

AWS DevOps Project Solution

Document

edureka!

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Creating Policies For New User To Have All Admin Or Limited Privileges

DEMO Steps:

Step1: Create the **CloudFormation** template using below zip file to create the Environment.

EdurekaDevOps CICD IAC Template

Extract the files separately, create your own bucket and upload all these files to your bucket

The screenshot shows the Amazon S3 console with a bucket named 'edureka-devops'. The 'Management' tab is selected. A search bar contains the placeholder 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'Download', and 'Actions'. A table lists six files:

Name	Last modified	Size
deployment-pipeline.yaml	Apr 23, 2019 4:11:25 PM GMT+0530	5.7 KB
ecs-cluster.yaml	Apr 23, 2019 4:11:25 PM GMT+0530	4.4 KB
ecs-continuous-deployment-mainstack.yaml	Apr 23, 2019 4:35:04 PM GMT+0530	4.1 KB
load-balancer.yaml	Apr 23, 2019 4:11:25 PM GMT+0530	2.0 KB
service.yaml	Apr 23, 2019 4:11:25 PM GMT+0530	3.2 KB
vpc.yaml	Apr 23, 2019 4:11:26 PM GMT+0530	1.8 KB

Open the mainstack yaml file to copy the S3 Url

The screenshot shows the 'Properties' tab for the 'ecs-continuous-deployment-mainstack.yaml' file. The 'Properties' tab is selected. Below it are buttons for 'Open', 'Download', 'Download as', 'Make public', and 'Copy path'. The file details are listed:

- Owner: support
- Last modified: Apr 23, 2019 4:35:04 PM GMT+0530
- Etag: 4ce5d149a954148ef80c96b8ffaaa2ec
- Storage class: Standard
- Server-side encryption: None
- Size: 4.1 KB
- Key: ecs-continuous-deployment-mainstack.yaml
- Object URL: <https://s3.amazonaws.com/edureka-devops/ecs-continuous-deployment-mainstack.yaml>

AWS DevOps Project Solution

a) Open the CloudFormation template and paste the S3 URL.

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

Template is ready Use a sample template Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file

Amazon S3 URL
`https://s3.amazonaws.com/edureka-devops/ecs-continuous-deployment-mainstack.yaml`

Amazon S3 template URL

S3 URL: `https://s3.amazonaws.com/edureka-devops/ecs-continuous-deployment-mainstack1.yaml` [View in Designer](#)

(You can click on View in Designer to see the overview of services being used)

File: 'template1'

Resource types

- ▶ AmazonMQ
- ▶ ApiGateway
- ▶ ApiGatewayV2
- ▶ AppMesh
- ▶ AppStream
- ▶ AppSync
- ▶ ApplicationAutoScaling
- ▶ Athena
- ▶ AutoScaling

Diagram

```
graph TD; VPC[VPC Stack] --> Cluster[Cluster Stack]; VPC --> Service[Service Stack]; LoadBalancing[LoadBalancing Stack] --> Cluster; LoadBalancing --> Service; Deployment[Deployment Stack] --> Service;
```

template1 [Edit](#)

Choose template language: **JSON** **YAML** [?](#)

```
1 ---  
2 AWSTemplateFormatVersion: 2010-09-09  
3  
4  
5 Description: >  
6 This template shows how to use AWS CodePipeline and AWS CodeBuild to build an  
7 automated continuous deployment pipeline to Amazon Elastic Container Service  
8 (Amazon ECS) using clusters powered by AWS Fargate or Amazon Elastic Compute  
9 Cloud (Amazon EC2).
```

b) Give stack name, keep the default auto populated values and fill the empty fields

Stack name

Stack name
 edurekaNVCloudformation

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

GitHub Configuration

Repo
The repo name of the sample service.
 ecs-demo-php-simple-app

Branch
The branch of the repo to continuously deploy.
 master

User
Your username on GitHub.
 edureka

Personal Access Token
Token for the user specified above. (<https://github.com/settings/tokens>)

Stack Configuration

TemplateBucket
The S3 bucket from which to fetch the templates used by this stack.
 edureka-devops

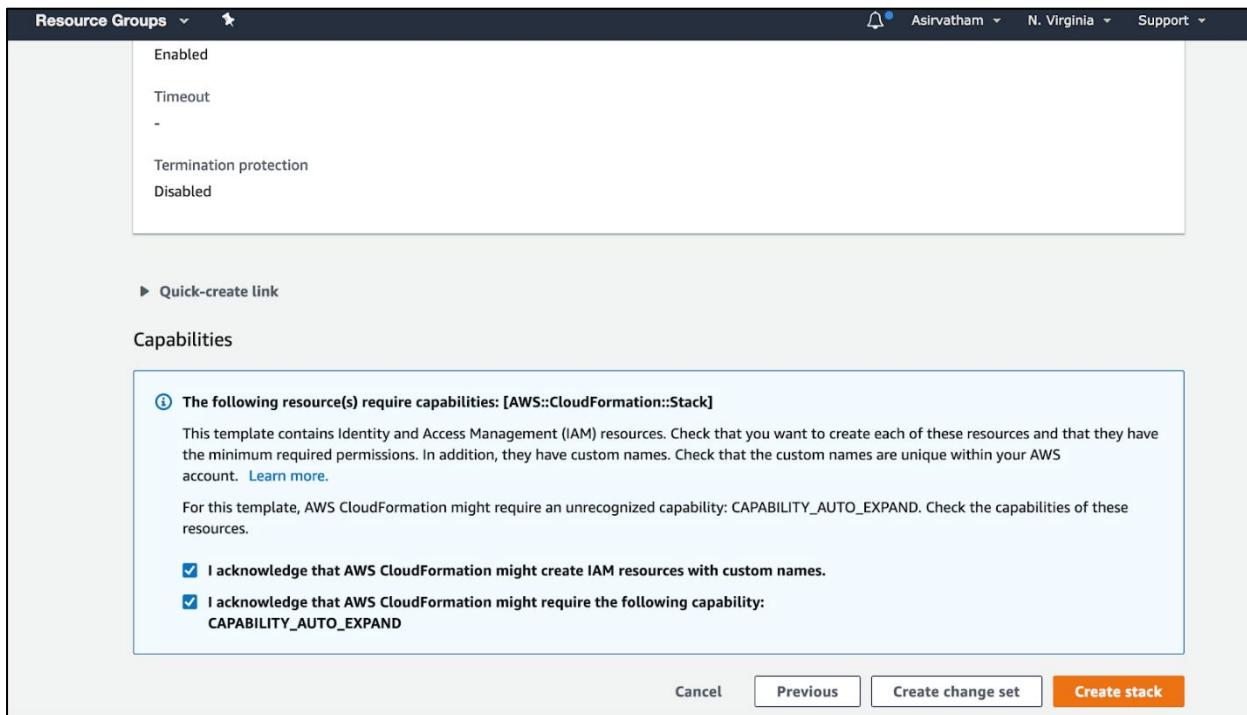
For the User (Your username for GitHub) type the value: edureka

For Personal Access Token use the value: devops

Change the TemplateBucket and point to your bucket which is containing the codes

(Note: The Above 2 values are null values and they can be any characters)

- c) Then Click next and then next till you get the below page, click on the check box to acknowledge and click create stack.



- d) It will take some 10 to 15 mins to create a full stack, later you can check the status of stack by clicking on stack info

Stacks (6)			
	Stack name	Status	
<input type="radio"/>	EdurekaNVCloudFormation-DeploymentPipeline-MP18YG1WCW5C	NESTED	✓ CREATE_COMPLETE
<input type="radio"/>	EdurekaNVCloudFormation-Service-66FJXJXKO2XG	NESTED	✓ CREATE_COMPLETE
<input type="radio"/>	EdurekaNVCloudFormation-Cluster-KSZNR6N01UH5	NESTED	✓ CREATE_COMPLETE
<input type="radio"/>	EdurekaNVCloudFormation-LoadBalancer-ZBFILQ87A2SX	NESTED	✓ CREATE_COMPLETE
<input type="radio"/>	EdurekaNVCloudFormation-VPC-XBA79F37CT52	NESTED	✓ CREATE_COMPLETE
<input type="radio"/>	EdurekaNVCloudFormation		✓ CREATE_COMPLETE

- e) Click on output and you will find the url to access the Load Balancer and the CodePipeline

The screenshot shows the AWS CloudFormation console with the 'Stacks' section selected. The stack name is 'EdurekaNVCloudFormation-DeploymentPipeline-MP18YG1WCW5C'. The 'Outputs' tab is active, highlighted with a red box. A table lists one output: 'PipelineUrl' with the value '<https://console.aws.amazon.com/codepipeline/home?region=us-east-1#/view/EdurekaNVCloudFormation-DeploymentPipeline-MP18YG1WCW5C-Pipeline-HPCF8709JZ8E>'. The 'Value' column is also highlighted with a red box.

