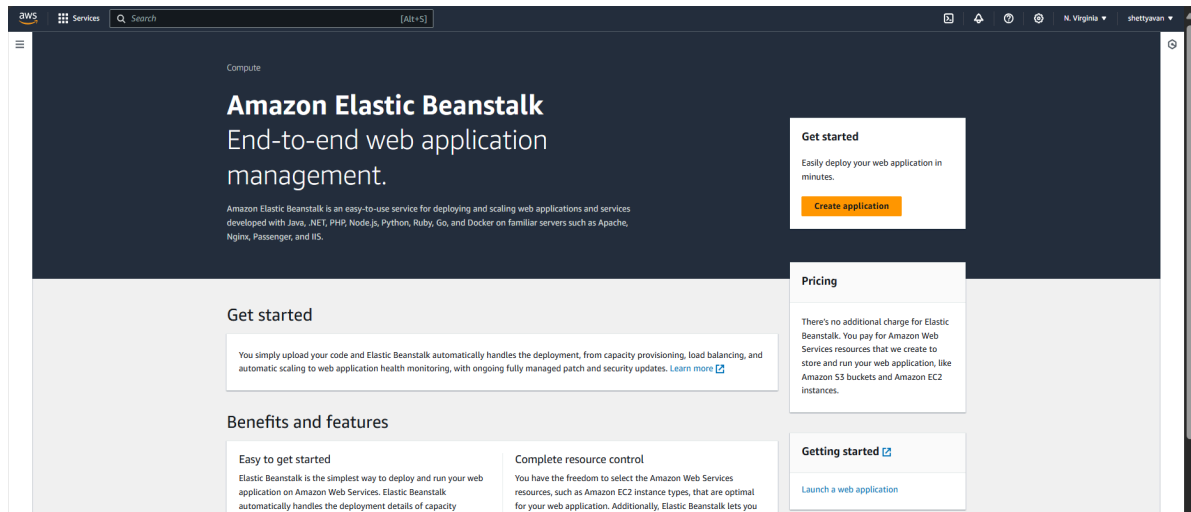


1. Open the aws console and then search Elastic Beanstalk (Opens a dashboard as seen below)



2. Click on create application and configure the environment by adding your application name

The screenshot shows the 'Configure environment' wizard in the AWS Elastic Beanstalk console. The wizard has six steps: 'Step 1: Configure environment', 'Step 2: Configure service access', 'Step 3 - optional: Set up networking, database, and tags', 'Step 4 - optional: Configure instance traffic and scaling', 'Step 5 - optional: Configure updates, monitoring, and logging', and 'Step 6: Review'. The current step is 'Step 1: Configure environment'. The main content area is divided into three sections: 'Environment tier', 'Application information', and 'Environment information'. The 'Environment tier' section has two options: 'Web server environment' (selected) and 'Worker environment'. The 'Application information' section has a text input field for 'Application name' with the value 'my-web-app'. The 'Environment information' section has a text input field for 'Environment name' with the value 'My-web-env' and a text input field for 'Domain' with the value 'Leave blank for autogenerated value'. There is a 'Check availability' button at the bottom right.

### 3. Choose PHP from the drop-down menu and click next

**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**

PHP

**Platform branch**

PHP 8.3 running on 64bit Amazon Linux 2023

**Platform version**

4.3.1 (Recommended)

**Application code** [Info](#)

- ☒ **Sample application**
- ☐ **Existing version**  
Application versions that you have uploaded.
- ☐ **Upload your code**  
Upload a source bundle from your computer or copy one from Amazon S3.

**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**

### 4. In this step you have to create a key pair

- Go to EC2 instance tab and from the left panel create a key pair. As the key pair might be useful for the further process

EC2 > Key pairs > Create key pair

## Create key pair [Info](#)

**Key pair**  
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

**Name**  
  
The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

**Key pair type** [Info](#)  
☒ RSA ☐ ED25519

**Private key file format**  
☒ .pem  
For use with OpenSSH  
☐ .ppk  
For use with PuTTY

**Tags - optional**  
No tags associated with the resource.  
  
You can add up to 50 more tags.

- b. In the same fashion go to IAM and then under roles section click create role and then Select AWS Service and under instances select EC2. Your EC2 Instance role is created

## Select trusted entity [Info](#)

**Trusted entity type**

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

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

☐ **Web identity**  
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.

