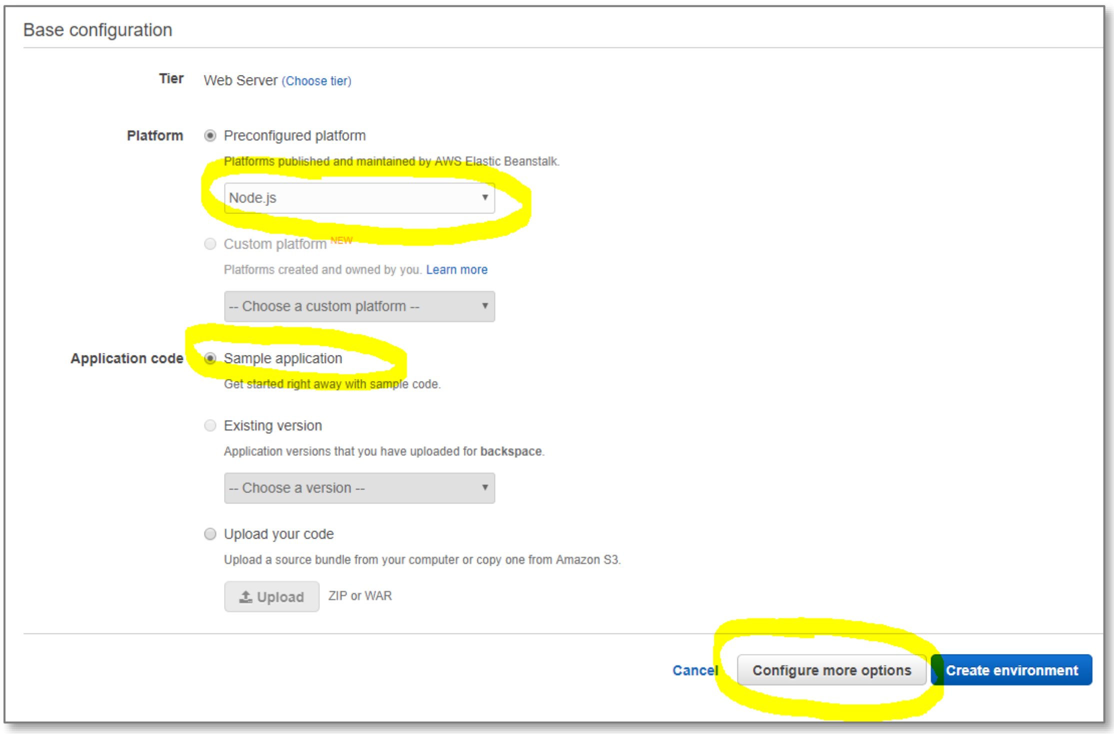
Choose the name, subdomain, and description for your environment. These cannot be changed later.

Application name	backspace-intro
Environment name	BackspaceIntro-env
Domain	backspace .us-east-1.elasticbeanstalk.com
	backspace.us-east-1.elasticbeanstalk.com is available
Description	Intro to Elastic Beanstalk

Select “Platform” - “Preconfigured platform” – “Node.js”

Select “Application code” -Sample application”

Click “Configure more options”



Base configuration

Tier: Web Server (Choose tier)

Platform: [✓ Preconfigured platform] Platforms published and maintained by AWS Elastic Beanstalk. Node.js

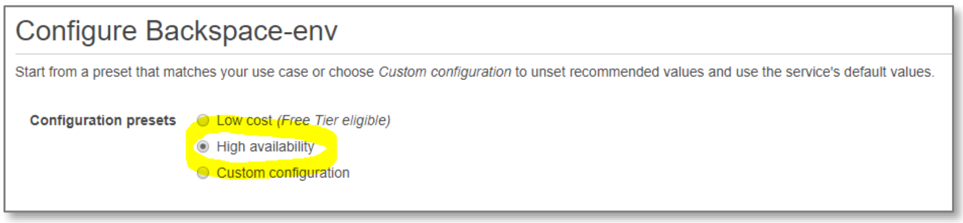
Custom platform NEW — Platforms created and owned by you. Learn more — -- Choose a custom platform --

Application code: [✓ Sample application] Get started right away with sample code.

Existing version — Application versions that you have uploaded for **backspace**. -- Choose a version --

Upload your code — Upload a source bundle from your computer or copy one from Amazon S3. Upload ZIP or WAR

Select “Configuration presets” – “High availability”

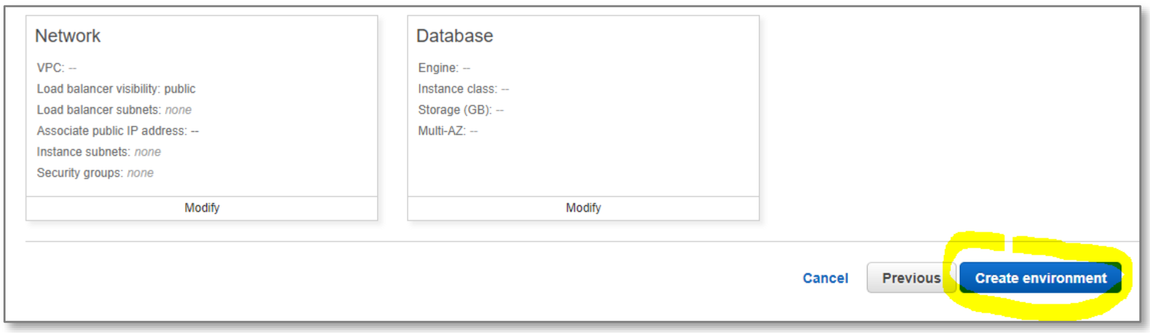


Configure Backspace-env

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: Low cost *(Free Tier eligible)* [✓ High availability] Custom configuration

Scroll down and Click “Create Environment”



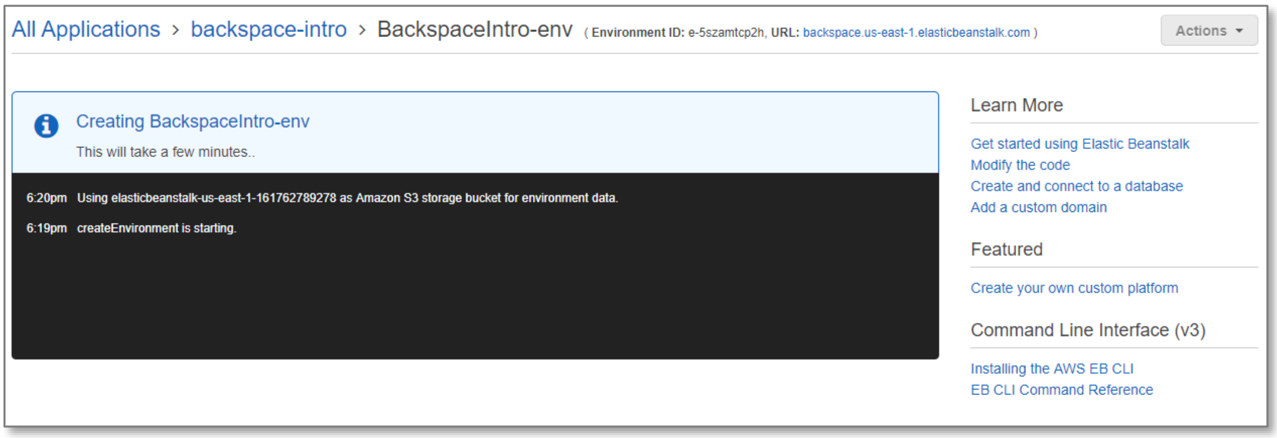
Network

VPC: --
Load balancer visibility: public
Load balancer subnets: *none*
Associate public IP address: --
Instance subnets: *none*
Security groups: *none*

Database

Engine: --
Instance class: --
Storage (GB): --
Multi-AZ: --

Elastic Beanstalk will start creating your environment and displaying log messages

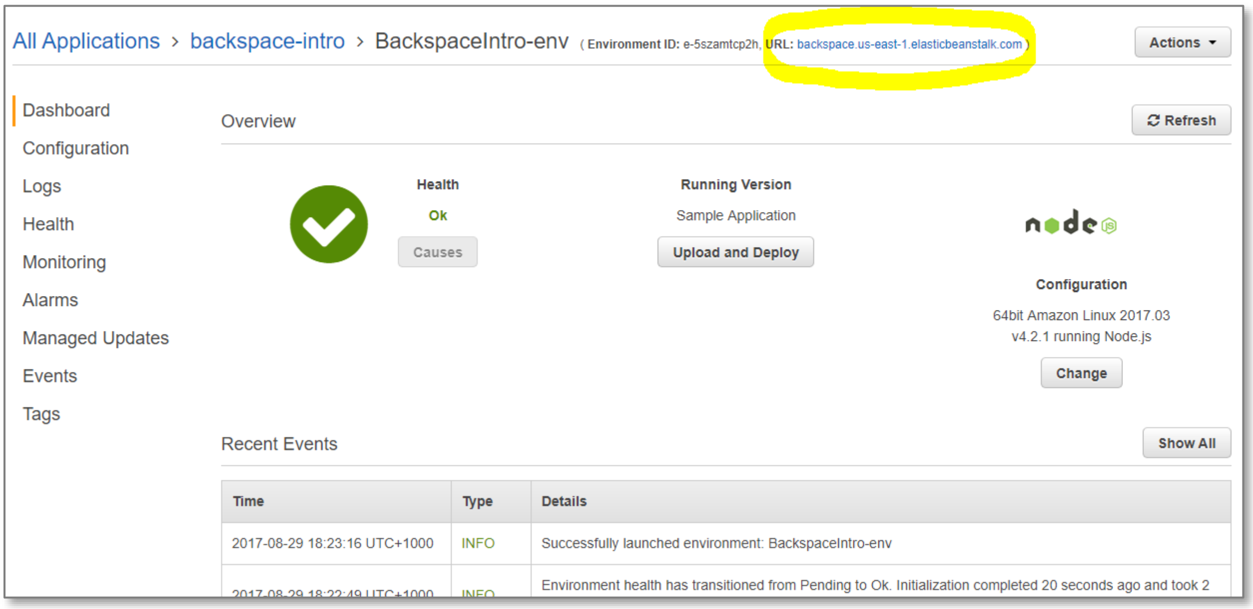

All Applications > backspace-intro > BackspaceIntro-env (Environment ID: e-5szamtcp2h, URL: backspace.us-east-1.elasticbeanstalk.com)

Creating BackspaceIntro-env
This will take a few minutes..

```
6:20pm  Using elasticbeanstalk-us-east-1-161762789278 as Amazon S3 storage bucket for environment data.
6:19pm  createEnvironment is starting.
```

Actions
Learn More
Get started using Elastic Beanstalk
Modify the code
Create and connect to a database
Add a custom domain
Featured
Create your own custom platform
Command Line Interface (v3)
Installing the AWS EB CLI
EB CLI Command Reference

When the environment has launched you can view the sample application by clicking on the URL

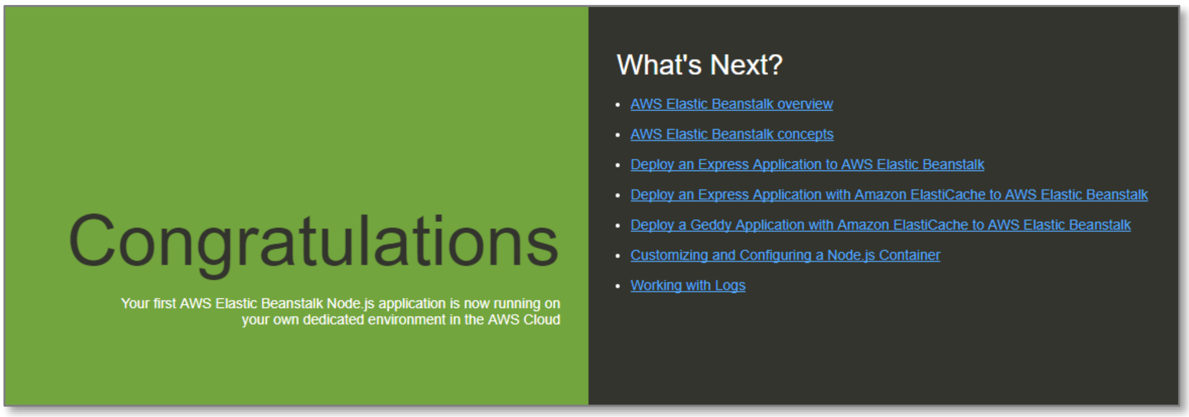

All Applications > backspace-intro > BackspaceIntro-env (Environment ID: e-5szamtcp2h, URL: backspace.us-east-1.elasticbeanstalk.com)

Overview

Health: Ok
Running Version: Sample Application
Configuration: 64bit Amazon Linux 2017.03 v4.2.1 running Node.js

Recent Events

Time	Type	Details
2017-08-29 18:23:16 UTC+1000	INFO	Successfully launched environment: BackspaceIntro-env
2017-08-29 18:22:49 UTC+1000	INFO	Environment health has transitioned from Pending to Ok. Initialization completed 20 seconds ago and took 2


Congratulations

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

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

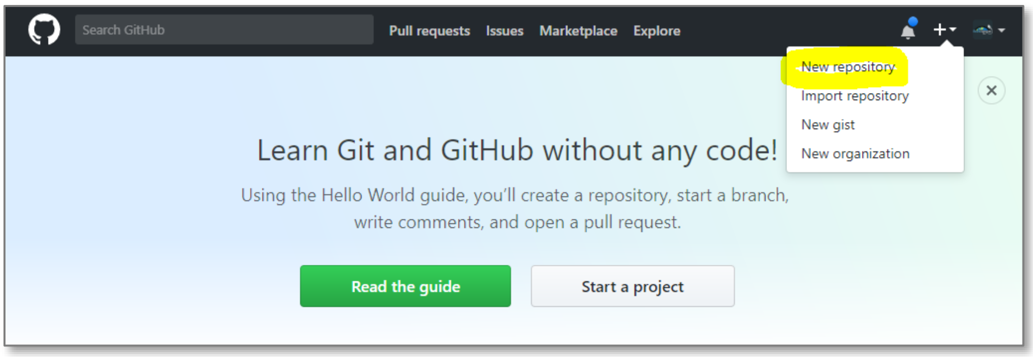
Creating a Continuous Integration and Delivery (CI/CD) Pipeline

In this section, we will use the CodePipeline Service to create a Continuous Integration and Continuous Delivery (CI/CD) pipeline our application. We will also create a GitHub repository to store our code and provide updates to be pushed our AWS environment.