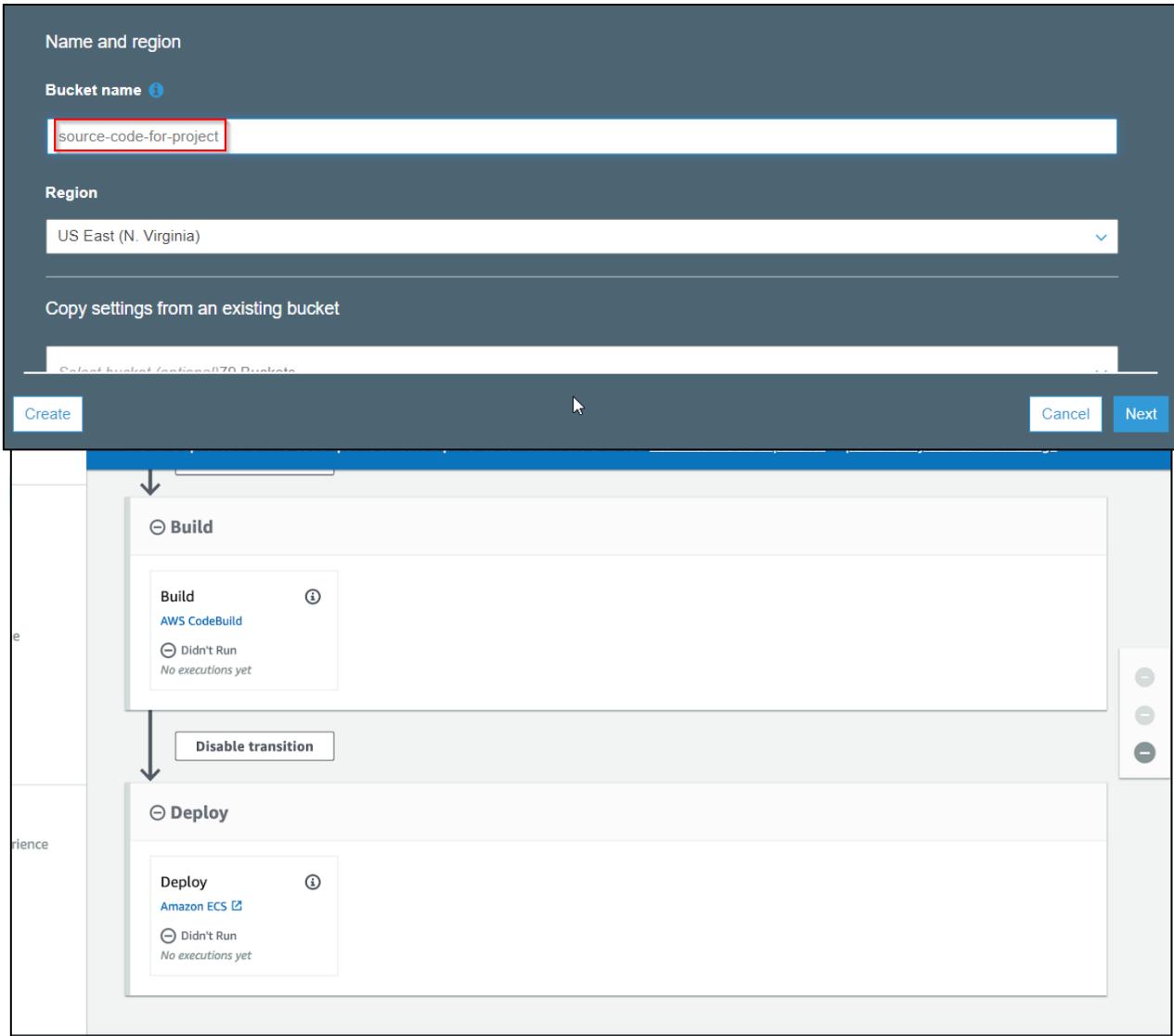
- f) Once you click on the CodePipeline url from the CloudFormation output, you will be able to view the 3-stage code pipeline. Source > Build > Deploy.

The screenshot shows the AWS CodePipeline console. The pipeline name is 'EdurekaNVEnvironment-DeploymentPipeline-MHV27T9H9DU-Pipeline-2A9XHQLAKH77'. The 'Source' stage is currently failed, indicated by a red circle with a minus sign. An arrow points from the Source stage to the 'Build' stage. Below the stages are buttons for 'Edit', 'View history', and 'Release change'. A sidebar on the left shows the pipeline's structure: 'CodePipeline started' -> 'Source' -> 'Build' -> 'Deploy'.

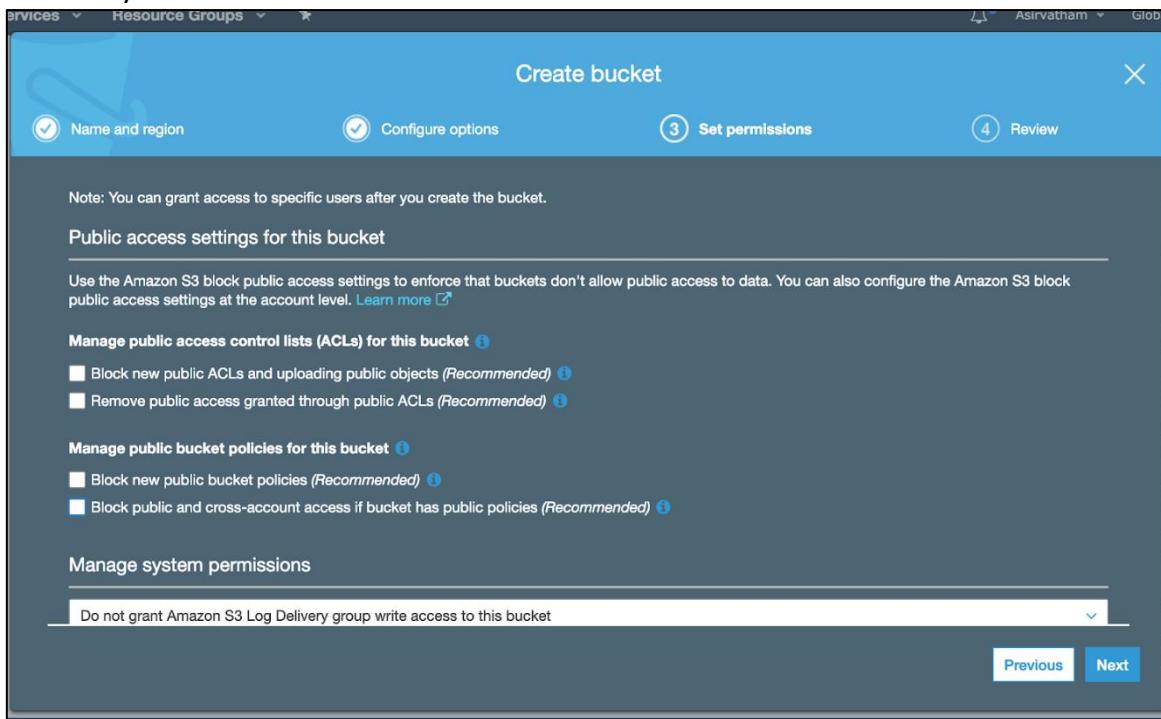
The CodePipeline is not set up yet. And we are going to setup and run the CodePipeline. The source stage is going to show an error since it is set up for GIT using null values. But our code is in S3

Step2: Create a **S3** Bucket in N.Virginia with CRR to Ohio and point Source Stage to S3 buckets in the appropriate region