☐ **SAML 2.0 federation**  
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

☐ **Custom trust policy**  
Create a custom trust policy to enable others to perform actions in this account.

**Use case**  
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case

5. Now comeback to Elastic Beanstalk page and from the drop down menu select the newly created key pair and instance profile

Step 1  
[Configure environment](#)

Step 2  
**Configure service access**

Step 3 - optional  
[Set up networking, database, and tags](#)

Step 4 - optional  
[Configure instance traffic and scaling](#)

Step 5 - optional  
[Configure updates, monitoring, and logging](#)

Step 6  
[Review](#)

## 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](#)

**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

## 6. No changes further in the Configure instance traffic section

### Configure instance traffic and scaling - optional Info

**Instances** Info

Configure the Amazon EC2 instances that run your application.

**Root volume (boot device)**

**Root volume type**

**Size**

The number of gigabytes of the root volume attached to each instance.

GB

**IOPS**

Input/output operations per second for a provisioned IOPS (SSD) volume.

IOPS

**Throughput**

The desired throughput to provision for the Amazon EBS root volume attached to your environment's EC2 instance

MiB/s

**Amazon CloudWatch monitoring**

The time interval between when metrics are reported from the EC2 instances

**Monitoring interval**

**Instance metadata service (IMDS)**

Your environment's platform supports both IMDSv1 and IMDSv2. To enforce IMDSv2, deactivate IMDSv1. [Learn more](#)

**IMDSv1**

With the current setting, the environment enables only IMDSv2.

☒ Deactivated

**Capacity rebalancing**  
Specifies whether to enable the capacity rebalancing feature for Spot Instances in your Auto Scaling Group. This option is only relevant when `EnableSpot` is true in the `aws:ec2:instances` namespace, and there is at least one Spot Instance in your Auto Scaling group.

☐ Turn on capacity rebalancing

**Architecture**  
The processor architecture determines the instance types that are made available. You can't change this selection after you create the environment. [Learn more](#)

☒ **x86\_64**  
This architecture uses x86 processors and is compatible with most third-party tools and libraries.

☐ **arm64 - new**  
This architecture uses AWS Graviton2 processors. You might have to recompile some third-party tools and libraries.

**Instance types**  
Add instance types for your fleet. Change the order that the instances are in to set the preferred launch order. This only affects On-Demand instances. We recommend you include at least two instance types. [Learn more](#)

Choose x86 instance types

t3.micro X t3.small X

**AMI ID**  
Elastic Beanstalk selects a default Amazon Machine Image (AMI) for your environment based on the Region, platform version, and processor architecture that you choose. [Learn more](#)

ami-083f545ce1a73bf03

**Availability Zones**  
Number of Availability Zones (AZs) to use.

Any

**Placement**  
Specify Availability Zones (AZs) to use.

Choose Availability Zones (AZs)

**Scaling cooldown**

360 seconds

Cancel

Skip to review

Previous

Next

7. Again no changes in the configure updates, monitoring, and logging part just click on next.

**Configure updates, monitoring, and logging - optional** [Info](#)

**Monitoring** [Info](#)

**Health reporting**  
Enhanced health reporting provides free real-time application and operating system monitoring of the instances and other resources in your environment. The `EnvironmentHealth` custom metric is provided free with enhanced health reporting. Additional charges apply for each custom metric. For more information, see [Amazon CloudWatch Pricing](#)

**System**  
☐ Basic  
☒ Enhanced

CloudWatch Custom Metrics - Instance  
Choose metrics

CloudWatch Custom Metrics - Environment  
Choose metrics

**Health event streaming to CloudWatch Logs**  
Configure Elastic Beanstalk to stream environment health events to CloudWatch Logs. You can set the retention up to a maximum of ten years and configure Elastic Beanstalk to delete the logs when you terminate your environment.

**Log streaming**  
☐ Activated (standard CloudWatch charges apply.)

**Retention**  
7

**Lifecycle**  
Keep logs after terminating environment

**Managed platform updates** [Info](#)

Activate managed platform updates to apply platform updates automatically during a weekly maintenance window that you choose. Your application stays available during the update process.