Create a GitHub Repository

If you don't already have a GitHub account go to github.com and sign up (its free).

Go to github.com and select "New repository" from the "Create" menu on the top right.

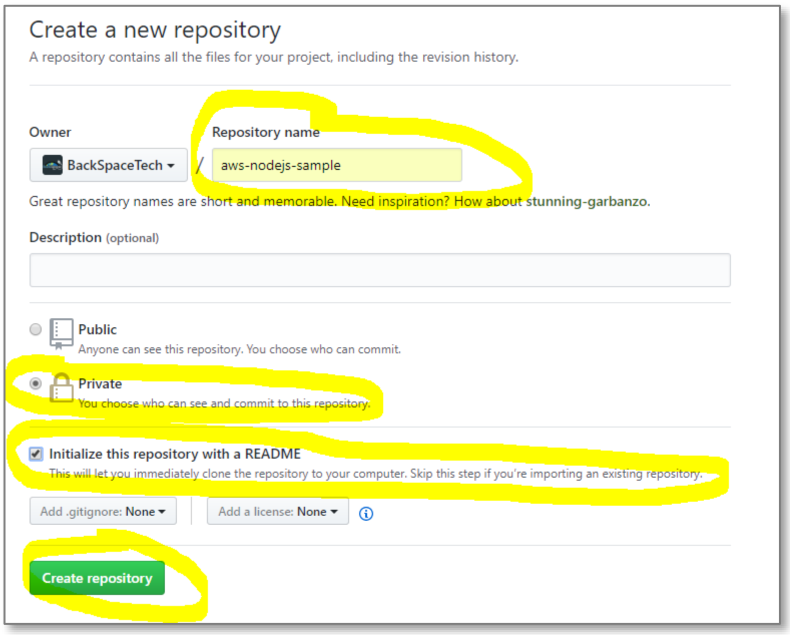

Search GitHub
Pull requests
Issues
Marketplace
Explore
New repository
Import repository
New gist
New organization
Learn Git and GitHub without any code!
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.
Read the guide
Start a project

Give your repository a name.

Set permission as "private"

Select "Initialize this repository with a README"

Click "Create repository"


Create a new repository

A repository contains all the files for your project, including the revision history.

Owner: BackSpaceTech / Repository name: aws-nodejs-sample

Great repository names are short and memorable. Need inspiration? How about stunning-garbanzo.

Description (optional)

Public
Anyone can see this repository. You choose who can commit.

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

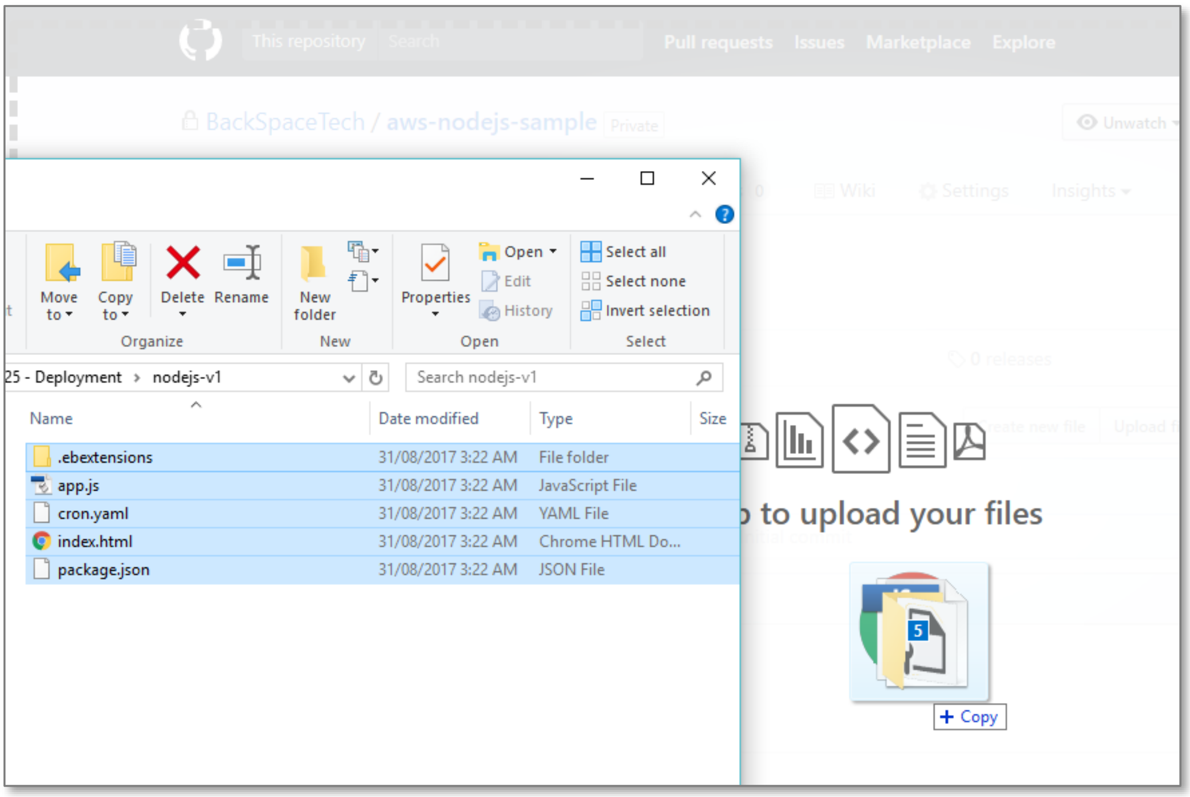
[✓] Initialize this repository with a README
This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None | Add a license: None

Download the sample NodeJS application from the Elastic Beanstalk documentation page:

http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/samples/nodejs-v1.zip

Unzip the archive and drag and drop the files onto the repository code page.

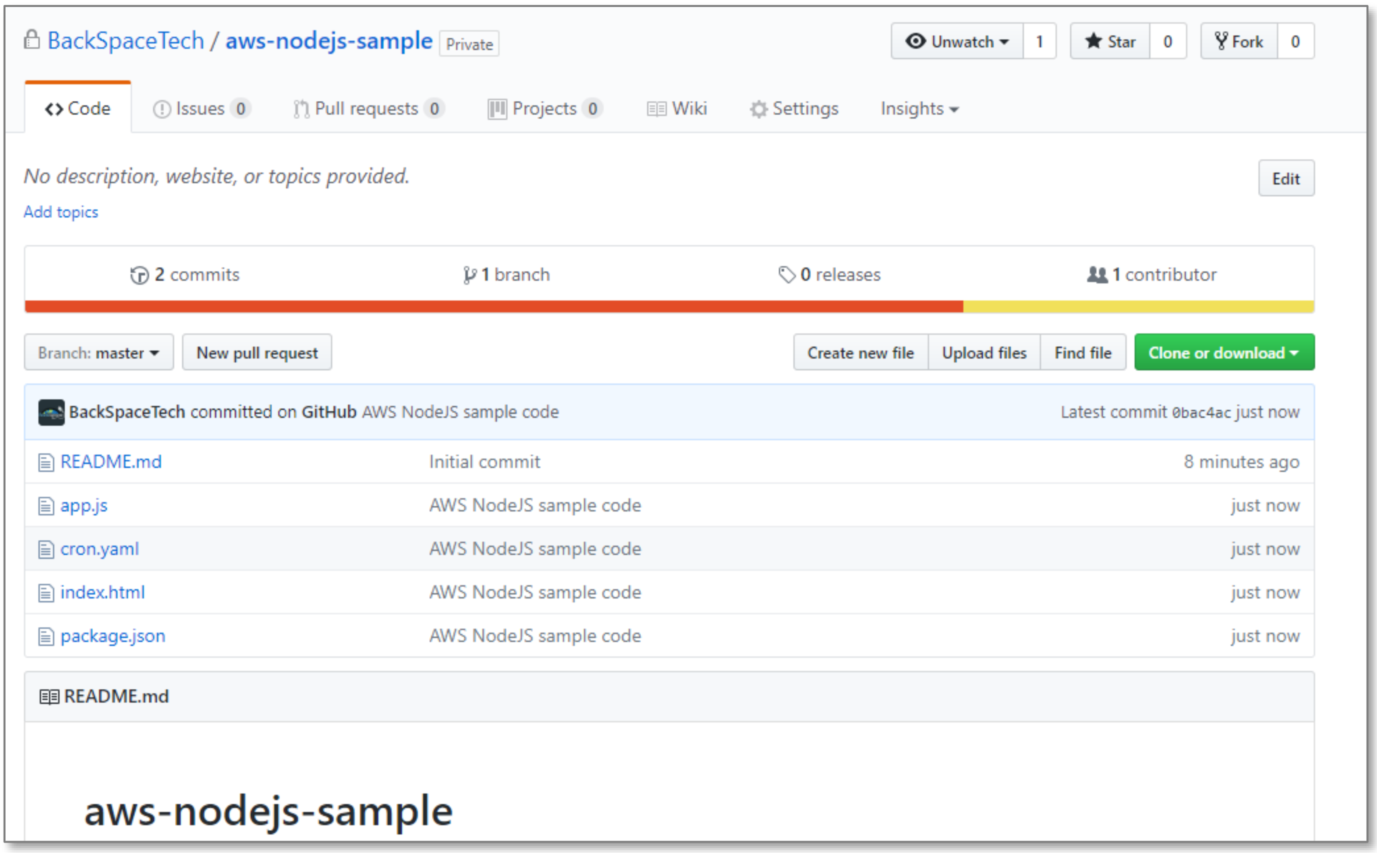


Name	Date modified	Type	Size
.ebextensions	31/08/2017 3:22 AM	File folder	
app.js	31/08/2017 3:22 AM	JavaScript File	
cron.yaml	31/08/2017 3:22 AM	YAML File	
index.html	31/08/2017 3:22 AM	Chrome HTML Do...	
package.json	31/08/2017 3:22 AM	JSON File	

Scroll down and enter a commit description.

Click “Commit changes”

Your files will now be uploaded.



BackSpaceTech / aws-nodejs-sample Private

Unwatch 1 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Settings | Insights

No description, website, or topics provided. Edit

Add topics

2 commits | 1 branch | 0 releases | 1 contributor

Branch: master | New pull request | Create new file | Upload files | Find file | Clone or download

BackSpaceTech committed on GitHub AWS NodeJS sample code — Latest commit `0bac4ac` just now

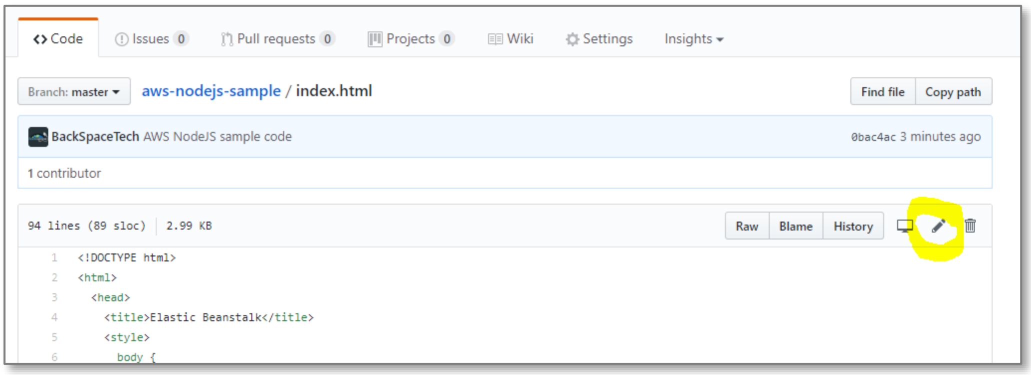
README.md	Initial commit	8 minutes ago
app.js	AWS NodeJS sample code	just now
cron.yaml	AWS NodeJS sample code	just now
index.html	AWS NodeJS sample code	just now
package.json	AWS NodeJS sample code	just now

README.md

aws-nodejs-sample

Click on index.html

Click on the edit icon



Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Settings | Insights

Branch: master | aws-nodejs-sample / index.html | Find file | Copy path

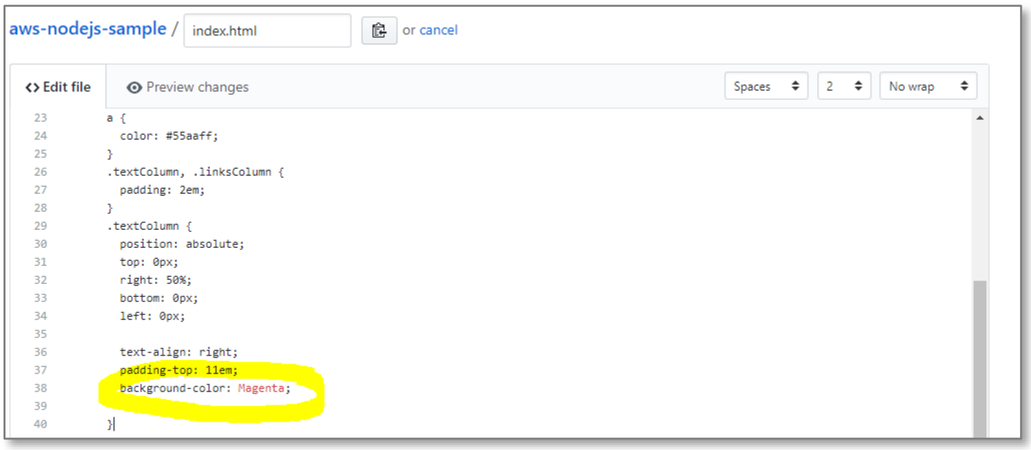
BackSpaceTech AWS NodeJS sample code — `0bac4ac` 3 minutes ago

1 contributor

94 lines (89 sloc) | 2.99 KB — Raw | Blame | History

```
<!DOCTYPE html>
<html>
  <head>
    <title>Elastic Beanstalk</title>
    <style>
      body {
```

Scroll down to line 38 and change the colour from #73A53E to Magenta


aws-nodejs-sample / index.html or cancel

Edit file | Preview changes | Spaces | 2 | No wrap

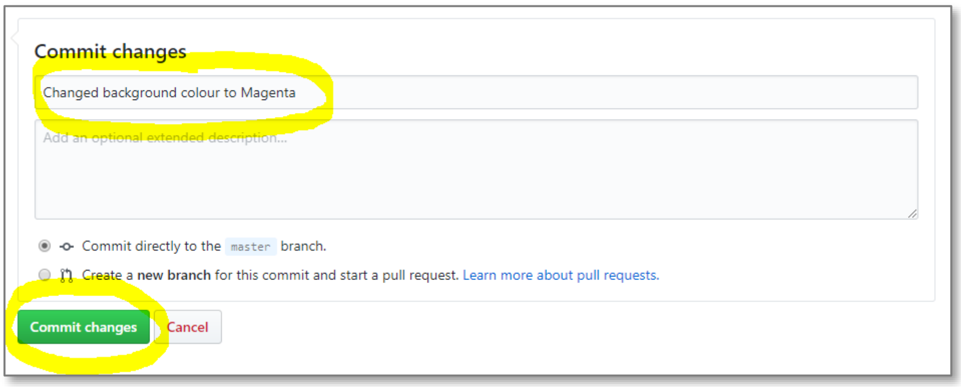
```
  a {
    color: #55aaff;
  }
  .textColumn, .linksColumn {
    padding: 2em;
  }
  .textColumn {
    position: absolute;
    top: 0px;
    right: 50%;
    bottom: 0px;
    left: 0px;

    text-align: right;
    padding-top: 11em;
    background-color: Magenta;

  }
```

Scroll down and enter a commit description.