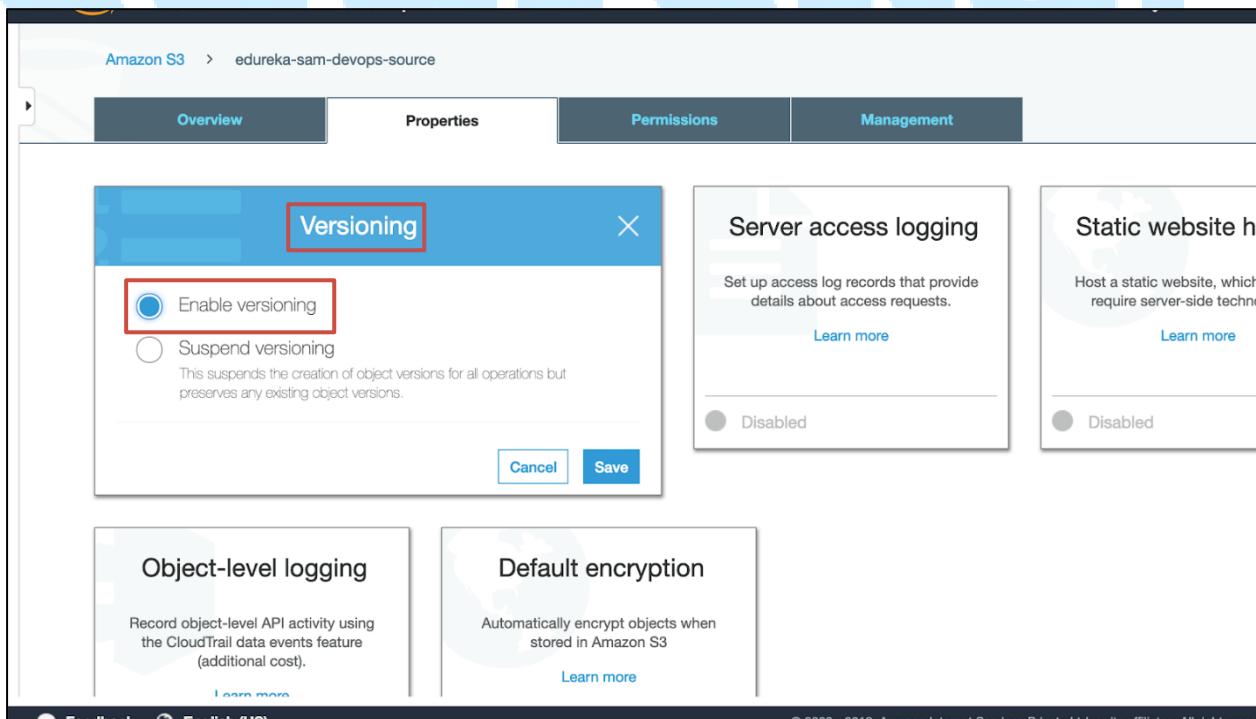
- Create Source S3 bucket in N.Virginia



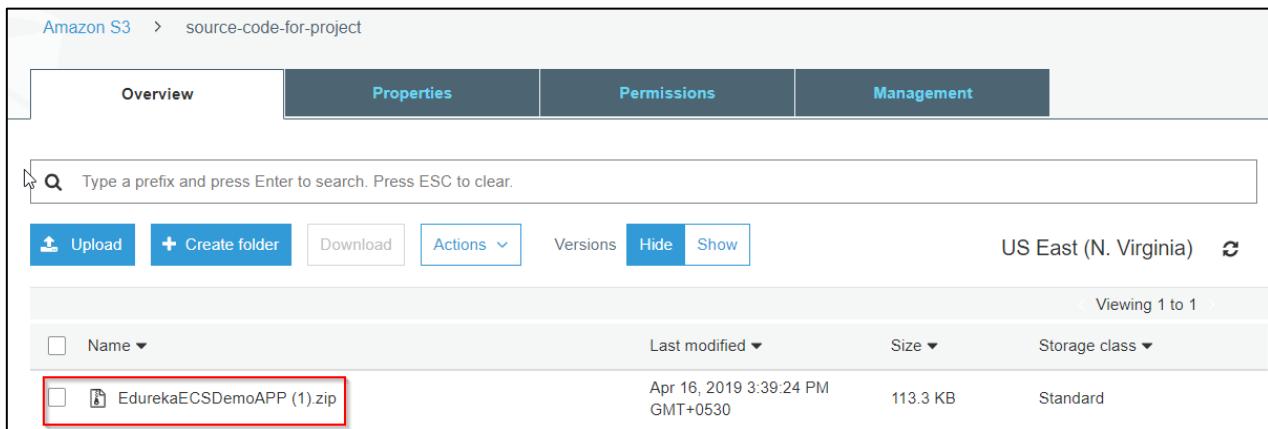
- b) Under Set Permissions uncheck all the options to drag and drop the code into the bucket.
 Finally click on create bucket.



- c) Since we will be storing source code in the bucket enable versioning



- d) Download the source code from ***EdurekaECSDemoAPP.zip*** and upload it to created source bucket

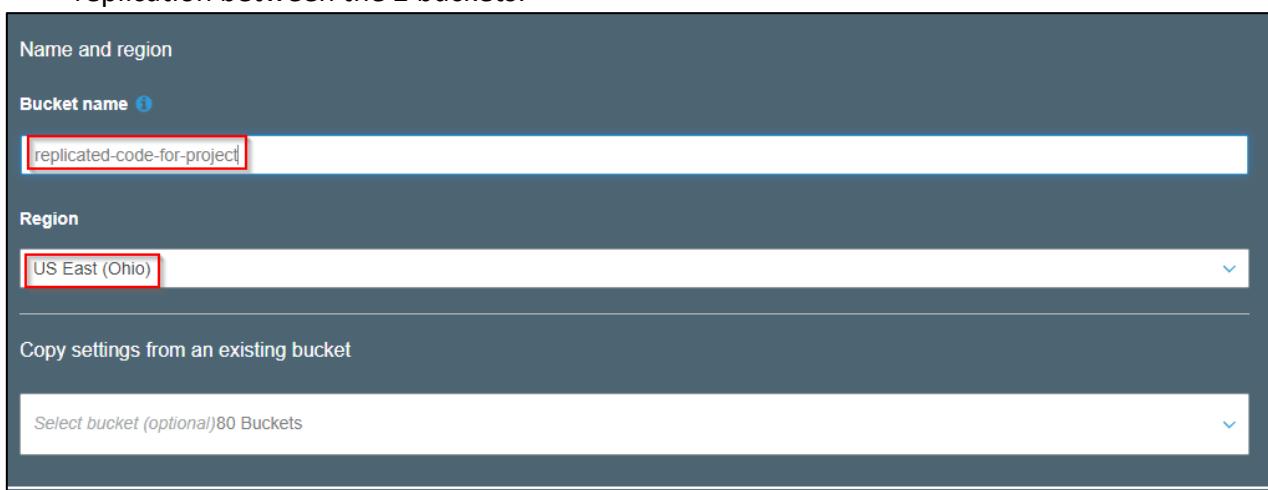


The screenshot shows the Amazon S3 console with a single object listed:

Name	Last modified	Size	Storage class
EdurekaECSDemoAPP (1).zip	Apr 16, 2019 3:39:24 PM GMT+0530	113.3 KB	Standard

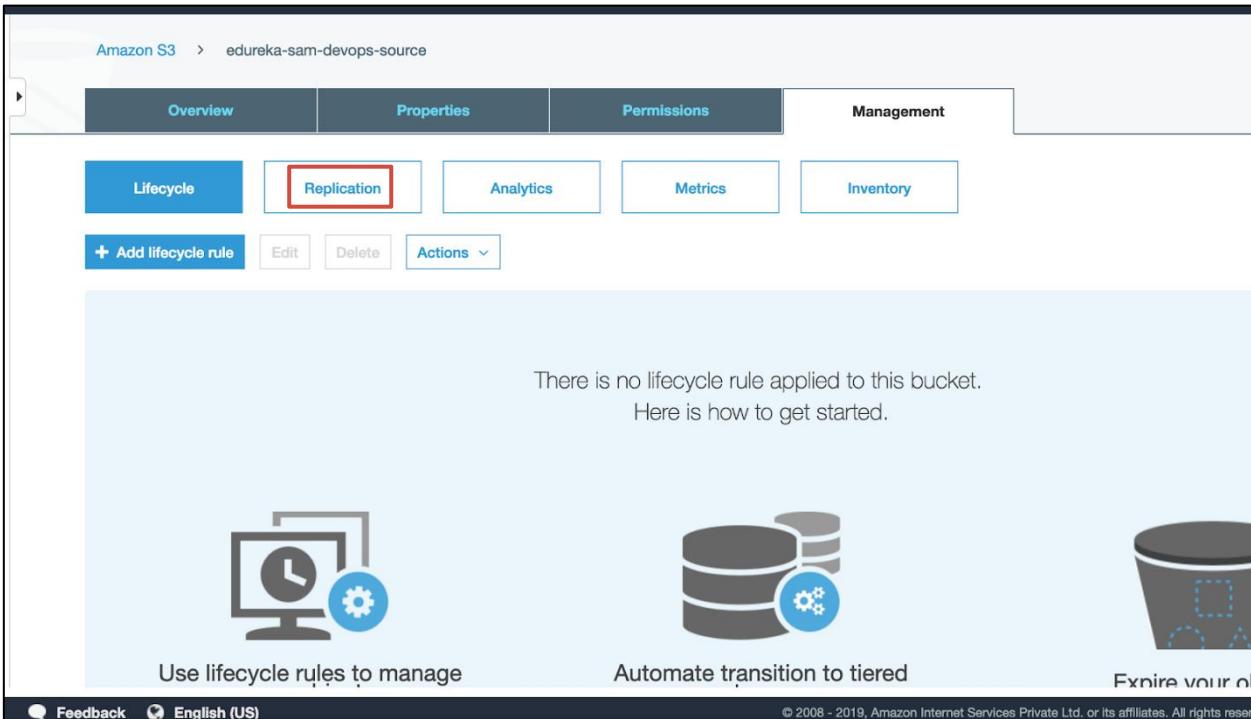
edureka!

- e) For Cross Region Replication environment. Create another S3 bucket in Ohio and set up replication between the 2 buckets.



The screenshot shows the 'Name and region' step of the 'Create New Bucket' wizard. It includes fields for 'Bucket name' (containing 'replicated-code-for-project') and 'Region' (set to 'US East (Ohio)'). There is also a section for 'Copy settings from an existing bucket' with a dropdown menu.

f) Set up replication between the 2 buckets.



Amazon S3 > edureka-sam-devops-source

Overview Properties Permissions Management

Lifecycle Replication Analytics Metrics Inventory

+ Add lifecycle rule Edit Delete Actions

There is no lifecycle rule applied to this bucket.
Here is how to get started.

Use lifecycle rules to manage

Automate transition to tiered

Expires your ob...

Feedback English (US)

© 2008 - 2019, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

edureka!

Set source

Entire bucket  source-code-for-project

Prefix or tags 

Replication criteria

Replicate objects encrypted with AWS KMS 

i Your CRR rule will be created using the new schema

Cross-region replication (CRR) now has a new schema that supports replication based on prefixes, one or more object tags or a combination of the two. As part of the new schema, you can set overlapping rules with priorities. The new schema is available for all new CRR rules. Existing CRR rules will continue to work as they do today.