### Amazon X-Ray

Amazon X-Ray is a service that collects data about the requests and responses that your application serves and receives. You can use the tools that X-Ray offers to view and filter the data that it provides to identify potential issues and optimization opportunities.

**X-Ray daemon**  
(service charges may apply.)

☐ Activated

### S3 log storage

Configure the instances in your environment to upload rotated logs to Amazon S3. [Learn more](#)

**Rotate logs**  
(standard S3 charges apply.)

☐ Activated

### Instance log streaming to CloudWatch logs

Configure the instances in your environment to stream logs to CloudWatch logs. You can set the retention to up to 10 years and configure Elastic Beanstalk to delete the logs when you terminate your environment. [Learn more](#)

**Log streaming**  
(standard CloudWatch charges apply.)

☐ Activated

**Retention**

7

**Lifecycle**

Keep logs after terminating envir...

### Environment properties

The following properties are passed in the application as environment properties. [Learn more](#)

No environment properties have been configured.

[Add environment property](#)

CancelPreviousNext

## 8. Now review the changes made and click on create application

### Review

**Step 1: Configure environment** [Edit](#)

#### Environment information

Environment tier	Application name
Web server environment	my-web-app
Environment name	Application code
My-web-env	Sample application
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	

**Step 2: Configure service access** [Edit](#)

#### Service access

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 key pair	EC2 instance profile
arn:aws:iam::934505240185:role/EMR_EC2_DefaultRole	vockey	EMR_EC2_DefaultRole

**Step 3: Set up networking, database, and tags** [Edit](#)

#### Networking, database, and tags

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

**Step 5: Configure updates, monitoring, and logging** Edit

**Updates, monitoring, and logging** [Info](#)  
Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.

**Monitoring**

System enhanced	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
—	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false

**Updates**

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	
false	false	

**Platform software**

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

**Environment properties**

Key	Value
No environment properties	
There are no environment properties defined	

Cancel Previous Submit

9. Your sample environment is created for you to deploy your application. By default, it creates an EC2 instance, a security group, an Auto Scaling group, an Amazon S3 Bucket, Amazon CloudWatch alarms and a domain name for your Application.

Environment successfully launched.

[Elastic Beanstalk](#) > [Environments](#) > My-web-env

**My-web-env** [Info](#) Actions Upload and deploy

**Environment overview**

Health	Environment ID
<span>Warning</span>	e-adppnnmsq
Domain	Application name
My-web-env.eba-m6b5xmgv.us-east-1.elasticbeanstalk.com	my-web-app

**Platform** Change version

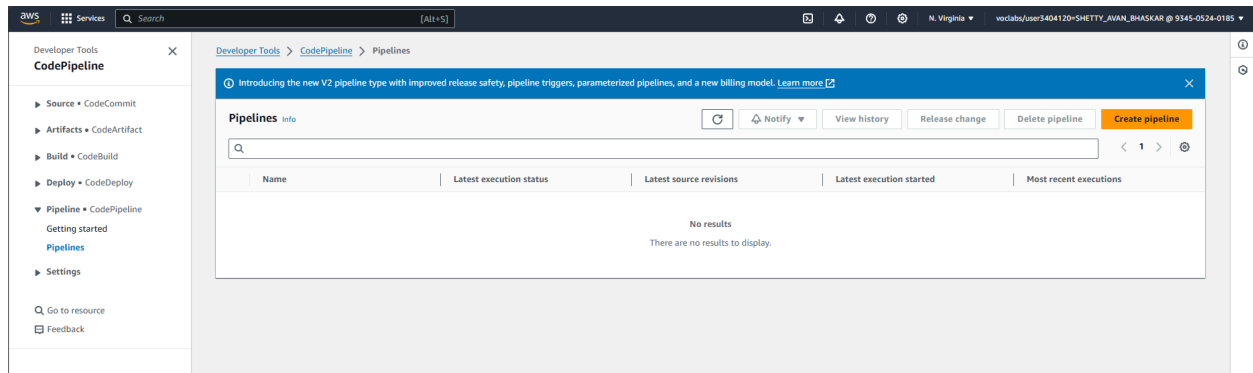
Platform	
PHP 8.3 running on 64bit Amazon Linux 2023/4.3.1	
Running version	Platform state
—	<span>Supported</span>