Click “Commit changes”


Commit changes

Changed background colour to Magenta

Add an optional extended description...

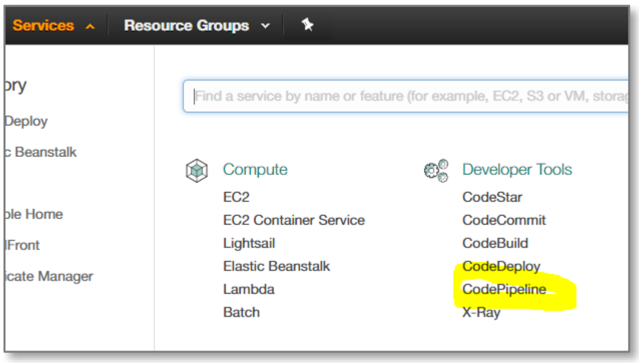
(●) Commit directly to the `master` branch.

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

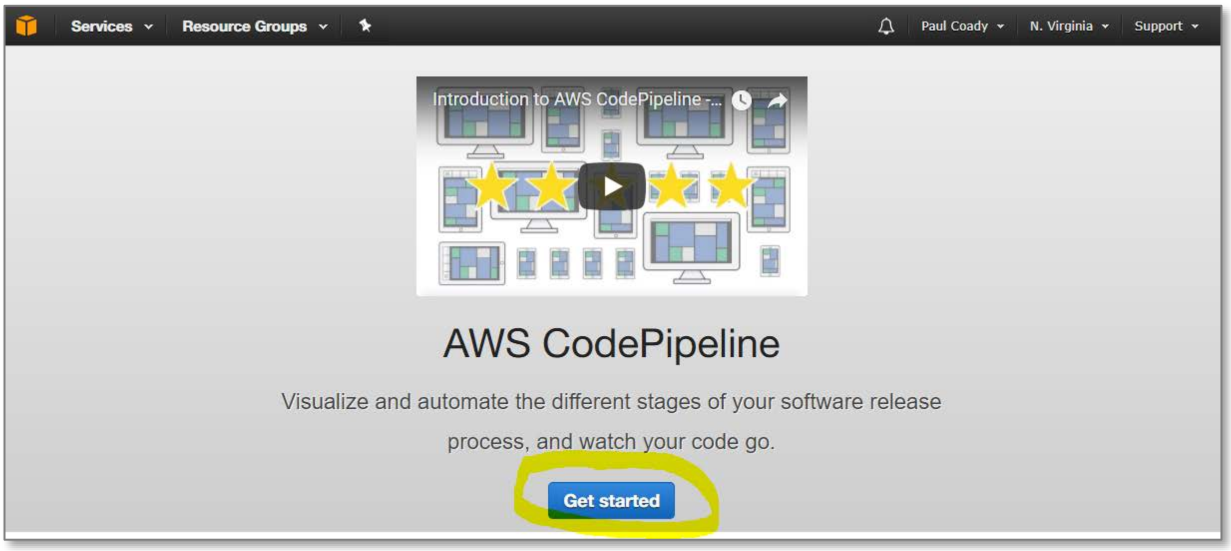
Commit changes | Cancel

Creating the Pipeline

Make sure you are in US-East (N. Virginia) region. From the AWS console select "CodePipeline" from the Developer Tools services.

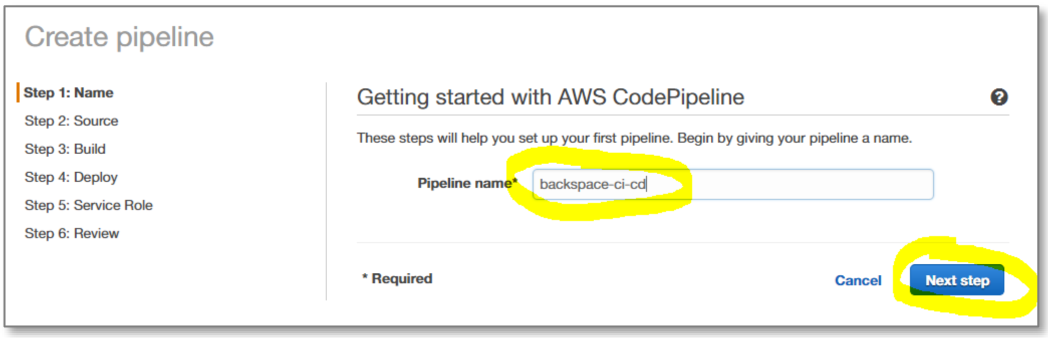


Click "Get Started"



Visualize and automate the different stages of your software release process, and watch your code go.

Give your pipeline a name and click "Next Step"



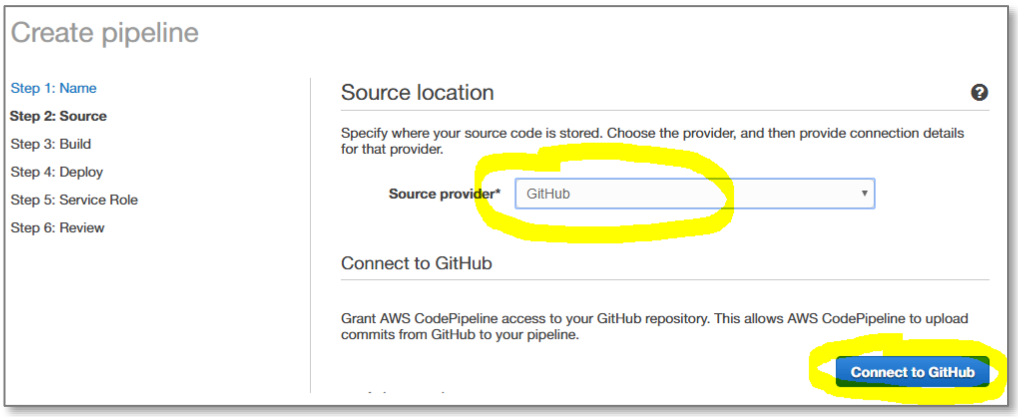
These steps will help you set up your first pipeline. Begin by giving your pipeline a name.

Pipeline name*: backspace-ci-cd

* Required

Select "GitHub" as Source Location

Click “Connect to GitHub”



Create pipeline

Step 1: Name
Step 2: Source
Step 3: Build
Step 4: Deploy
Step 5: Service Role
Step 6: Review

Source location

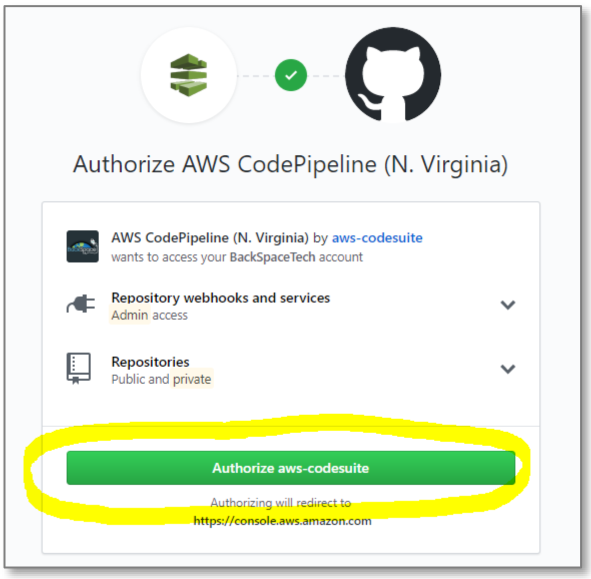
Specify where your source code is stored. Choose the provider, and then provide connection details for that provider.

Source provider* GitHub

Connect to GitHub

Grant AWS CodePipeline access to your GitHub repository. This allows AWS CodePipeline to upload commits from GitHub to your pipeline.

Click “Authorize aws-codesuite”



Authorize AWS CodePipeline (N. Virginia)

AWS CodePipeline (N. Virginia) by aws-codesuite
wants to access your BackSpaceTech account

Repository webhooks and services
Admin access

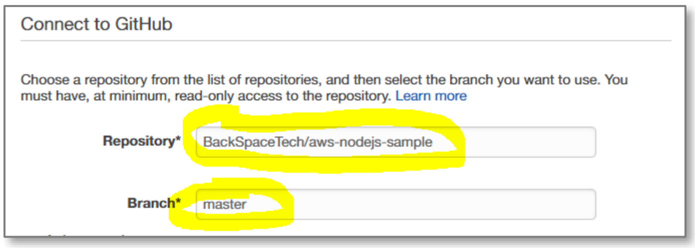
Repositories
Public and private

Authorizing will redirect to
https://console.aws.amazon.com

*Note this step may force a re-login to the console which may reset your screen. If this occurs go back and start the pipeline creation process again including “Connect to GitHub”.

Select the GitHub repository you created.

Select master branch



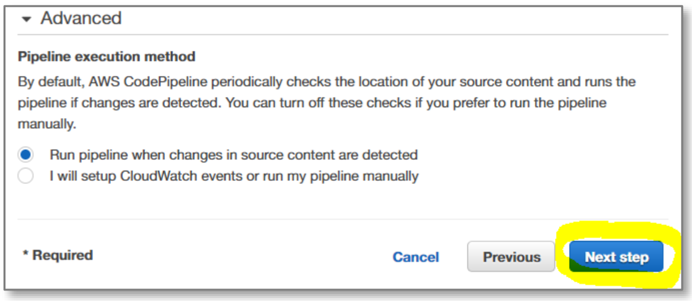
Connect to GitHub

Choose a repository from the list of repositories, and then select the branch you want to use. You must have, at minimum, read-only access to the repository. Learn more

Repository* BackSpaceTech/aws-nodejs-sample

Branch* master

Leave advanced settings as they are and click “Next Step”



Advanced

Pipeline execution method

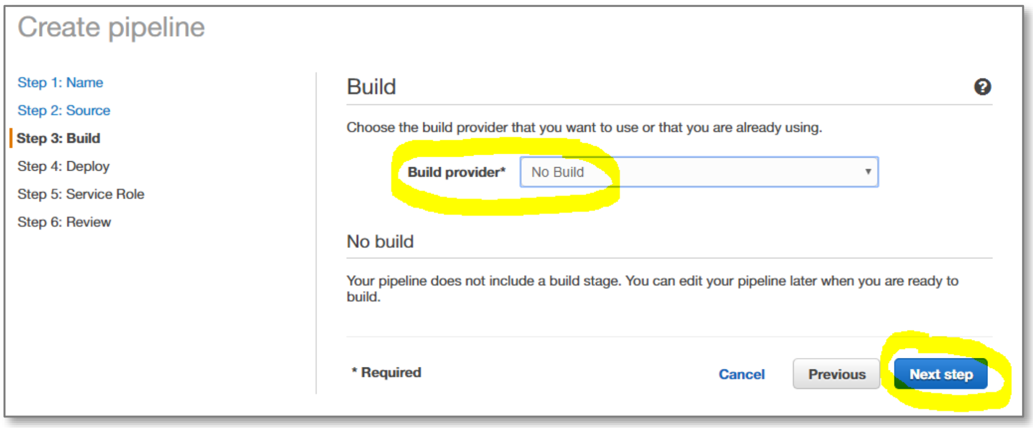
By default, AWS CodePipeline periodically checks the location of your source content and runs the pipeline if changes are detected. You can turn off these checks if you prefer to run the pipeline manually.

- [✓ Run pipeline when changes in source content are detected]
- I will setup CloudWatch events or run my pipeline manually

* Required

Select “No Build” for Build provider.

Click “Next step”



Create pipeline

Step 1: Name
Step 2: Source
Step 3: Build
Step 4: Deploy
Step 5: Service Role
Step 6: Review

Build

Choose the build provider that you want to use or that you are already using.

Build provider* No Build

No build

Your pipeline does not include a build stage. You can edit your pipeline later when you are ready to build.