Destination bucket

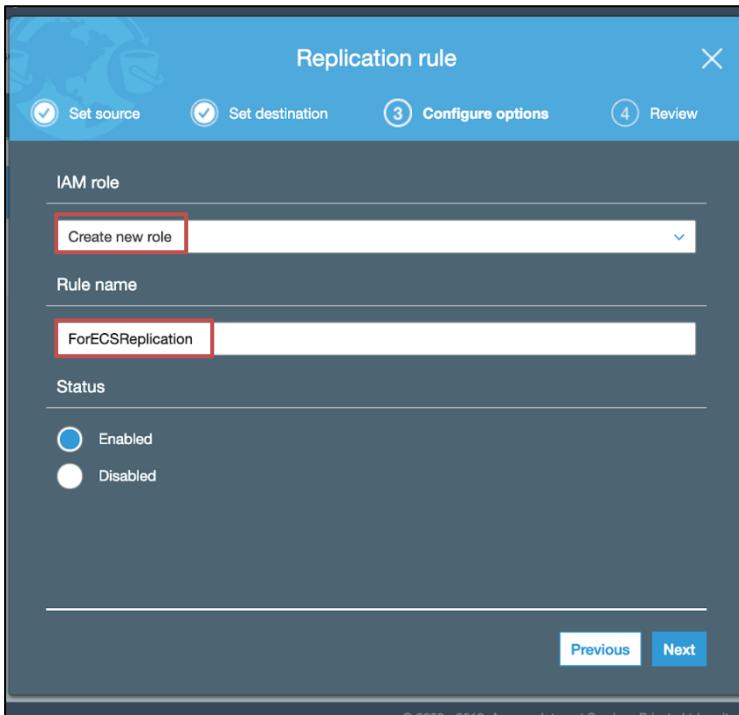
replicated-code-for-project 

Options

Change the storage class for the replicated object(s)

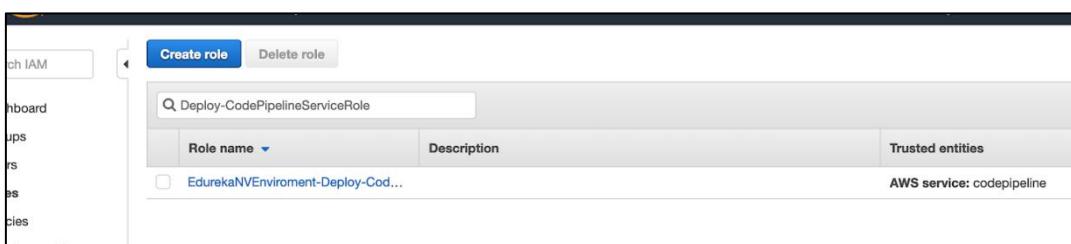
Change object ownership to destination bucket owner 

AWS DevOps Project Solution



The screenshot shows the 'source-code-for-project' bucket's replication configuration. The 'Permissions' tab is selected. Under 'Replication', a success message 'Cross-region replication updated successfully.' is displayed. The 'Source' and 'Destination' sections show the setup: Source is 'Bucket source-code-for-project' from 'Region US East (N. Virginia)'; Destination is 'Bucket replicated-code-for-project' in 'Region US East (Ohio)'. The 'Permissions' section lists 'IAM role s3crrole_for_source-code-for-project_to_replicated-code-for-pr' and 'Bucket policy Copy'. An 'Edit global settings' link is also present.

Step 3: Now we can point to the CodePipeline to pull codes from S3. But the CodePipeline needs to be given enough access to pull codes from S3. Go to **IAM** provide S3 Full Access.



Role ARN: arn:aws:iam::245376966395:role/EdurekaNVCloudFormation-De-CodePipelineServiceRole-8B345TYD579C

Role description: Edit

Instance Profile ARNs: [Edit](#)

Path: /

Creation time: 2019-04-16 15:14 UTC+0530

Maximum CLI/API session duration: 1 hour [Edit](#)

Permissions [Trust relationships](#) [Tags](#) [Access Advisor](#) [Revoke sessions](#)

▼ Permissions policies (3 policies applied)

[Attach policies](#) [Add inline policy](#)

Policy name	Policy type	X
▶ AmazonS3FullAccess	AWS managed policy	X
▶ AmazonSNSFullAccess	AWS managed policy	X

Show 1 more

Step 4: Go to **CodePipeline** and edit the source stage

- The Pipeline would have failed to move to the next stage since it is pointing to GIT instead of S3, so remove GIT and add S3. Now click on Edit

Developer Tools X

Introducing the new AWS CodePipeline console experience!

We have updated the AWS CodePipeline console experience to make it easier to use. [Return to the old experience](#) or [permanently dismiss this message](#).

Source • CodeCommit

Build • CodeBuild

Deploy • CodeDeploy

Pipeline • CodePipeline

Getting started

Pipelines

Pipeline

History

Feedback

Return to the old experience

EdurekaNVEnvironment-DeploymentPipeline-MHV27T9H9DU-Pipeline-2A9XHQLAKH77

[Edit](#) [View history](#) [Release change](#)

Source

App GitHub ⓘ

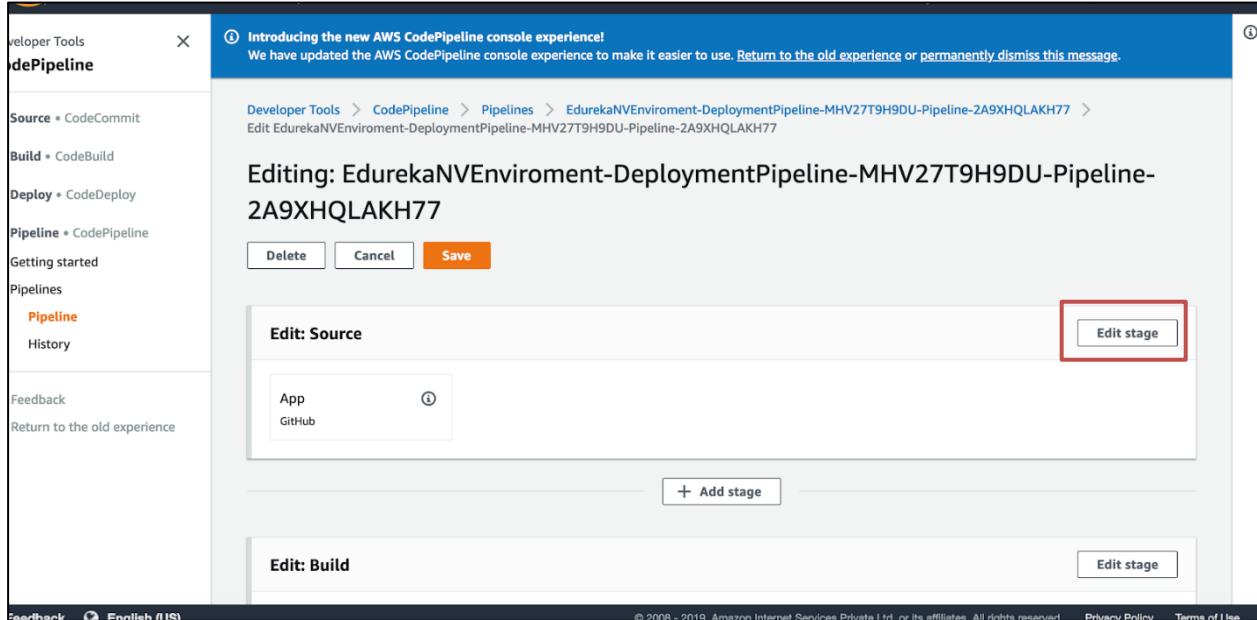
Failed - 56 minutes ago [Details](#)