## Pipeline creation:

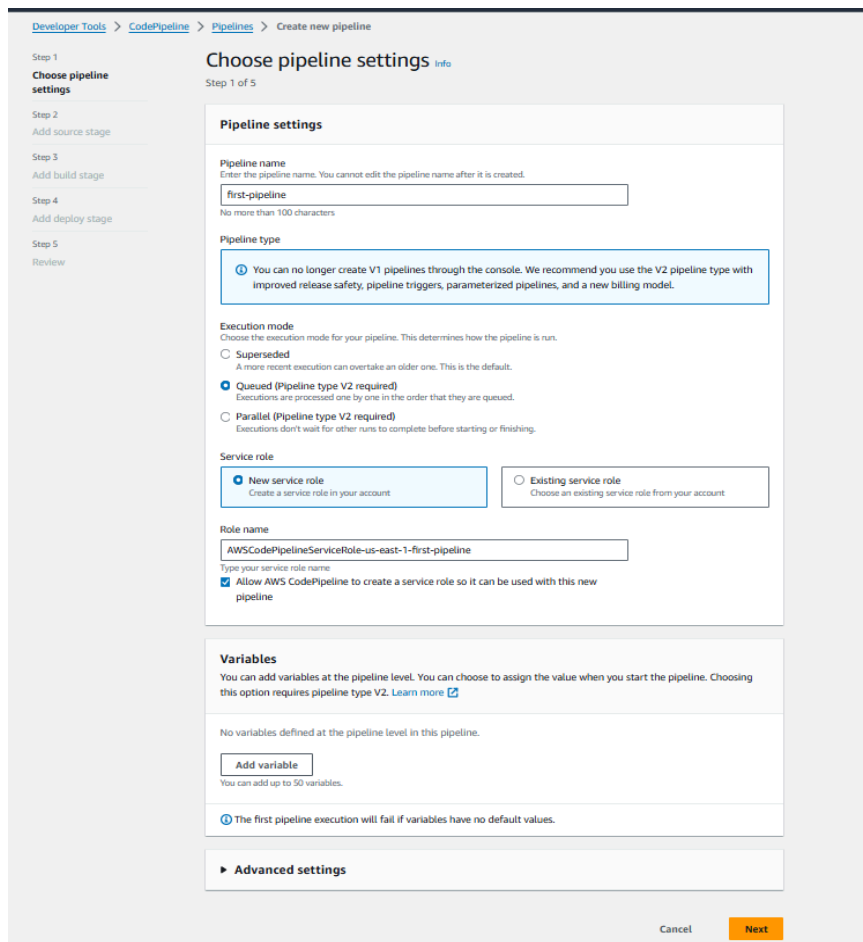
1. Fork a github repo for aws codepipeline available as The pipeline takes code from the source and then performs actions on it. We don't need to code from scratch in this manner



2. Go to developer tools and select CodePipeline and create a new pipeline



3. Name your pipeline and select the desired service role





4. In the source stage select Github v2 as the provider and then connect your github connect so that the pipeline can access the forked source code .

- a. For this purpose create aws github connection and with your credentials install the AWS under the forked repository

Developer Tools > Connections > Create connection

Beginning July 1, 2024, the console will create connections with codeconnections in the resource ARN. Resources with both service prefixes will continue to display in the console. [Learn more](#)

## Connect to GitHub

**GitHub connection settings** info

Connection name  
avan-github

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

Q 53621842 X or Install a new app

Tags - optional

Connect

## Install AWS Connector for GitHub

Install on your personal account Avan Shetty

for these repositories:

☒ **All repositories**  
This applies to all current and future repositories owned by the resource owner. Also includes public repositories (read-only).

☐ **Only select repositories**  
Select at least one repository. Also includes public repositories (read-only).

with these permissions:

- ✓ Read access to issues and metadata
- ✓ Read and write access to administration, code, commit statuses, pull requests, and repository hooks

Install Cancel

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

5. Once the connection is established from the drop down menu select the repository and the branch

### Add source stage Info

Step 2 of 5

#### Source

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

GitHub (Version 2) ▼

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

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

arn:aws:codeconnections:us-east-1:014498640047:connection/Zabf8fba-2c4 ✕ or [Connect to GitHub](#)

**Ready to connect**  
Your GitHub connection is ready for use.

**Repository name**  
Choose a repository in your GitHub account.

avanshh99/aws-codepipeline-s3-codedeploy-linux-2.0 ✕  
You can type or paste the group path to any project that the provided credentials can access. Use the format 'group/subgroup/project'.