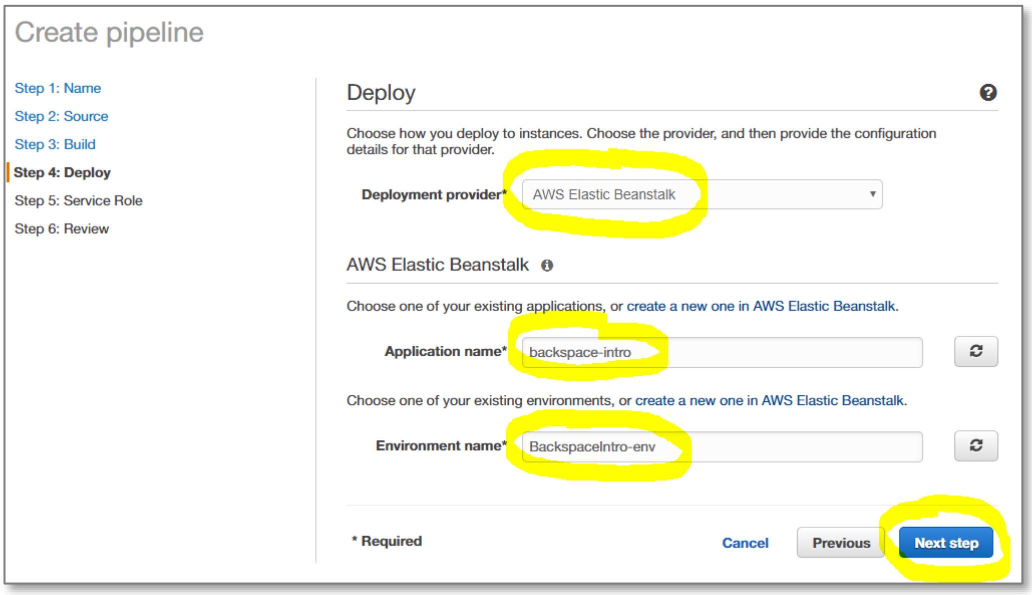
* Required

Select “Elastic Beanstalk” for Deployment provider

Select the Elastic Beanstalk application

Select the Elastic Beanstalk environment

Click “Next step”


Create pipeline

Step 1: Name
Step 2: Source
Step 3: Build
Step 4: Deploy
Step 5: Service Role
Step 6: Review

Deploy

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

Deployment provider* AWS Elastic Beanstalk

AWS Elastic Beanstalk

Choose one of your existing applications, or create a new one in AWS Elastic Beanstalk.

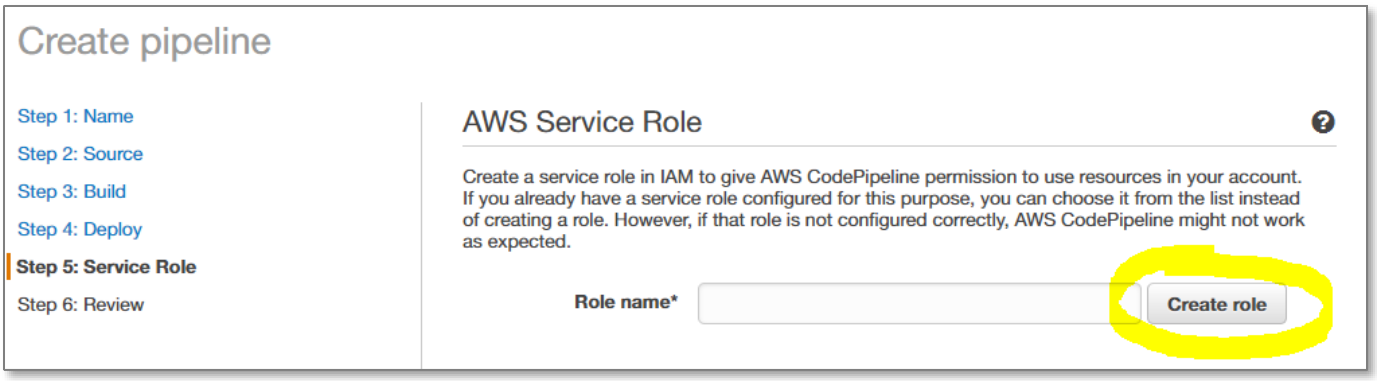
Application name* backspace-intro

Choose one of your existing environments, or create a new one in AWS Elastic Beanstalk.

Environment name* BackspaceIntro-env

*** Required**

Click "Create role"


Create pipeline

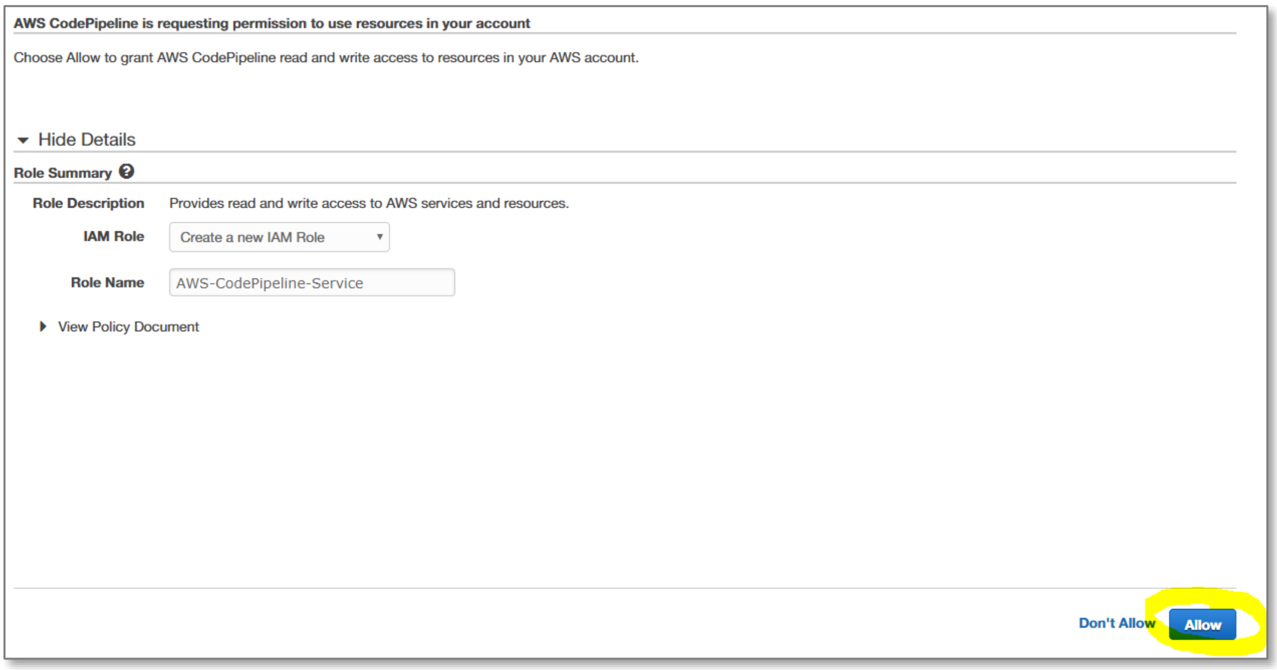
Step 1: Name
Step 2: Source
Step 3: Build
Step 4: Deploy
Step 5: Service Role
Step 6: Review

AWS Service Role

Create a service role in IAM to give AWS CodePipeline permission to use resources in your account. If you already have a service role configured for this purpose, you can choose it from the list instead of creating a role. However, if that role is not configured correctly, AWS CodePipeline might not work as expected.

Role name*

Click "Allow" to create the role



AWS CodePipeline is requesting permission to use resources in your account

Choose Allow to grant AWS CodePipeline read and write access to resources in your AWS account.

Hide Details

Role Summary

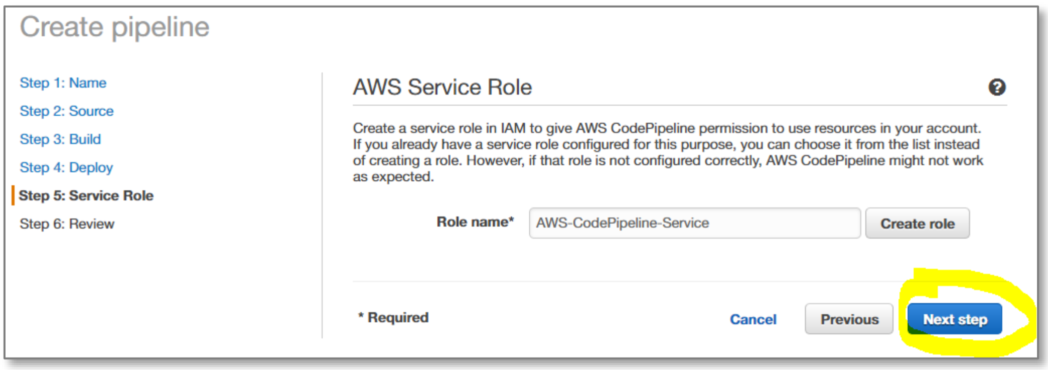
Role Description: Provides read and write access to AWS services and resources.

IAM Role: Create a new IAM Role

Role Name: AWS-CodePipeline-Service

You will now be returned to the Code Pipeline console with the new role entered.

Click "Next step"



Create pipeline

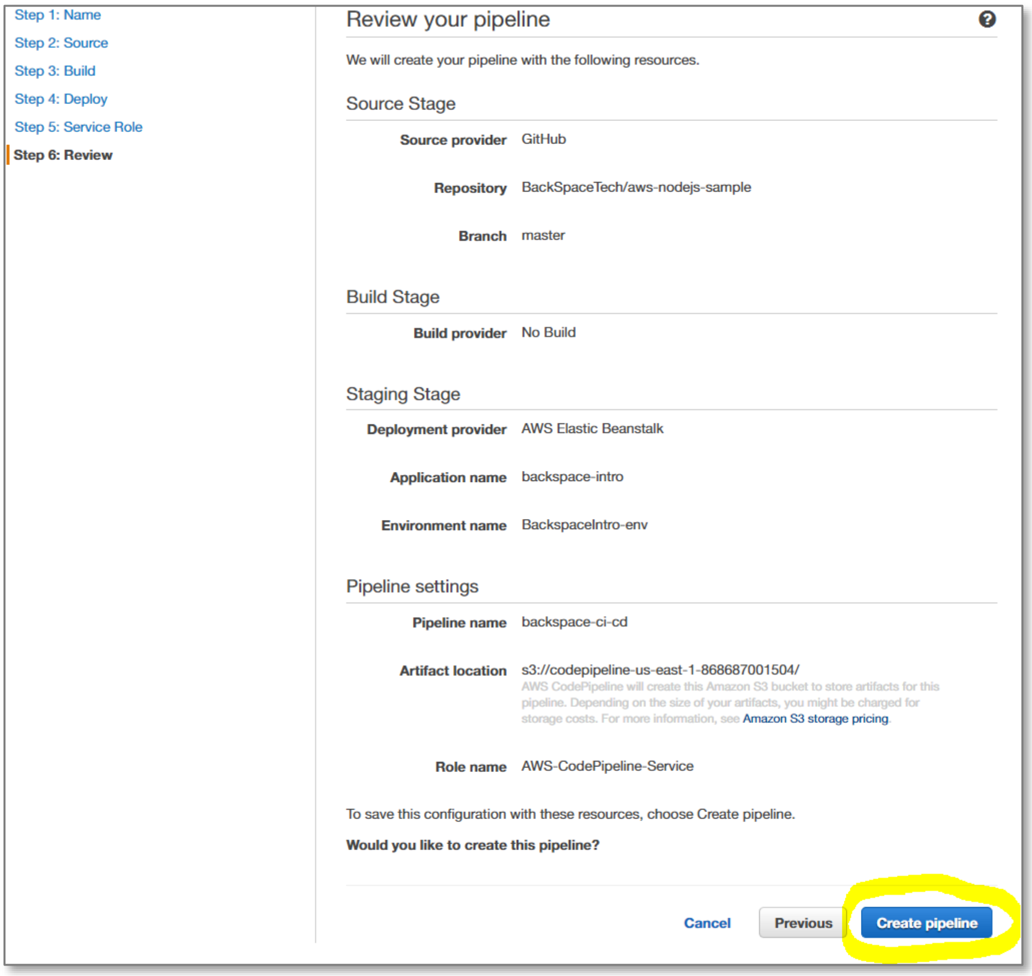
AWS Service Role

Create a service role in IAM to give AWS CodePipeline permission to use resources in your account. If you already have a service role configured for this purpose, you can choose it from the list instead of creating a role. However, if that role is not configured correctly, AWS CodePipeline might not work as expected.

Role name*: AWS-CodePipeline-Service

* Required

Review the pipeline and click "Create pipeline"


Step 1: Name
Step 2: Source
Step 3: Build
Step 4: Deploy
Step 5: Service Role
Step 6: Review

Review your pipeline

We will create your pipeline with the following resources.

Source Stage

Source provider	GitHub
Repository	BackSpaceTech/aws-nodejs-sample
Branch	master

Build Stage

Build provider	No Build

Staging Stage

Deployment provider	AWS Elastic Beanstalk
Application name	backspace-intro
Environment name	BackspaceIntro-env

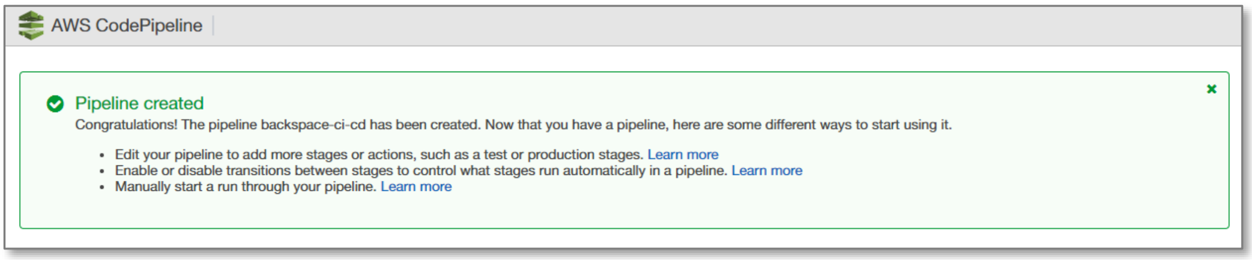
Pipeline settings

Pipeline name	backspace-ci-cd
Artifact location	s3://codepipeline-us-east-1-868687001504/ AWS CodePipeline will create this Amazon S3 bucket to store artifacts for this pipeline. Depending on the size of your artifacts, you might be charged for storage costs. For more information, see Amazon S3 storage pricing.
Role name	AWS-CodePipeline-Service

To save this configuration with these resources, choose Create pipeline.

Would you like to create this pipeline?

Your pipeline has been created.