Disable transition

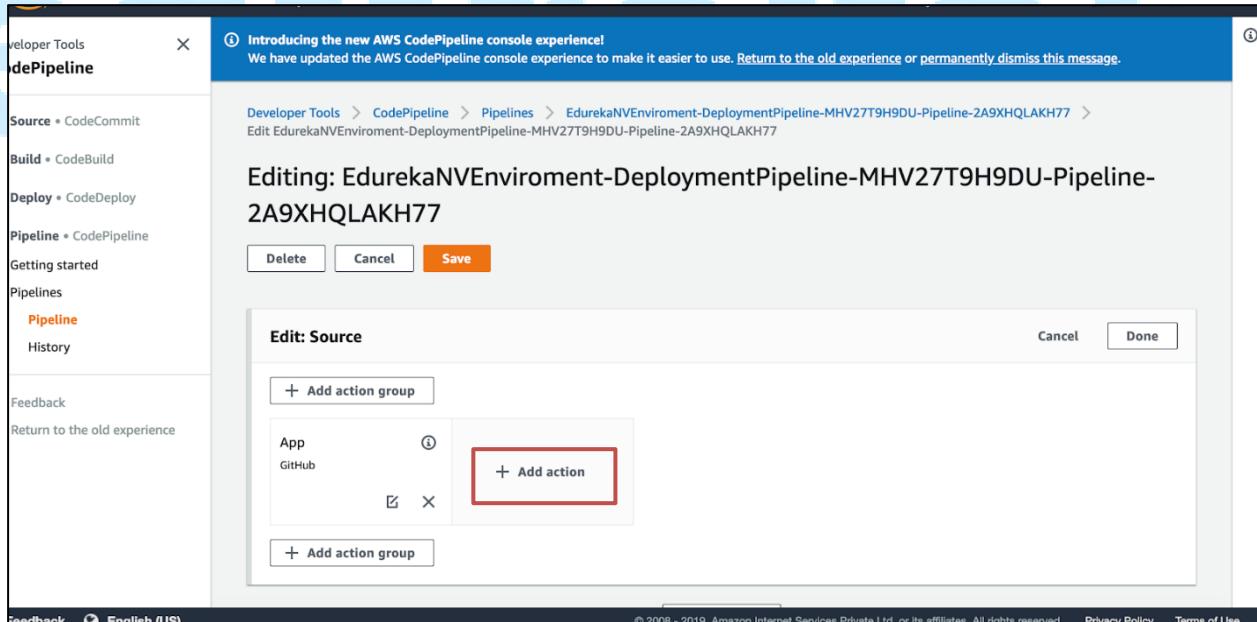
Build

Build ⓘ

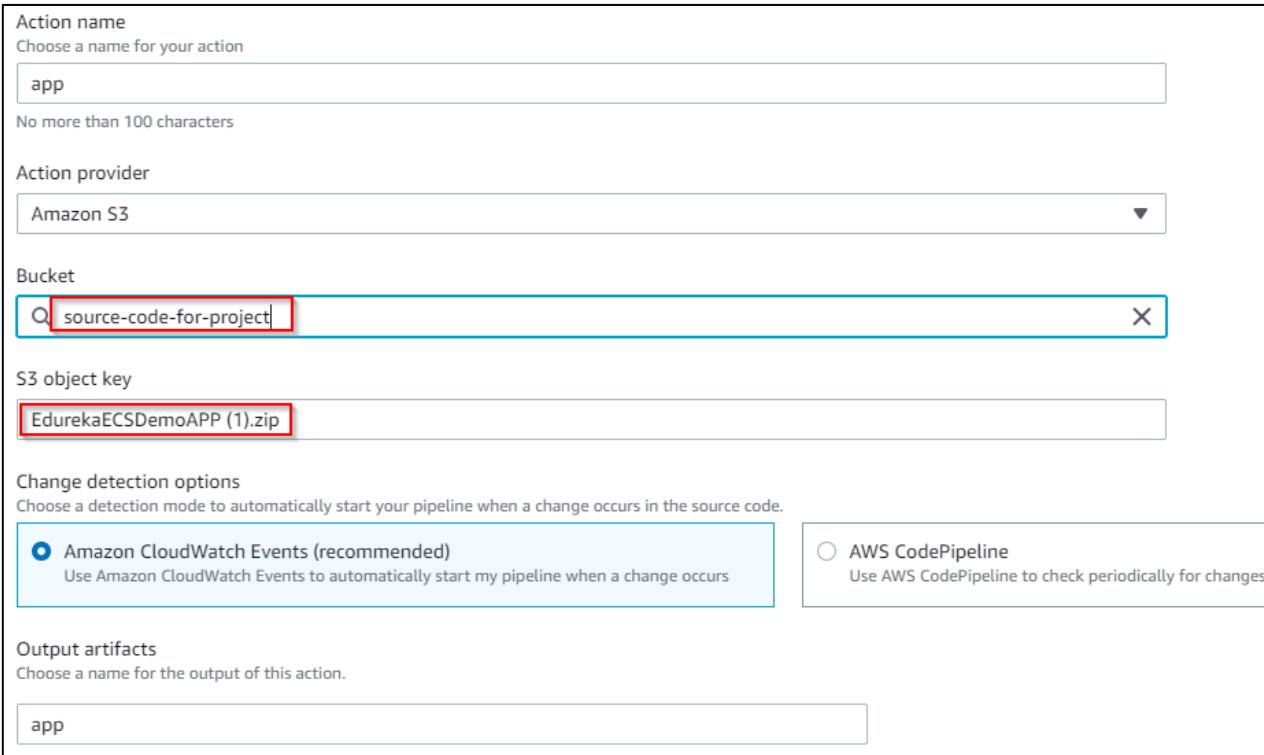
b) Click on edit stage of Source



c) Remove App GitHub and click on Add Actions



d) Fill up the details as mentioned below and click on save:



Action name
Choose a name for your action

No more than 100 characters

Action provider

Bucket

S3 object key

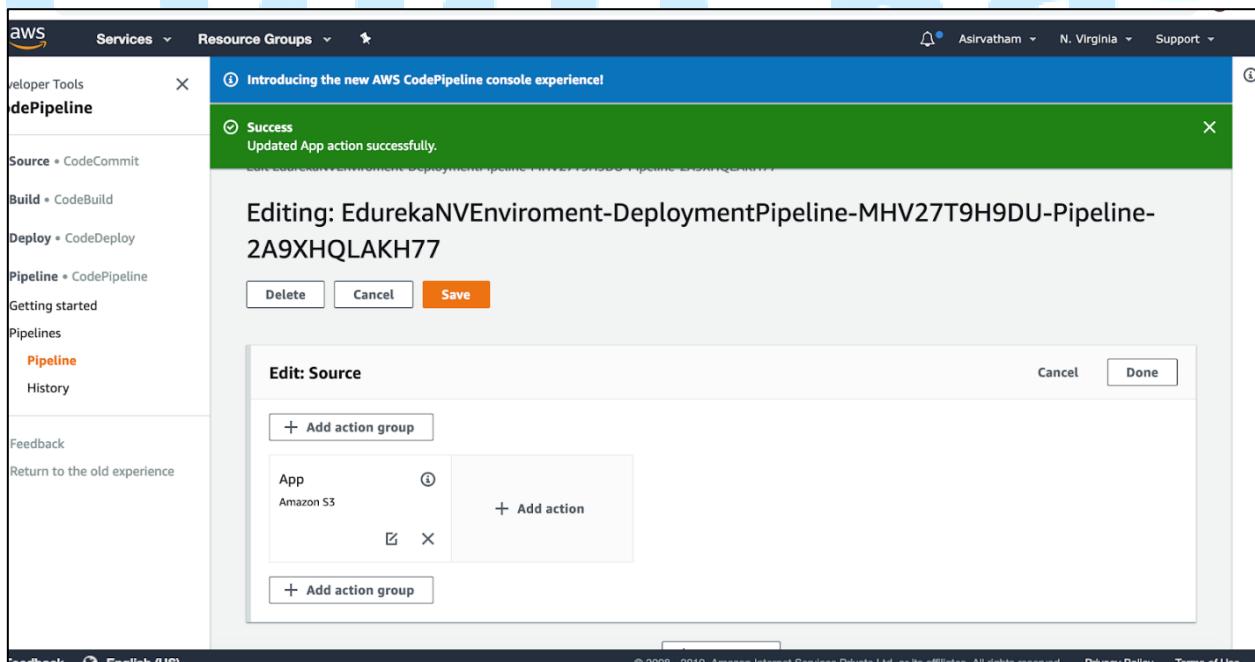
Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
 Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs

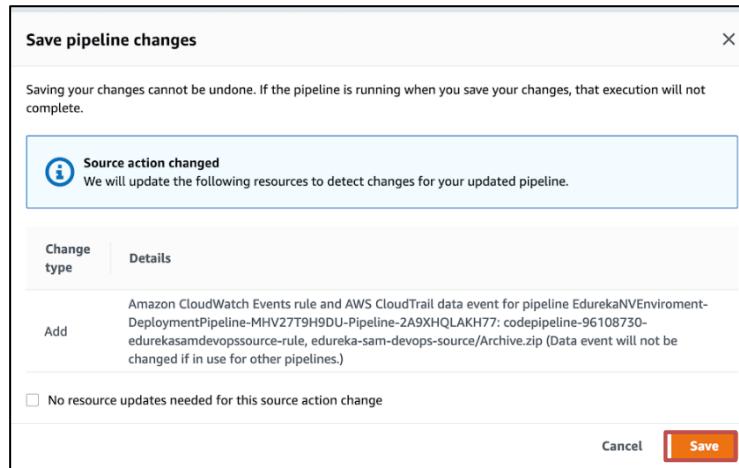
AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Output artifacts
Choose a name for the output of this action.