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

master ✕

**Output artifact format**  
Choose the output artifact format.

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

☐ **Full clone**  
AWS CodePipeline passes metadata about the repository that allows subsequent actions to do a full Git clone. Only supported for AWS CodeBuild actions.

6. Skip the build stage part as we are not plugging in any build provider and in choose Beanstalk as the Deploy Provider, same region as the Bucket and Beanstalk, name and environment name.

## Add deploy stage [Info](#)

Step 4 of 5

**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) ▼

**Input artifacts**  
Choose an input artifact for this action. [Learn more](#) [🔗](#)

▼

No more than 100 characters

**Application name**  
Choose an application that you have already created in the AWS Elastic Beanstalk console. Or create an application in the AWS Elastic Beanstalk console and then return to this task.

Q my-web-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 My-web-app-env X

☐ Configure automatic rollback on stage failure

Cancel Previous **Next**

## 7. Review the settings and click on create pipeline

## Review [Info](#)

Step 5 of 5

### Step 1: Choose pipeline settings

**Pipeline settings**

Pipeline name  
firstpipeline

Pipeline type  
V2

Execution mode  
QUEUED

Artifact location  
A new Amazon S3 bucket will be created as the default artifact store for your pipeline

Service role name  
AWSCodePipelineServiceRole-us-east-1-firstpipeline

**Variables**

Name	Default value	Description
No variables		
No variables defined at the pipeline level in this pipeline.		

### Step 2: Add source stage

**Source action provider**

Source action provider  
GitHub (Version 2)

OutputArtifactFormat

**Step 2: Add source stage**

Source action provider

Source action provider  
GitHub (Version 2)  
OutputArtifactFormat  
CODE\_ZIP  
DetectChanges  
true  
FullRepositoryId  
avanshh99/aws-codepipeline-s3-codedeploy-linux-2.0  
Default branch  
master  
ConnectionArn  
arn:aws:codeconnections:us-east-1:014498640047:connection/2abf8fba-2c48-4737-a578-0dc8486ca8f4

Trigger configuration  
You can add additional pipeline triggers after the pipeline is created.

Trigger type  
No filter

**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  
my-web-app  
EnvironmentName  
My-web-app-env  
Configure automatic rollback on stage failure  
Disabled

Cancel Previous Create pipeline

8. Once the Successfully created message appears, your pipeline is created. Then go ahead and check the URL provided in the EBS environment.

Success  
Congratulations! The pipeline firstpipeline has been created.

Create a notification rule for this pipeline

Developer Tools > CodePipeline > Pipelines > firstpipeline

**firstpipeline**

Pipeline type: V2 Execution mode: QUEUED

Notify Edit Stop execution Clone pipeline Release change

Source  
Succeeded  
Pipeline execution ID: cf56b21b-a739-4461-a00a-a26d7579bdf3

Source  
GitHub (Version 2)  
Succeeded - 2 minutes ago  
#cf56b21b  
View details