Pipeline created

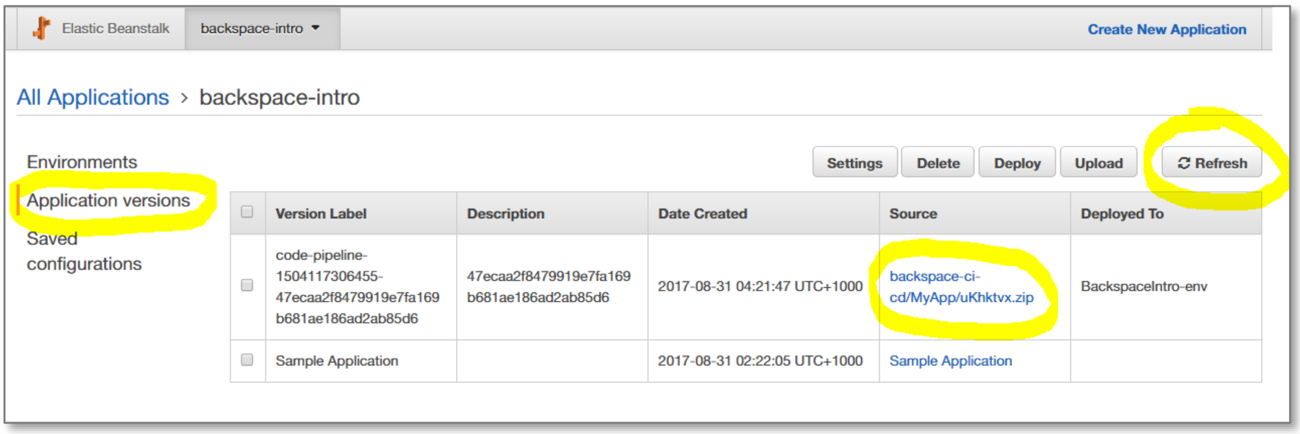
Congratulations! The pipeline backspace-ci-cd has been created. Now that you have a pipeline, here are some different ways to start using it.

- Edit your pipeline to add more stages or actions, such as a test or production stages. Learn more
- Enable or disable transitions between stages to control what stages run automatically in a pipeline. Learn more
- Manually start a run through your pipeline. Learn more

Pushing Updates to our Elastic Beanstalk Environment

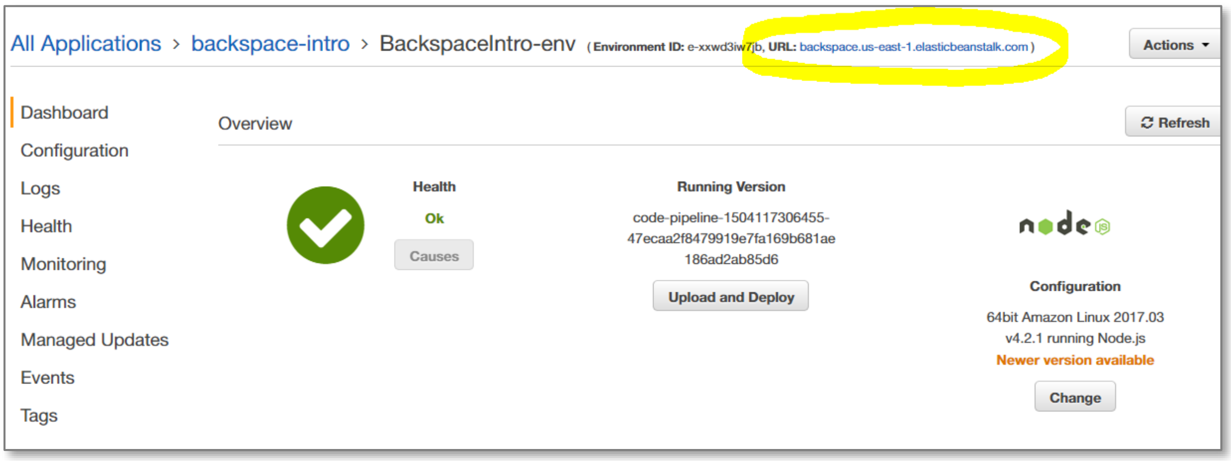
Go back to the Elastic Beanstalk console.

Go to “Application versions” and click refresh to see the new version from your GitHub repository.



	Version Label	Description	Date Created	Source	Deployed To
☐	code-pipeline-1504117306455-47ecaa2f8479919e7fa169b681ae186ad2ab85d6	47ecaa2f8479919e7fa169b681ae186ad2ab85d6	2017-08-31 04:21:47 UTC+1000	backspace-ci-cd/MyApp/uKhktvx.zip	BackspaceIntro-env
☐	Sample Application		2017-08-31 02:22:05 UTC+1000	Sample Application	

Go back to your Elastic Beanstalk environment and click the URL.



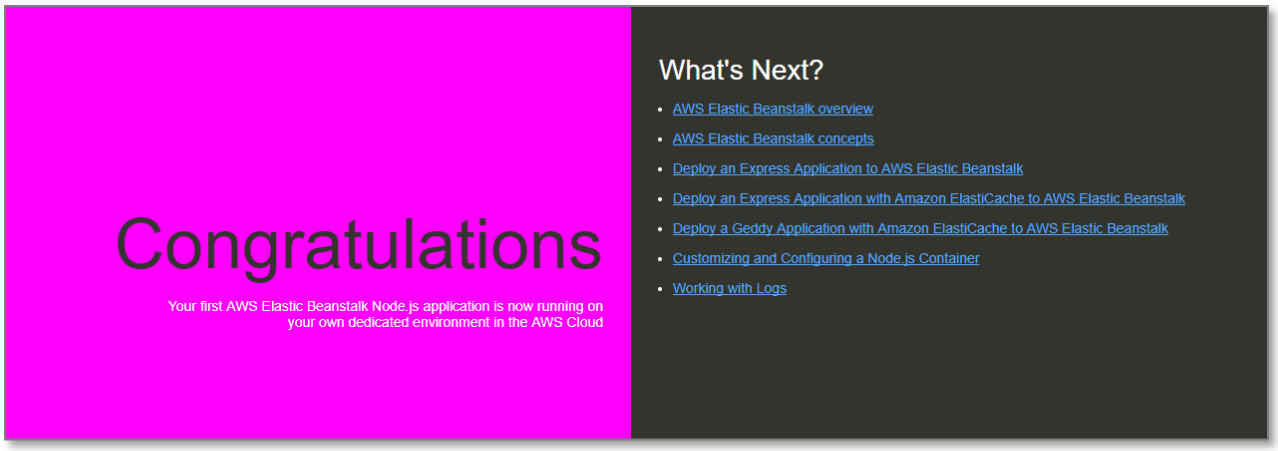
(Environment ID: e-xxwd3iw7jb, URL: backspace.us-east-1.elasticbeanstalk.com)

Health
Ok

Running Version
code-pipeline-1504117306455-47ecaa2f8479919e7fa169b681ae186ad2ab85d6

Configuration
64bit Amazon Linux 2017.03 v4.2.1 running Node.js
Newer version available

You will see the updated version with the Magenta background deployed.



Congratulations

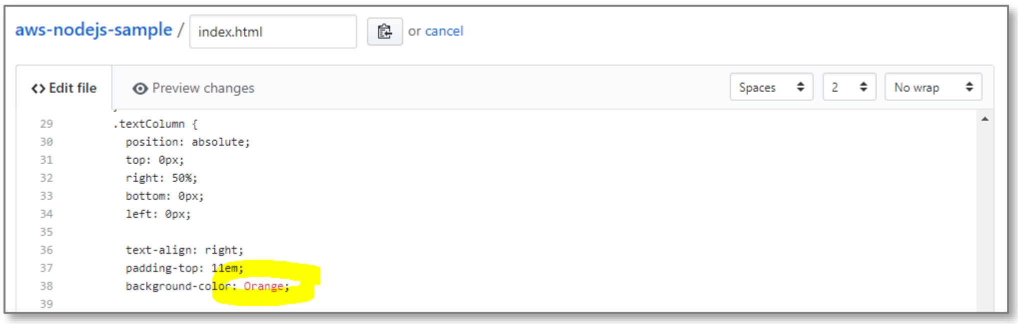
Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud

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

Now go back to GitHub to edit index.html again.

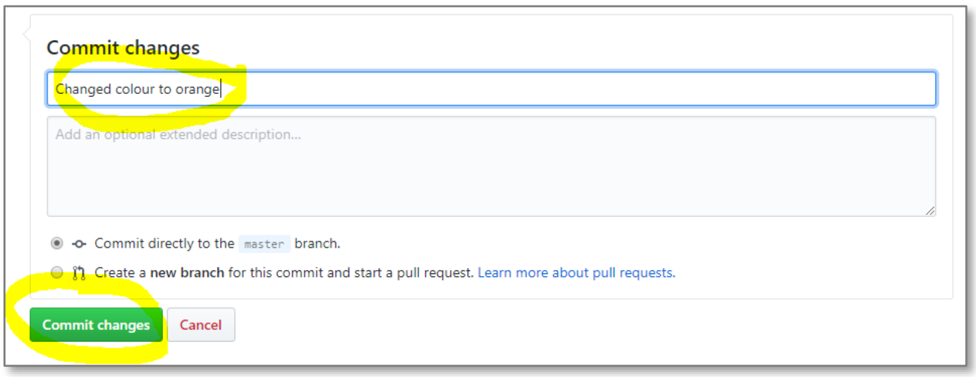
Change the colour at line 38 to Orange.



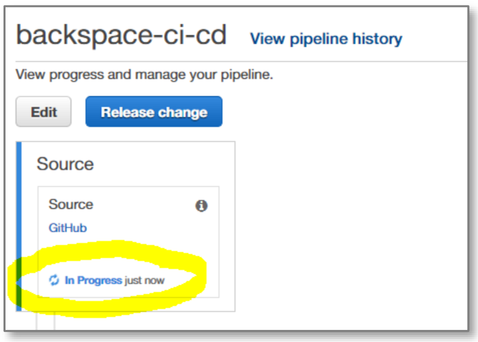
```
.textColumn {
  position: absolute;
  top: 0px;
  right: 50%;
  bottom: 0px;
  left: 0px;

  text-align: right;
  padding-top: 11em;
  background-color: Orange;
```

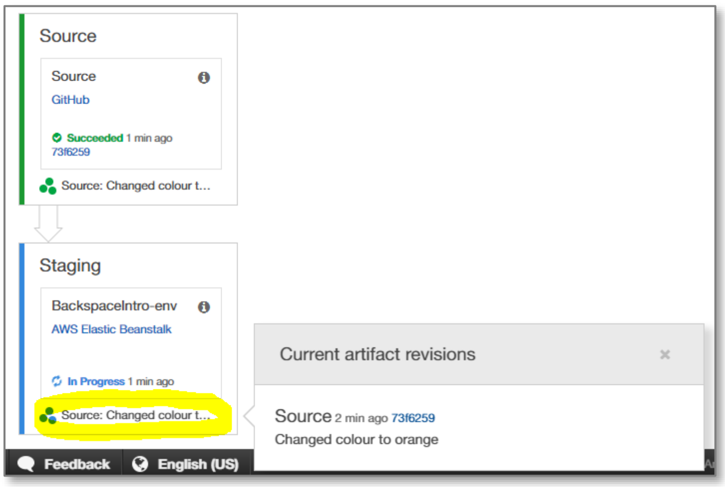
Add commit description and click “Commit changes”


Commit changes
Changed colour to orange
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.
Commit changes
Cancel

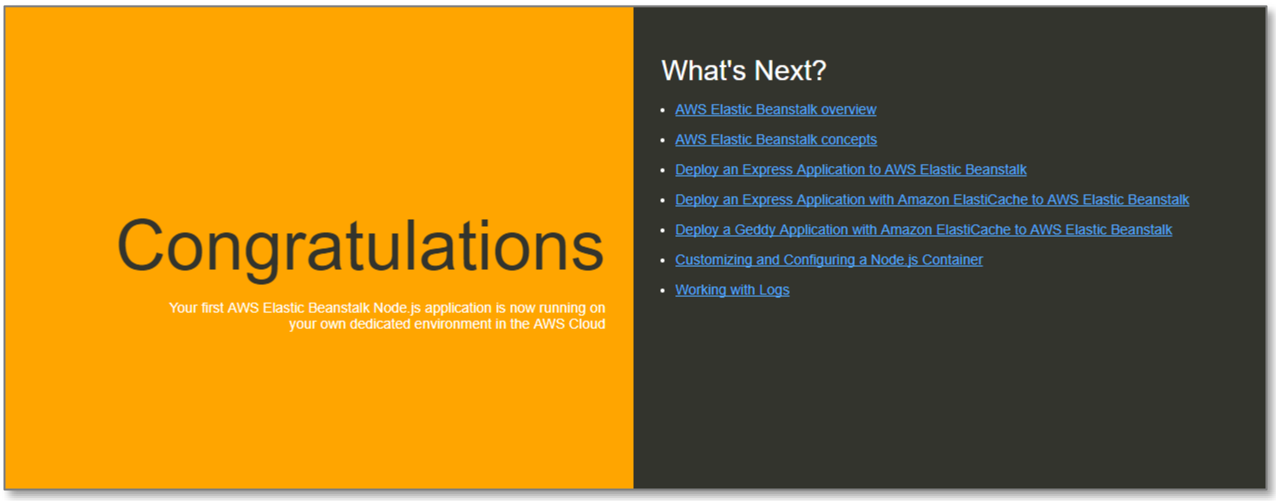
If you go back to the Code Pipeline console you will see the change taking effect.