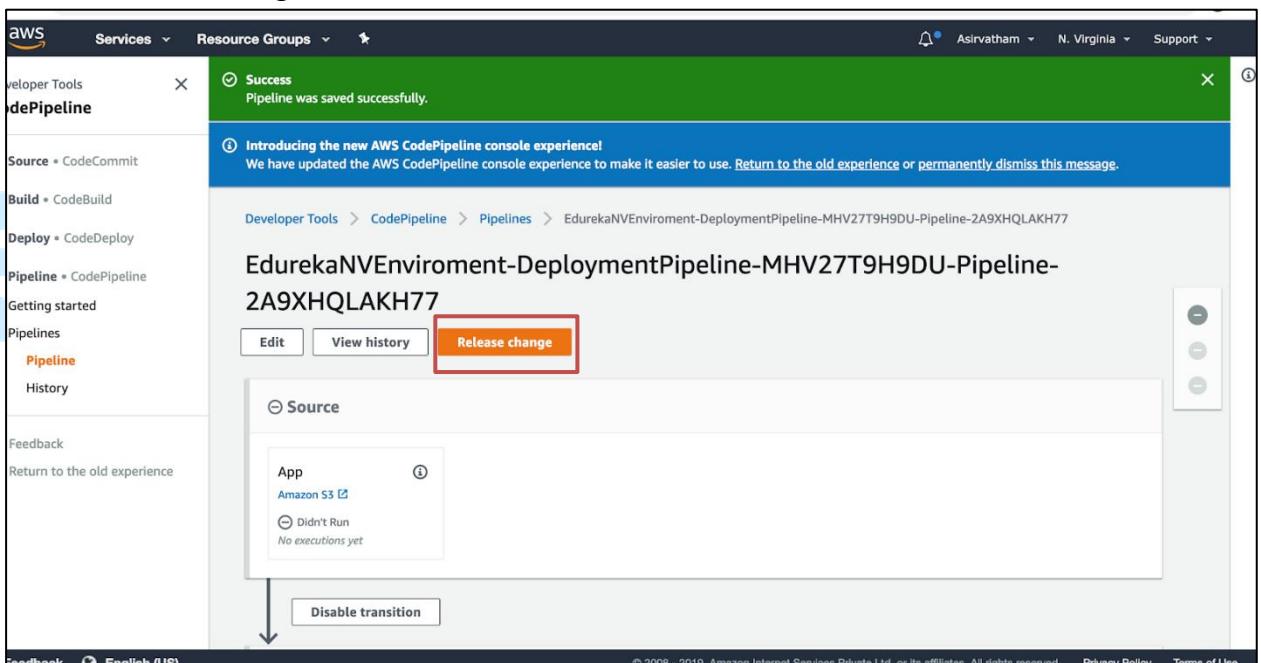
No more than 100 characters

e) On successfully adding the action click on save again



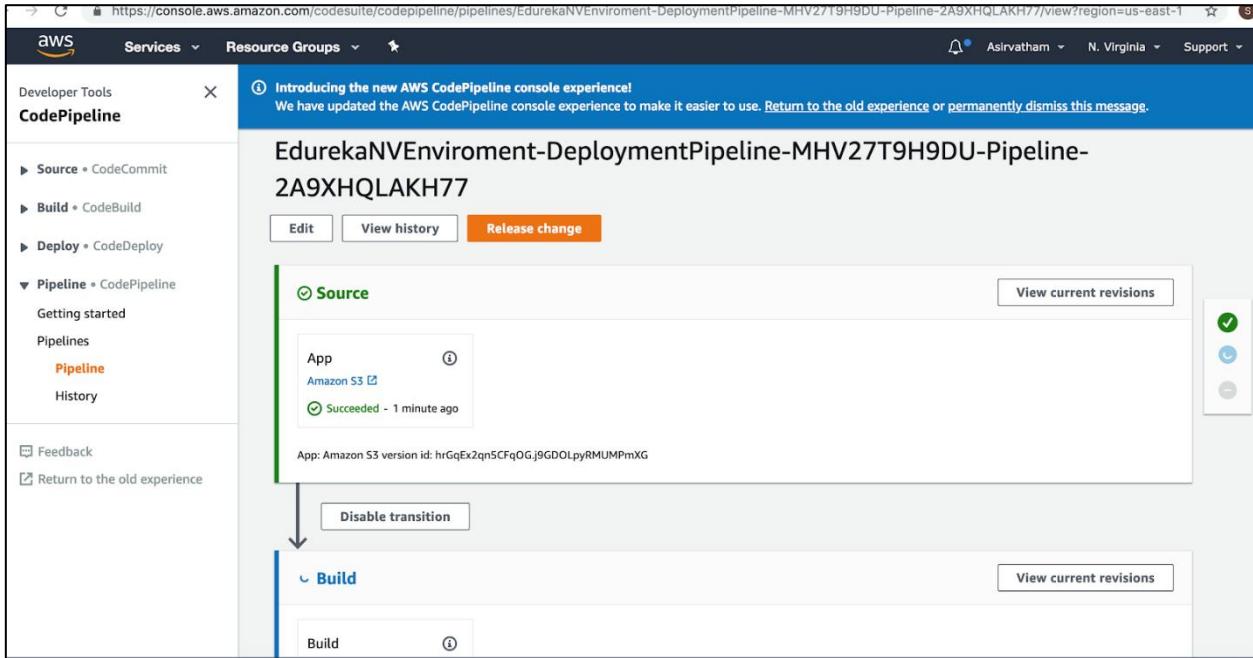


f) Click on release changes

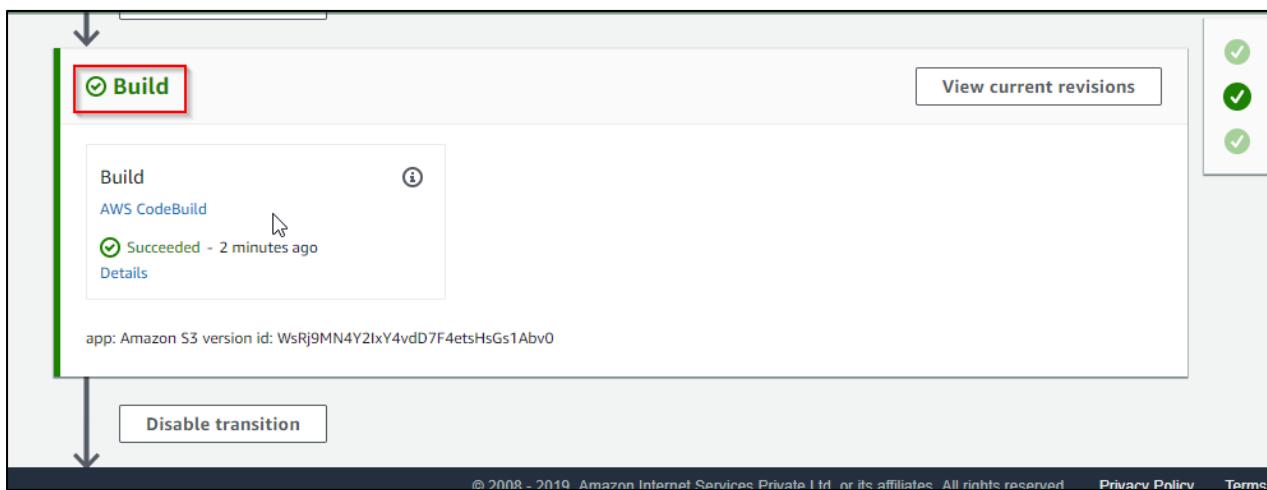


g) Now source stage starts executing, then build and finally deploy as shown below:

Source:-



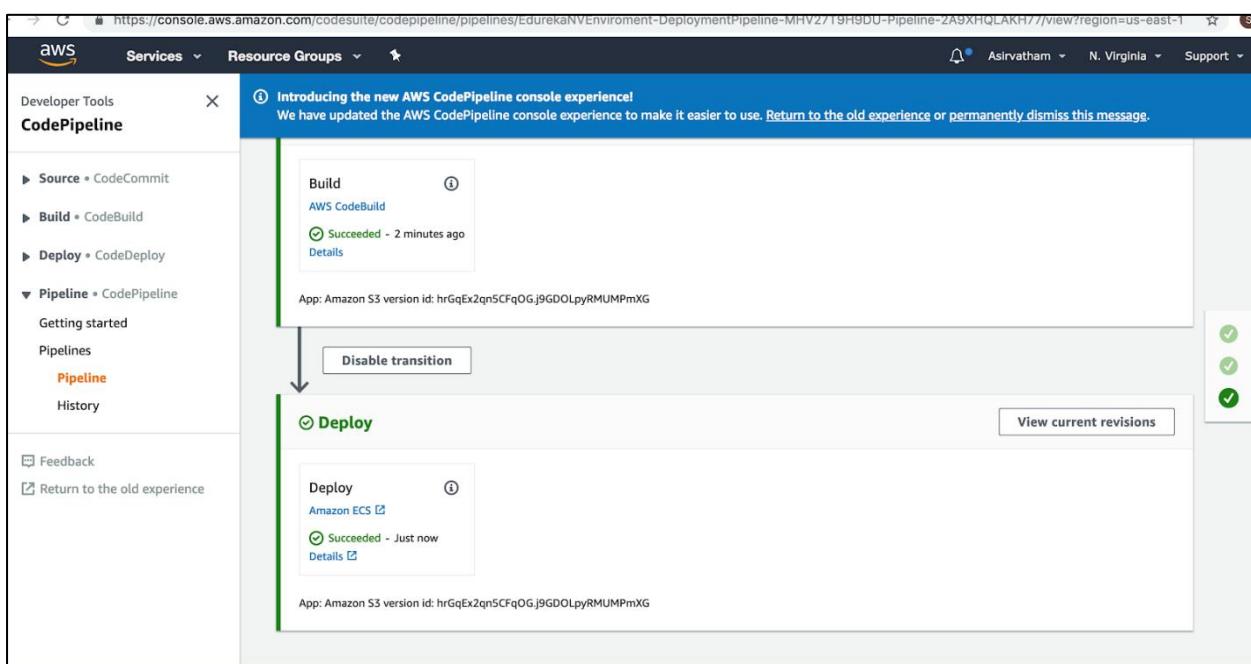
Build:-



Deploy:-



h) Then the CodePipeline would have fully run with all Green



- i) By this time the Custom code which we have in S3 is pushed to the ECS and served to the users through ELB as shown below.

The screenshot shows the AWS Elastic Load Balancing (ELB) console. On the left, there's a sidebar with various AWS services like Launch Templates, Spot Requests, Reserved Instances, etc. The main area shows a table of existing load balancers. One row is selected, and its details are displayed below. The 'Basic Configuration' section includes fields for Name, ARN, DNS name, State, Type, Scheme, and IP address type. The 'DNS name' field contains the value 'Edure-LoadB-RW6APJDMFGQA-107626265.us-east-1.elb.amazonaws.com (A Record)', which is highlighted with a red box.