#cf56b21b Source: Update README.md

Disable transition

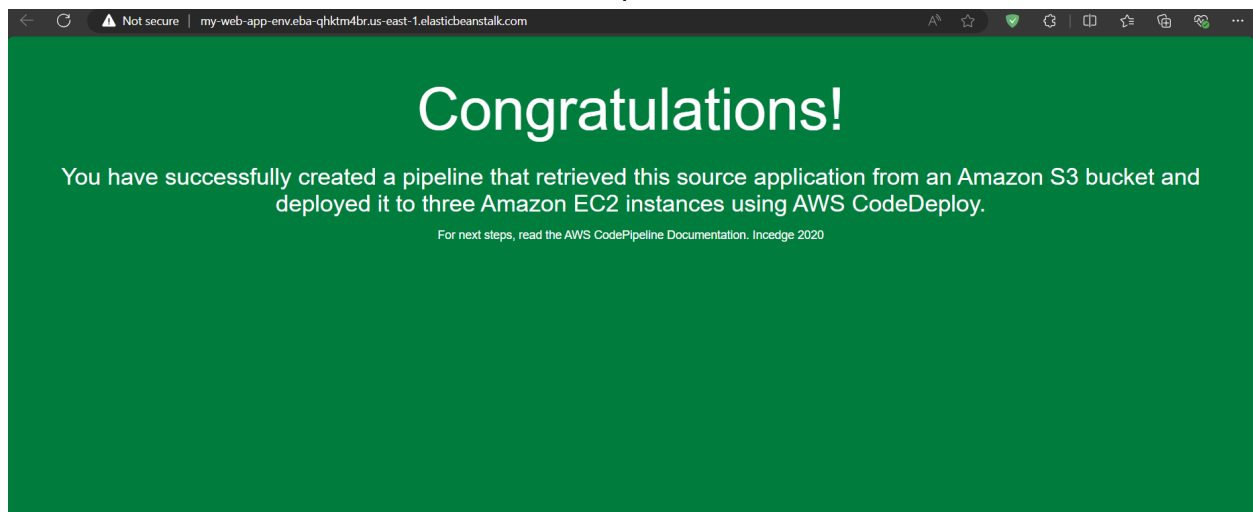
Deploy  
Succeeded  
Pipeline execution ID: cf56b21b-a739-4461-a00a-a26d7579bdf3

Deploy  
AWS Elastic Beanstalk  
Succeeded - 1 minute ago  
View details

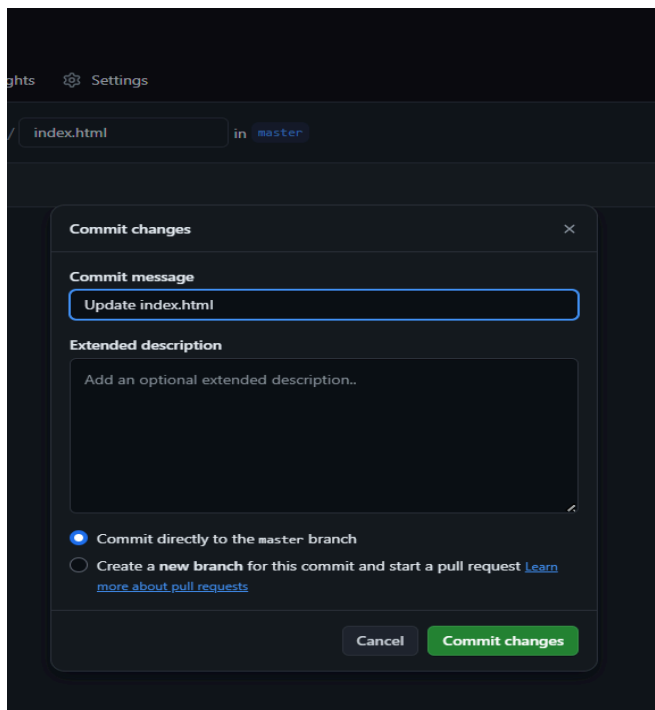
#cf56b21b Source: Update README.md

Start rollback

9. This is the website hosted from that forked repo in our beanstalk environment



10. Go to the repository and make the changes in the index.html file and commit them



11. The changes that are committed can be noticed in the source panel in real time and to view the changes check the url (refresh it) and you can view the changes once the deployment section shows success.

Developer Tools > CodePipeline > Pipelines > firstpipeline

firstpipeline

Pipeline type: V2 Execution mode: QUEUED

Source Succeeded

Pipeline execution ID: ad61d044-2167-4ac1-8c78-e1b10a8ab799

Source

GitHub (Version 2) [View details](#)

Succeeded - Just now

892b72fa [View details](#)

View details

892b72fa [View details](#) Source: Update index.html

Disable transition

Deploy Succeeded

Pipeline execution ID: ad61d044-2167-4ac1-8c78-e1b10a8ab799

Deploy

AWS Elastic Beanstalk [View details](#)

Succeeded - Just now

View details

892b72fa [View details](#) Source: Update index.html

Start rollback

Not secure | my-web-app-env.eba-qhktm4br.us-east-1.elasticbeanstalk.com

# Hello this is my first deployment D15C-52!

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

For next steps, read the AWS CodePipeline Documentation. Incedge 2020