When complete the change will appear in “Staging”

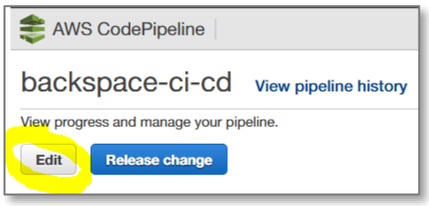


Go back to Elastic Beanstalk and click on the environment URL to see the updated version with the orange background

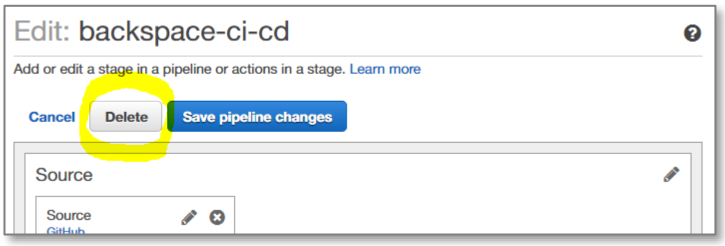

Congratulations
Your first AWS Elastic Beanstalk Node.js application is now running on your own dedicated environment in the AWS Cloud
What's Next?
AWS Elastic Beanstalk overview
AWS Elastic Beanstalk concepts
Deploy an Express Application to AWS Elastic Beanstalk
Deploy an Express Application with Amazon 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

Clean Up

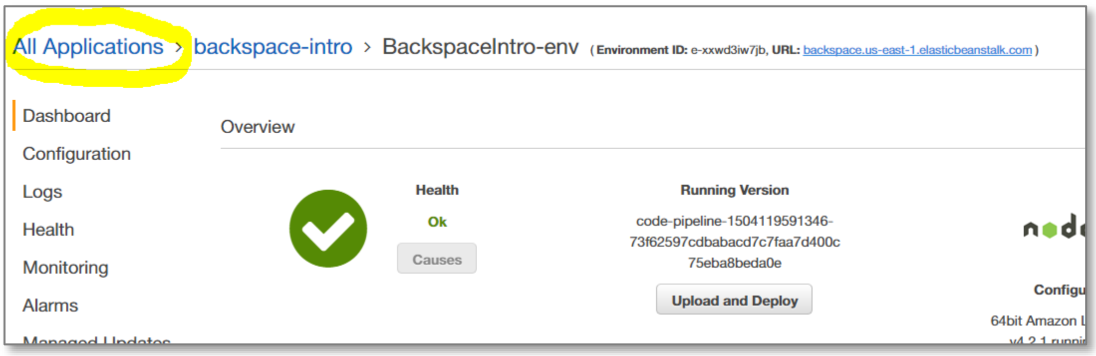
Go back to the Code Pipeline console and click “Edit”



Click “Delete”

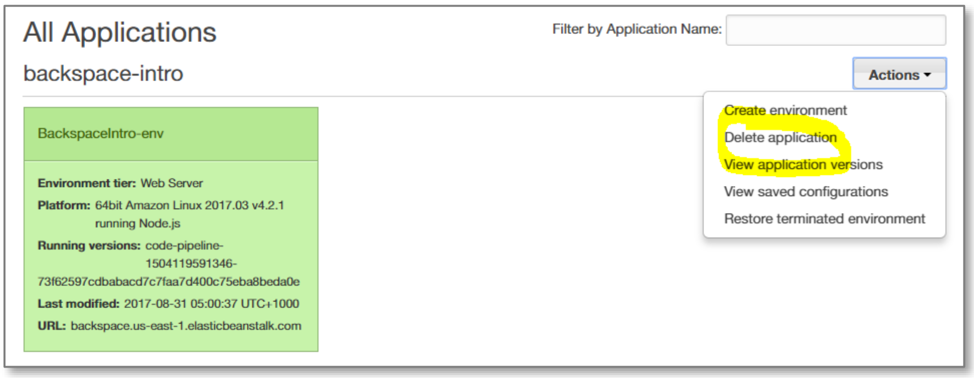


Go back to the Elastic Beanstalk console and delete the application.



(Environment ID: e-xxwd3iw7jb, URL: backspace.us-east-1.elasticbeanstalk.com)

Running Version: code-pipeline-1504119591346-73f62597cdbabacd7c7faa7d400c75eba8beda0e


All Applications
Filter by Application Name:
backspace-intro
Actions
Create environment
Delete application
View application versions
View saved configurations
Restore terminated environment
BackspaceIntro-env

Environment tier: Web Server
Platform: 64bit Amazon Linux 2017.03 v4.2.1 running Node.js
Running versions: code-pipeline-1504119591346-73f62597cdbabacd7c7faa7d400c75eba8beda0e
Last modified: 2017-08-31 05:00:37 UTC+1000
URL: backspace.us-east-1.elasticbeanstalk.com

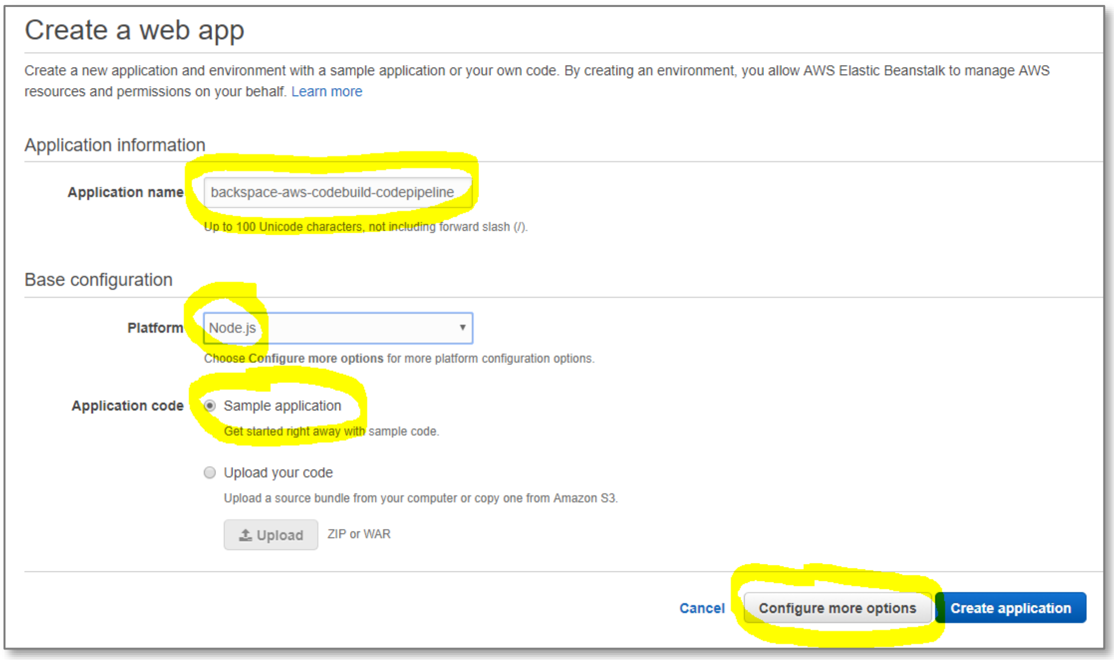
Integrating CI/CD Build and Test Processes with CodeBuild

In this section, we will integrate the CodeBuild Service with CodePipeline to create build and test processes in our Continuous Integration and Continuous Delivery (CI/CD) pipeline.

Deploy an Elastic Beanstalk Application

Enter details to create a NodeJS application called *backspace-aws-codebuild-codepipeline*.

Click "Configure more options"


Create a web app

Create a new application and environment with a sample application or your own code. By creating an environment, you allow AWS Elastic Beanstalk to manage AWS resources and permissions on your behalf. Learn more

Application information

Application name: backspace-aws-codebuild-codepipeline

Up to 100 Unicode characters, not including forward slash (/).

Base configuration

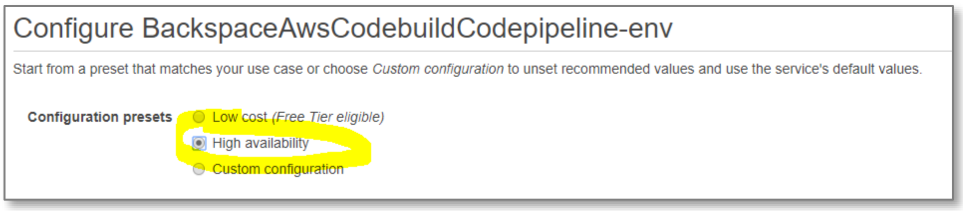
Platform: Node.js

Choose Configure more options for more platform configuration options.

Application code: [✓ Sample application] Get started right away with sample code.

Upload your code — Upload a source bundle from your computer or copy one from Amazon S3.

Select "High Availability"


Configure BackspaceAwsCodebuildCodepipeline-env

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: Low cost *(Free Tier eligible)* [✓ High availability] Custom configuration

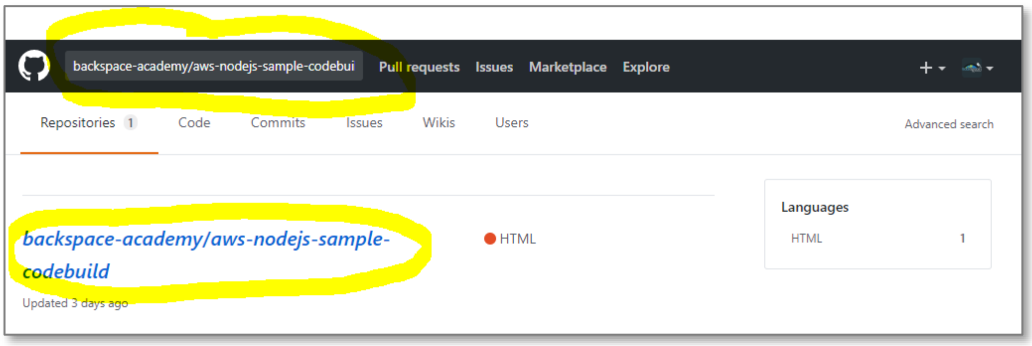
Click “Create app”

Fork a GitHub Repository

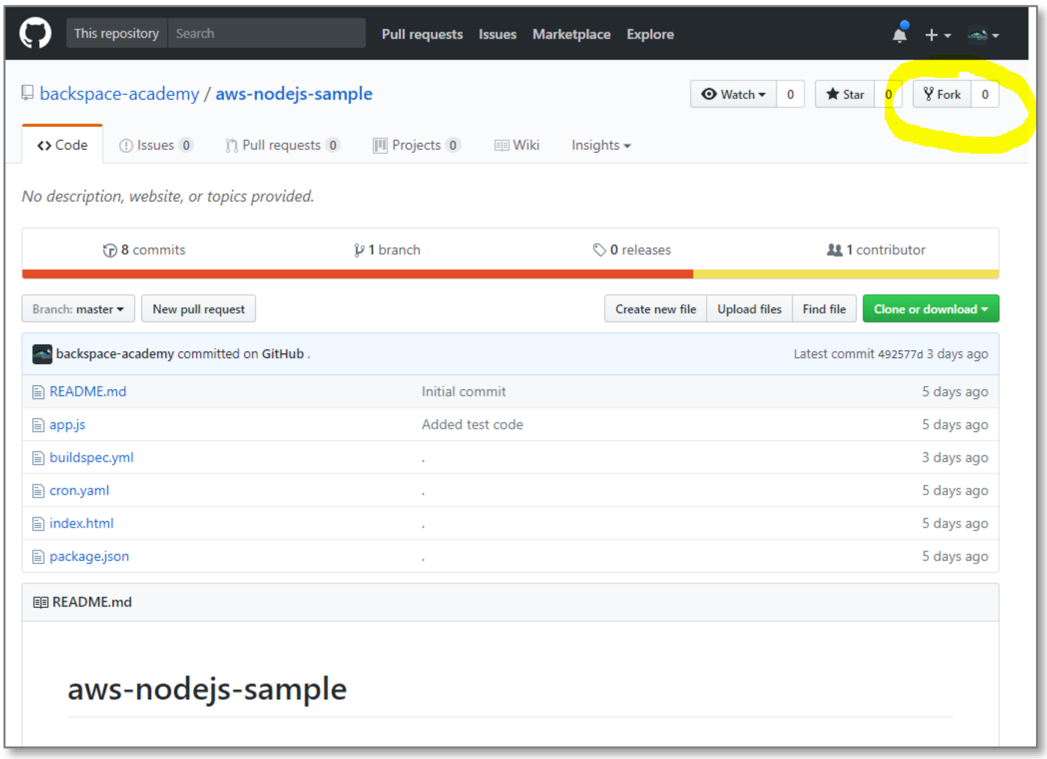
The code for this lab is located in a GitHub repository. We can save time and simply fork a copy of this to your GitHub account.

Sign in to your GitHub account.

Search for “backspace-academy/aws-nodejs-sample-codebuild”



Click on “Fork” to fork the repository.



backspace-academy / aws-nodejs-sample

No description, website, or topics provided.

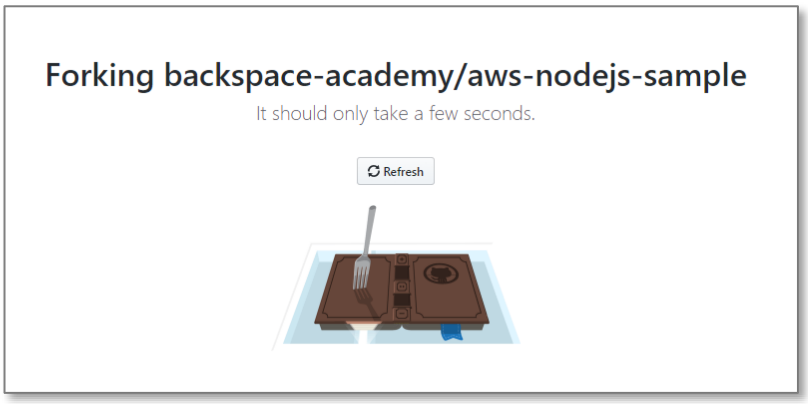
8 commits | 1 branch | 0 releases | 1 contributor

backspace-academy committed on GitHub . Latest commit 492577d 3 days ago

README.md	Initial commit	5 days ago
app.js	Added test code	5 days ago
buildspec.yml	.	3 days ago
cron.yaml	.	5 days ago
index.html	.	5 days ago
package.json	.	5 days ago

README.md

aws-nodejs-sample



Forking backspace-academy/aws-nodejs-sample

It should only take a few seconds.

You will now have a forked repository in your account that can be used to deploy the sample code.

AWS CodeBuild Files

The repository contains a modified copy of the AWS NodeJS sample application.