The screenshot shows a web browser window with the URL 'edure-loadb-ybunwxqsfibx-922023863.us-east-1.elb.amazonaws.com'. The page content is a deployment confirmation message:

**Version 2 : Edureka DevOps Project
Using AWS Code* Services,
CloudFormation and Containers**

**Congratulations! Your App is now Deployed and is running on a
container**

Versioning! From S3.
The container is running PHP version 5.3.10-1ubuntu3.26.
Tue Apr 16 11:50:41 UTC 2019

Step 5: Now Lets EDIT the code and see if CodeBuild picks the changed code from S3 and deploys to the containers running in **ECS**

- To check the working of CodeBuild, unzip the source code, go to **SCR directory** and using any of the editor make changes in Index.php file

```

<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <title>Simple PHP App</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="assets/css/bootstrap.min.css" rel="stylesheet">
    <style>body {margin-top: 40px; background-color: #333;}</style>
    <link href="assets/css/bootstrap-responsive.min.css" rel="stylesheet">
    <!--[if lt IE 9]><script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script><![endif]-->
</head>

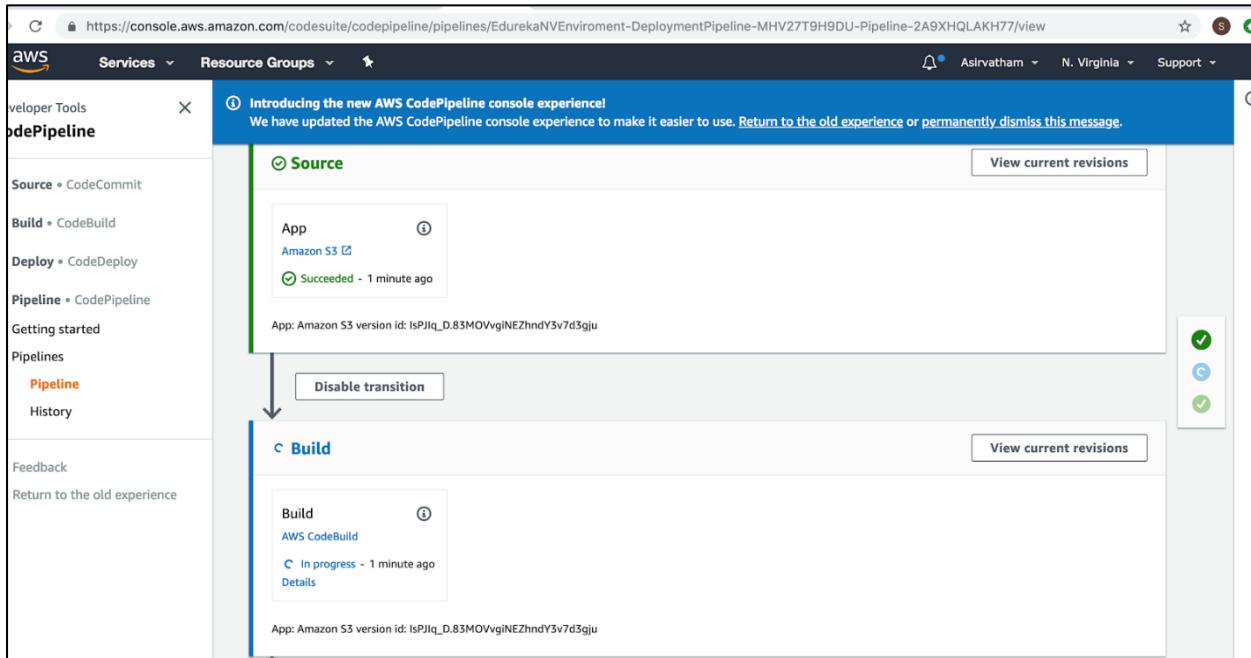
<body>
    <div class="container">
        <div class="hero-unit">
            <h1> Edureka DevOps Project </h1>
            <h2>Congratulations! Your App is now Deployed and is running on a container</h2>
            <p>Versioning! From S3.</p>
            <p>The container is running PHP version <?php echo phpversion(); ?>.</p>
            <?php
                $myfile = fopen("/var/www/my-vol/date", "r") or die("");
                echo fread($myfile,filesize("/var/www/my-vol/date"));
                fclose($myfile);
            ?>
        </div>
    </div>
    <script src="//ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
    <script src="assets/js/bootstrap.min.js"></script>
</body>
</html>

```

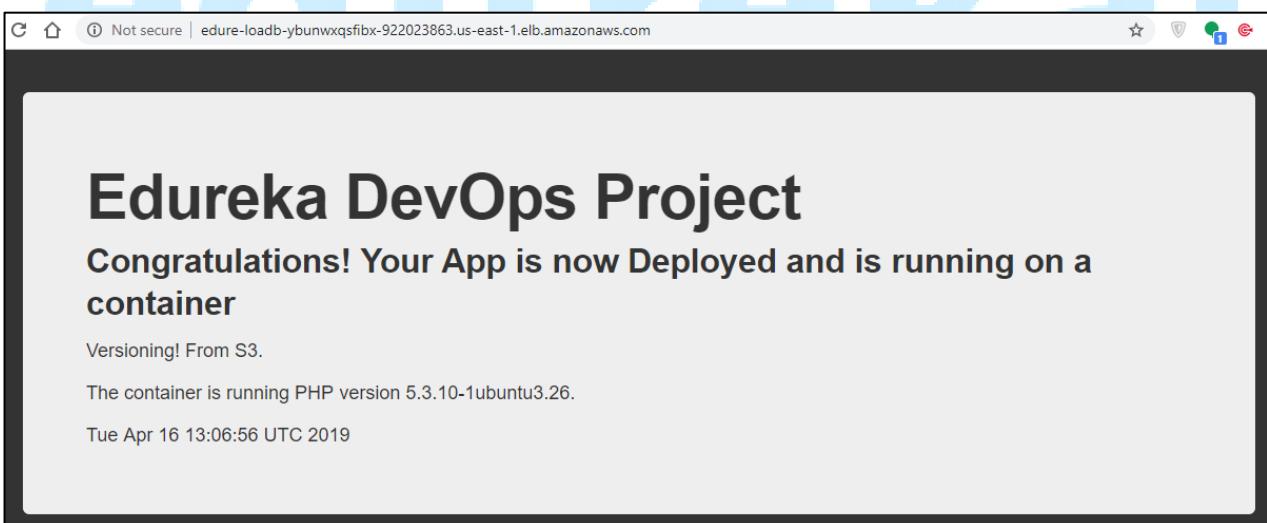
- Once you are done (Delete the old zipped file), save the changes and zip it with same name to be uploaded to S3.

Name	Date modified	Type	Size
src	16-04-2019 16:31	File folder	
.DS_Store	10-04-2019 23:57	DS_STORE File	9 KB
Dockerfile	18-03-2019 22:12	File	1 KB
EdurekaECSDemoAPP (1)	16-04-2019 17:37	WinRAR ZIP archive	38 KB
LICENSE	18-03-2019 22:12	File	12 KB
NOTICE.md	18-03-2019 22:12	MD File	2 KB
README.md	18-03-2019 22:12	MD File	1 KB
simple-app-task-def.json	18-03-2019 22:12	JSON File	2 KB

- c) Once you have uploaded, you will see, CodePipeline noticed the change in the source code and started a new release.

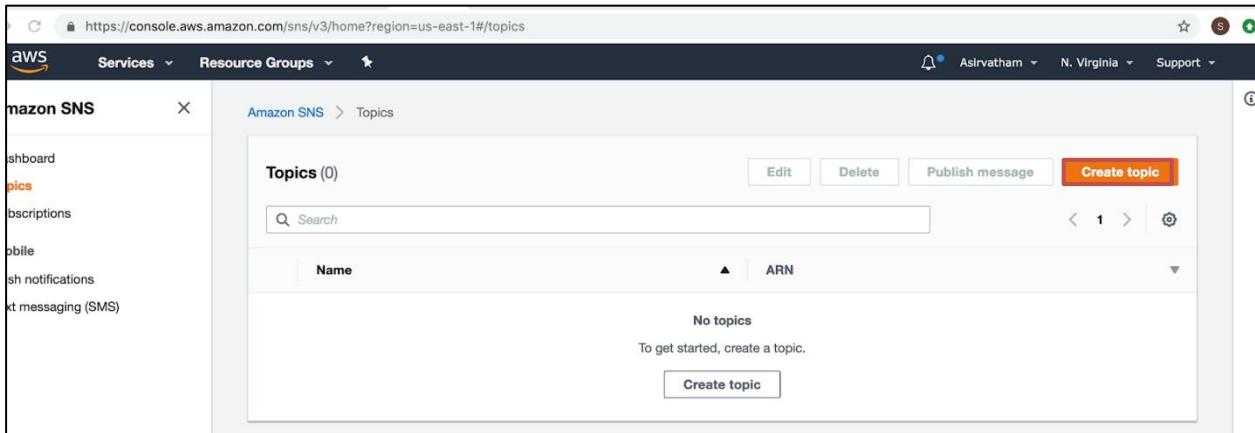


- d) Check the deployment and confirm if the changes are reflected.