Additional files for AWS CodeBuild include:

test.js – code to test the application before deployment

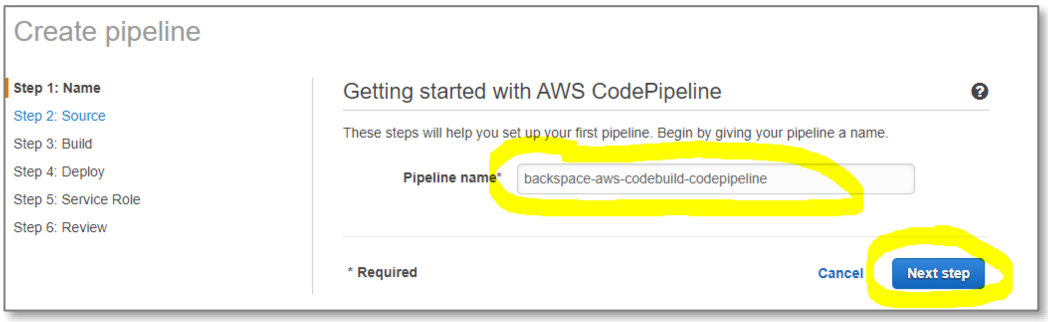
package.json – modified to include an “npm async” package dependency (so that we have something to build!)

buildspec.yaml (see: Build Specification Reference for AWS CodeBuild)

- Details build and test commands for CodeBuild
- Install package.json dependencies
- Install Mocha and UnitJS
- Test with Mocha
- Artifacts required for Elastic Beanstalk.

Create the CI/CD Pipeline

Go to the CodePipeline console and create a pipeleline named “backspace-aws-codebuild-codepipeline”


Create pipeline

Getting started with AWS CodePipeline

These steps will help you set up your first pipeline. Begin by giving your pipeline a name.

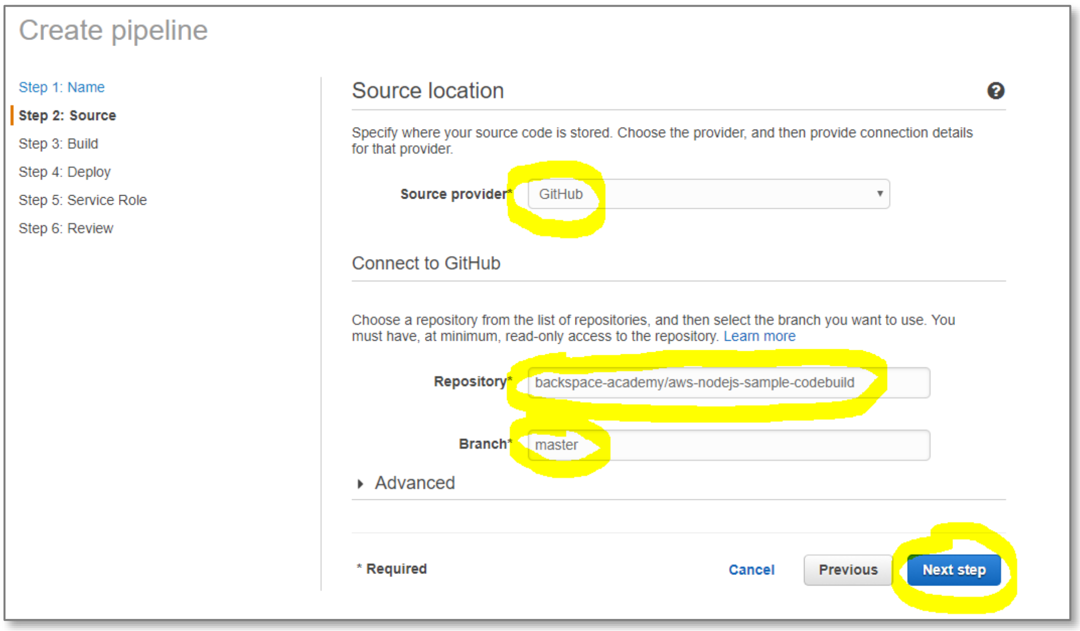
Pipeline name*: backspace-aws-codebuild-codepipeline

* Required

Select GitHub as Source provider.

Connect to GitHub and select the forked repository and master branch.

Click “Next step”


Create pipeline

Source location

Specify where your source code is stored. Choose the provider, and then provide connection details for that provider.

Source provider*: GitHub

Connect to GitHub

Choose a repository from the list of repositories, and then select the branch you want to use. You must have, at minimum, read-only access to the repository. Learn more

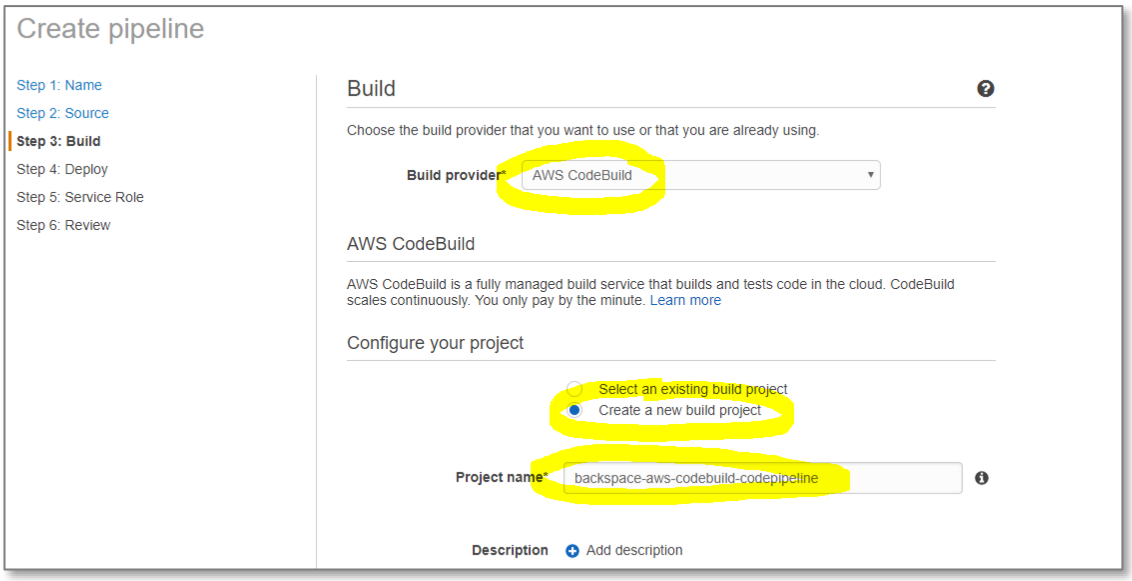
Repository*: backspace-academy/aws-nodejs-sample-codebuild

Branch*: master

* Required

Select “AWS CodeBuild” as the Build provider.

Select “Create a new build project” and name the project



Create pipeline

Build

Choose the build provider that you want to use or that you are already using.

Build provider* AWS CodeBuild

AWS CodeBuild

AWS CodeBuild is a fully managed build service that builds and tests code in the cloud. CodeBuild scales continuously. You only pay by the minute. Learn more

Configure your project

Select an existing build project [✓ Create a new build project]

Project name* backspace-aws-codebuild-codepipeline

Description Add description

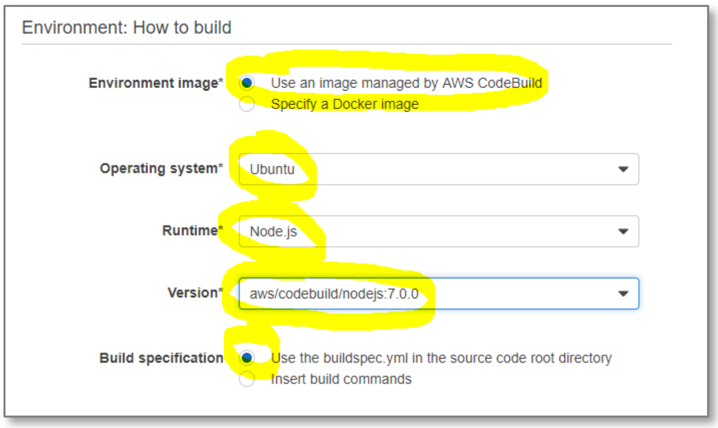
Select “Use an image managed by AWS CodeBuild”

Select “Ubuntu” for Operating system

Select “NodeJS” for Runtime

Select latest NodeJS version

Select “Use the buildspec.yml in the source code root directory”



Environment: How to build

Environment image* [✓ Use an image managed by AWS CodeBuild] Specify a Docker image

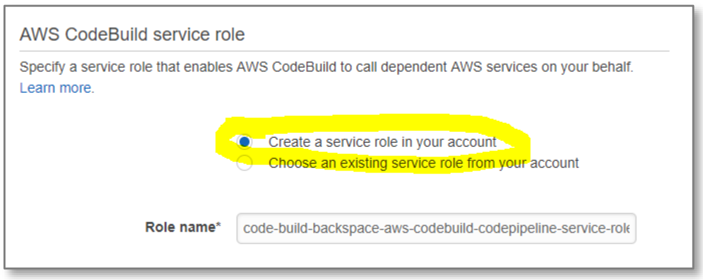
Operating system* Ubuntu

Runtime* Node.js

Version* aws/codebuild/nodejs:7.0.0

Build specification [✓ Use the buildspec.yml in the source code root directory] Insert build commands

Select “Create a service role in your account”



AWS CodeBuild service role

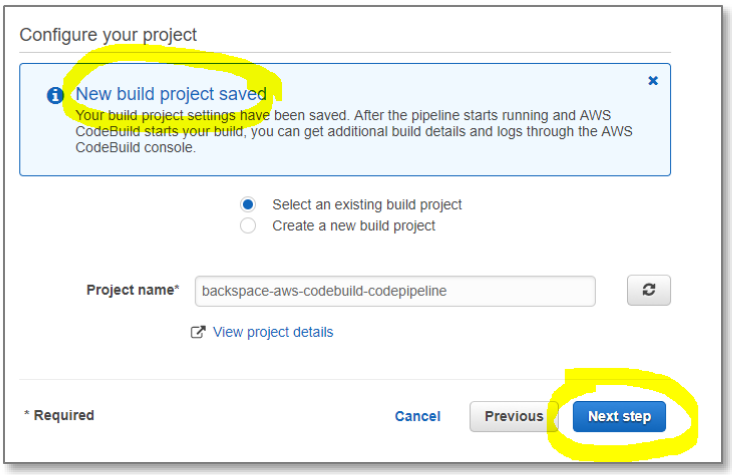
Specify a service role that enables AWS CodeBuild to call dependent AWS services on your behalf. Learn more.

[✓ Create a service role in your account]
Choose an existing service role from your account

Role name*	code-build-backspace-aws-codebuild-codepipeline-service-role

Click "Save Build Project"

After Build project has been saved click "Next step"



Configure your project

New build project saved
Your build project settings have been saved. After the pipeline starts running and AWS CodeBuild starts your build, you can get additional build details and logs through the AWS CodeBuild console.

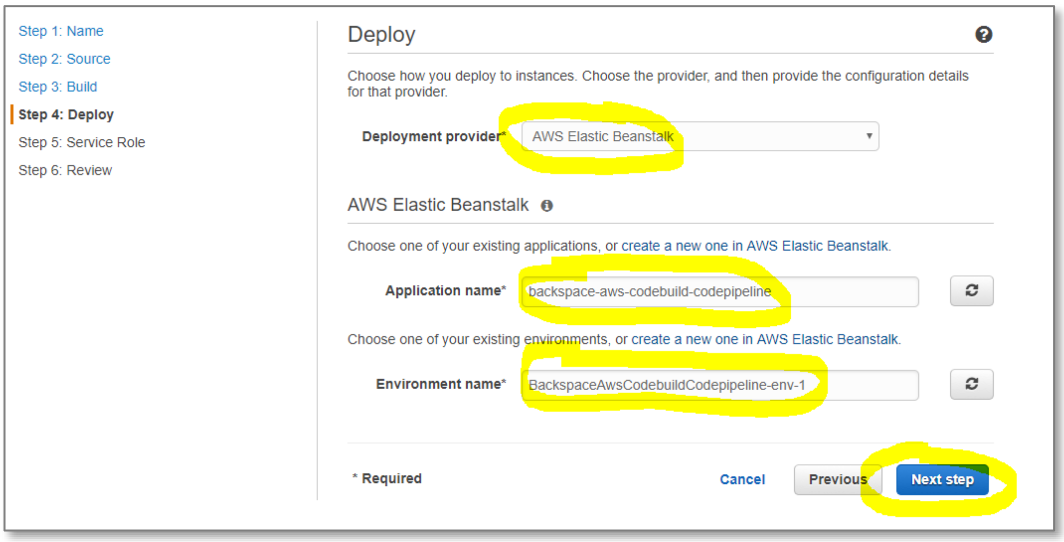
[✓ Select an existing build project]
Create a new build project

Project name*	backspace-aws-codebuild-codepipeline

Select "AWS Elastic Beanstalk" as Deployment provider.

Select the application and environment created by Elastic Beanstalk.

Click "Next step"



Deploy

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

Deployment provider*	AWS Elastic Beanstalk

AWS Elastic Beanstalk

Choose one of your existing applications, or create a new one in AWS Elastic Beanstalk.

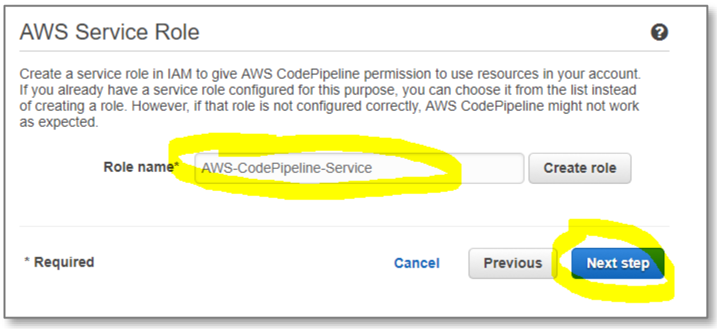
Application name*	backspace-aws-codebuild-codepipeline

Choose one of your existing environments, or create a new one in AWS Elastic Beanstalk.

Environment name*	BackspaceAwsCodebuildCodepipeline-env-1

Select the service role created for the previous lab

Click “Next step”

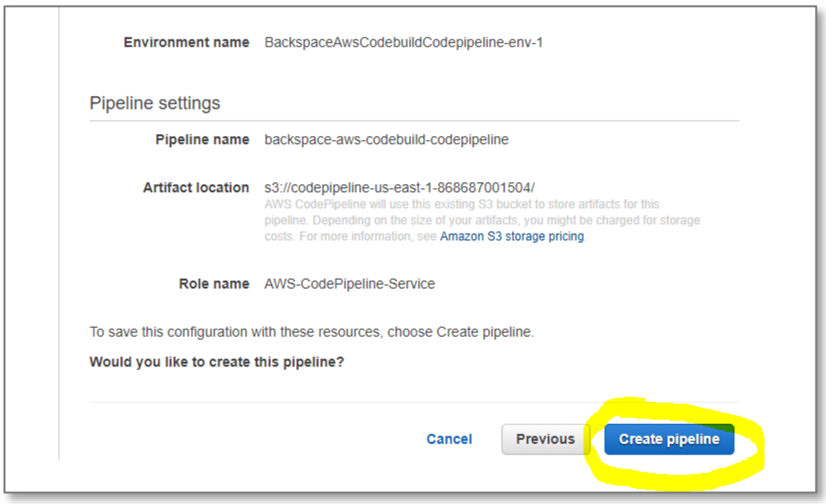


AWS Service Role

Create a service role in IAM to give AWS CodePipeline permission to use resources in your account. If you already have a service role configured for this purpose, you can choose it from the list instead of creating a role. However, if that role is not configured correctly, AWS CodePipeline might not work as expected.

Role name* AWS-CodePipeline-Service

Click “Create pipeline”



Environment name BackspaceAwsCodebuildCodepipeline-env-1

Pipeline settings

Pipeline name backspace-aws-codebuild-codepipeline

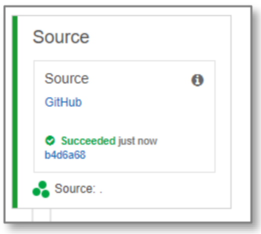
Artifact location s3://codepipeline-us-east-1-868687001504/
AWS CodePipeline will use this existing S3 bucket to store artifacts for this pipeline. Depending on the size of your artifacts, you might be charged for storage costs. For more information, see Amazon S3 storage pricing

Role name AWS-CodePipeline-Service

To save this configuration with these resources, choose Create pipeline.

Would you like to create this pipeline?

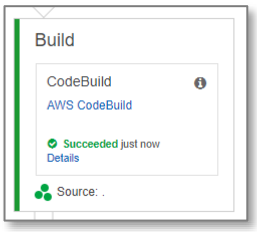
Your source will pulled from GitHub:



Source

Source
GitHub
Succeeded just now
b4d6a68
Source: .

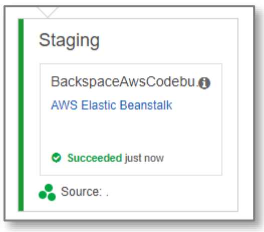
Your application will then be built and tested by CodeBuild:



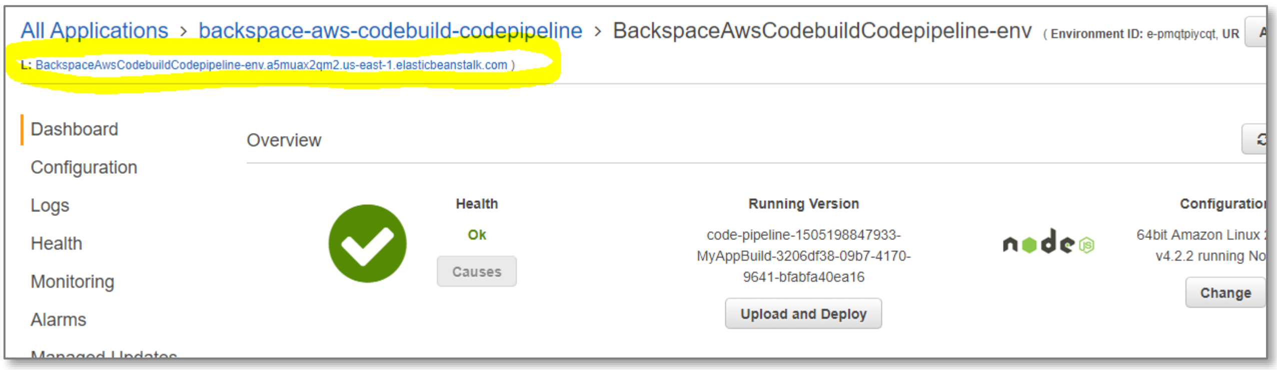
Build

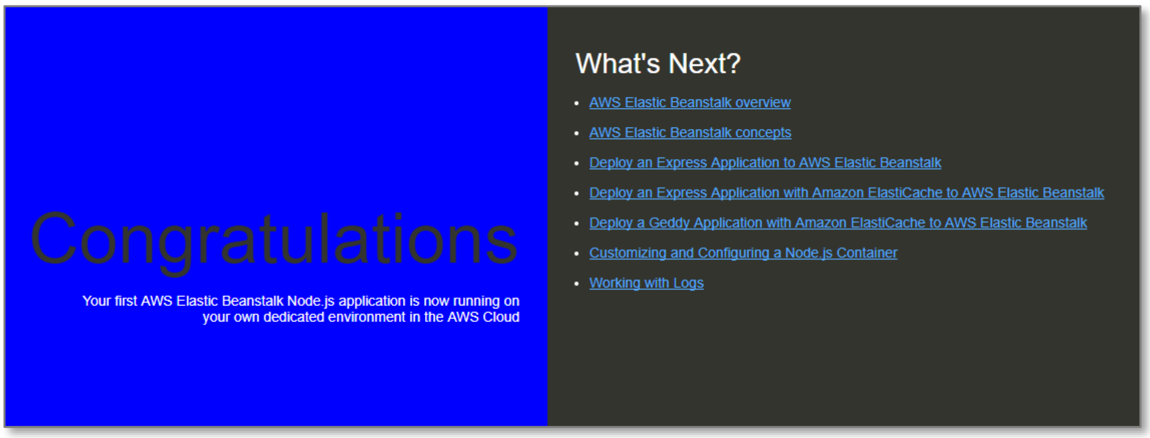
CodeBuild
AWS CodeBuild
Succeeded just now
Details
Source: .

Your application will then be staged:



Go back to the Elastic Beanstalk console to view the updated application.


All Applications > backspace-aws-codebuild-codepipeline > BackspaceAwsCodebuildCodepipeline-env (Environment ID: e-pmqtpiycqt, UR
L: BackspaceAwsCodebuildCodepipeline-env.a5muax2qm2.us-east-1.elasticbeanstalk.com)


Congratulations

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

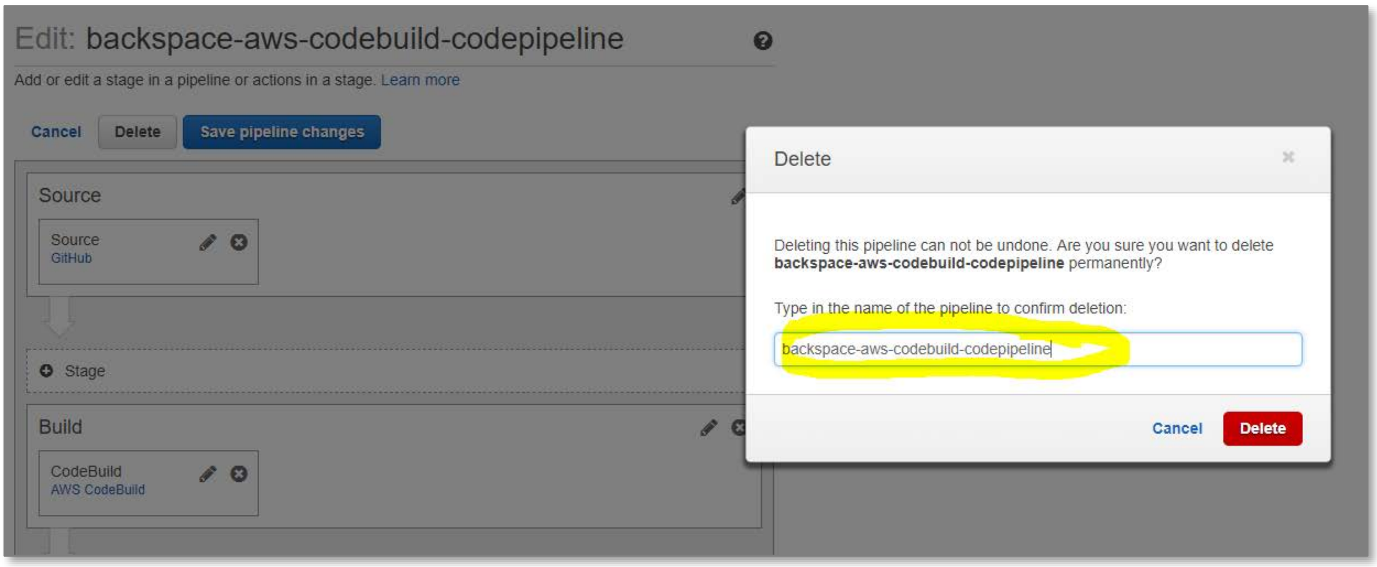
What's Next?

- 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

Clean Up

Go back to the Code Pipeline console and click “Edit”

Click “Delete”


Edit: backspace-aws-codebuild-codepipeline

Add or edit a stage in a pipeline or actions in a stage. Learn more

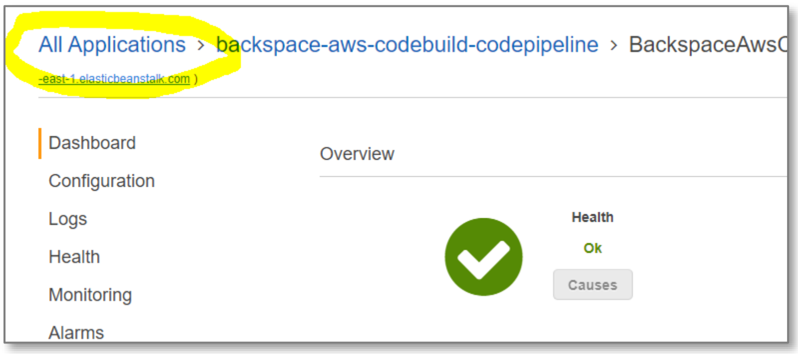
Delete

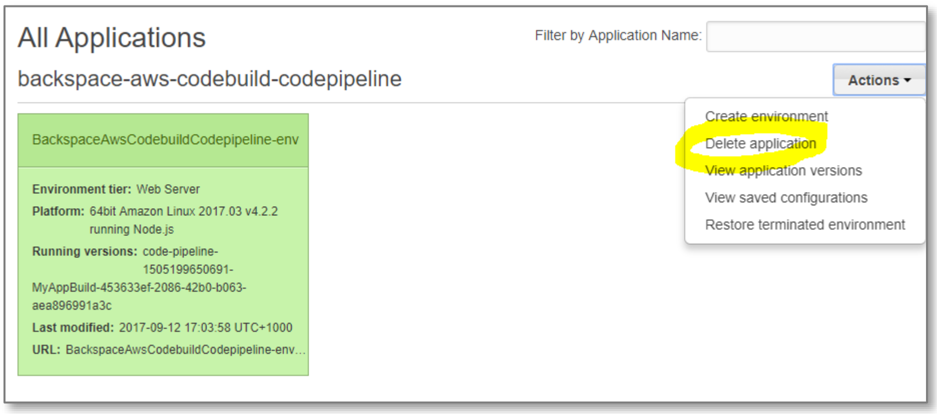
Deleting this pipeline can not be undone. Are you sure you want to delete **backspace-aws-codebuild-codepipeline** permanently?

Type in the name of the pipeline to confirm deletion:

backspace-aws-codebuild-codepipeline

Go back to the Elastic Beanstalk console and delete the application.


All Applications > backspace-aws-codebuild-codepipeline > BackspaceAwsC
-east-1.elasticbeanstalk.com)
Dashboard
Configuration
Logs
Health
Monitoring
Alarms
Overview
Health
Ok
Causes


All Applications

Filter by Application Name:

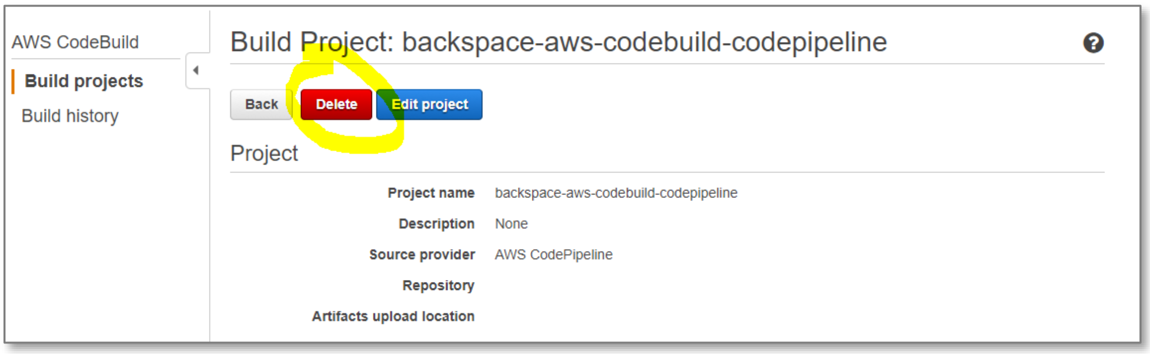
backspace-aws-codebuild-codepipeline

Actions
Create environment
Delete application
View application versions
View saved configurations
Restore terminated environment

BackspaceAwsCodebuildCodepipeline-env

Environment tier: Web Server
Platform: 64bit Amazon Linux 2017.03 v4.2.2 running Node.js
Running versions: code-pipeline-1505199650691-MyAppBuild-453633ef-2086-42b0-b063-aea896991a3c
Last modified: 2017-09-12 17:03:58 UTC+1000
URL: BackspaceAwsCodebuildCodepipeline-env...

Go Back to AWS CodeBuild and delete the project



Build Project: backspace-aws-codebuild-codepipeline

Project

Project name	backspace-aws-codebuild-codepipeline
Description	None
Source provider	AWS CodePipeline
Repository	
Artifacts upload location	