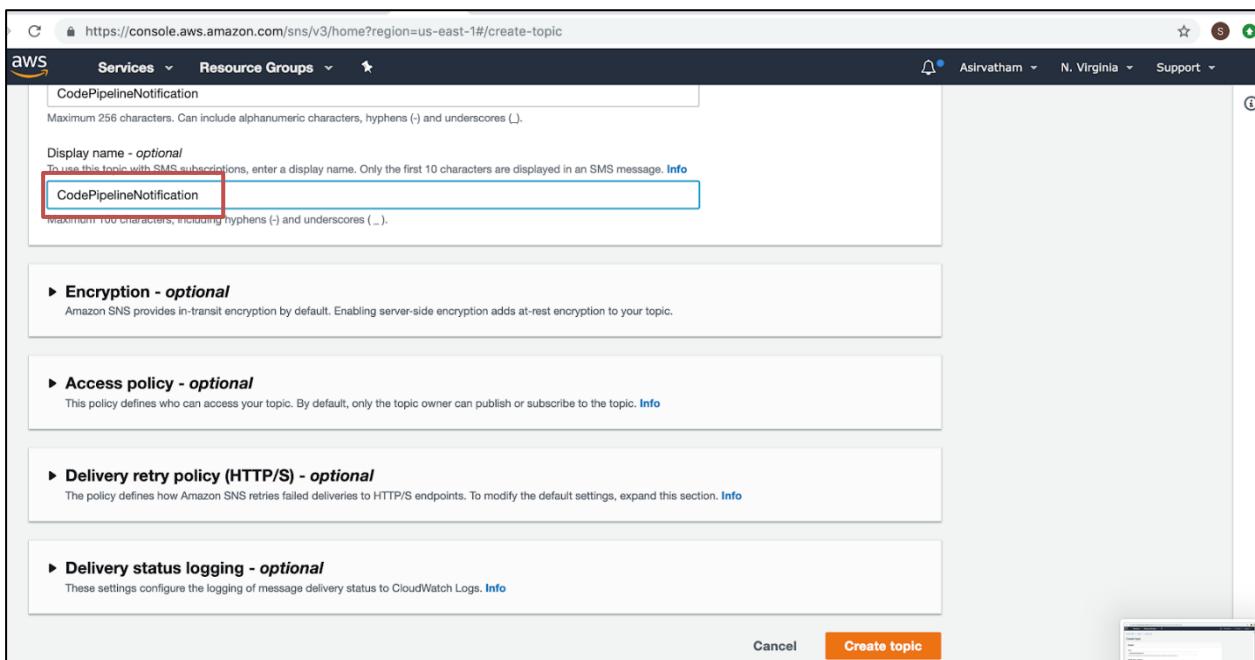


Step 6: Let's now set up **CloudWatch** to monitor the Pipeline whenever a new deployment is triggered and emails the application owner through **SNS**

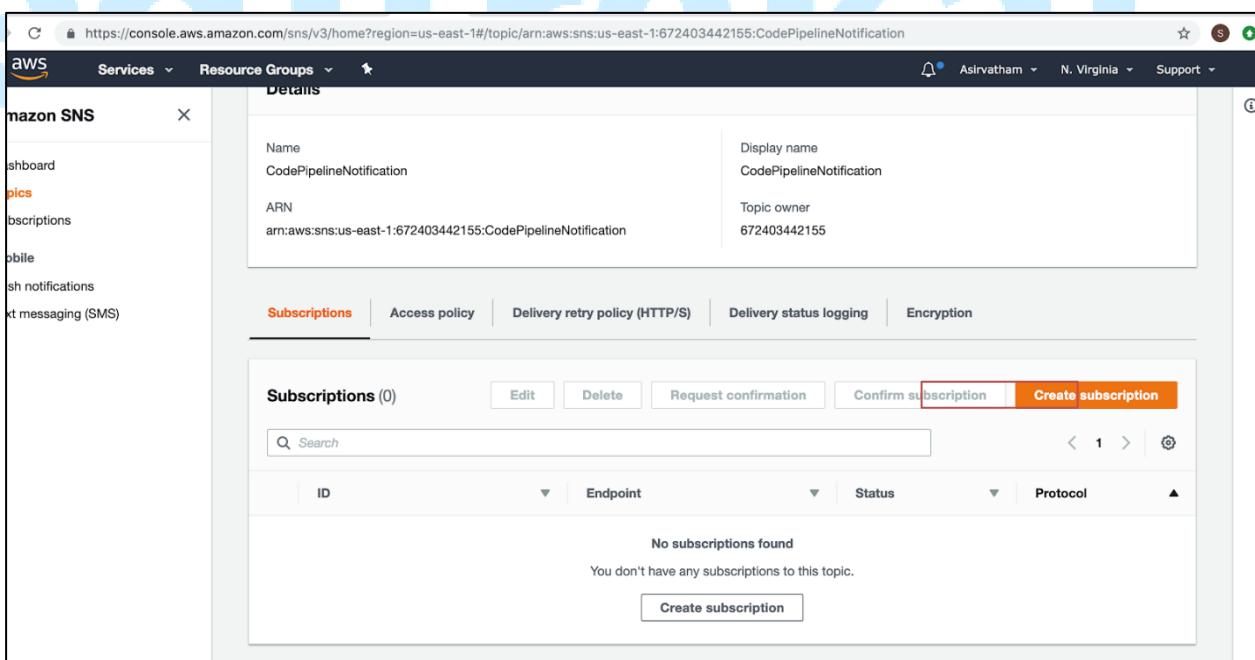
a) Navigate to SNS and create a topic

A screenshot of the 'Create topic' form. The 'Details' section has a 'Name' field containing 'CodePipelineNotification' and a 'Display name - optional' field also containing 'CodePipelineNotification'. Below these are sections for 'Encryption - optional' and 'Access policy - optional'. The entire 'Name' field is highlighted with a red box.

AWS DevOps Project Solution

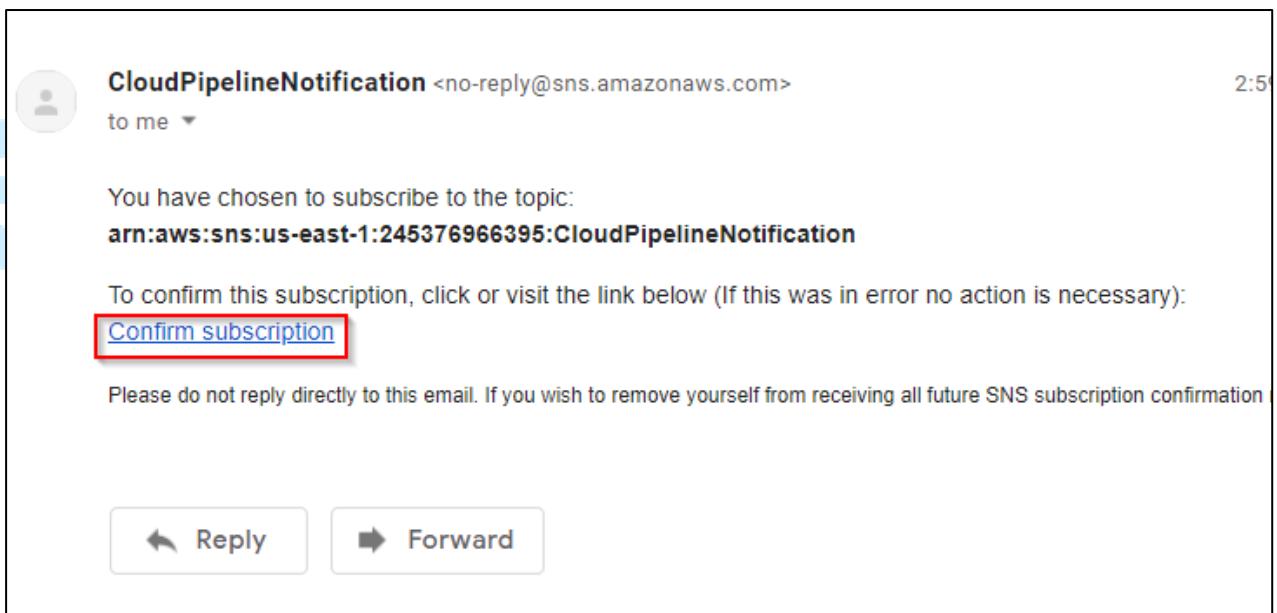


b) Create subscription and confirm it via the received mail



AWS DevOps Project Solution

The screenshot shows the 'Create subscription' page in the AWS SNS console. The 'Topic ARN' field contains 'arn:aws:sns:us-east-1:672403442155:Codef'. The 'Protocol' dropdown is set to 'Email', and the 'Endpoint' field contains 'sam_miee2000@yahoo.com'. A note below states: 'After your subscription is created, you must confirm it.' A 'Subscription filter policy - optional' section is present. At the bottom are 'Cancel' and 'Create subscription' buttons.



AWS DevOps Project Solution

The screenshot shows the AWS SNS console for the 'CodePipelineNotification' topic. The 'Subscriptions' tab is active. A single subscription is listed:

ID	Endpoint	Status	Protocol
8828e963-f2c5-47b5-94b6-a4829fb3f478	sam_miee2000@yahoo.com	Confirmed	EMAIL

Step 7: Now let's create **CloudWatch alarm** that gets triggered whenever there is an execution that has started, passed or failed.

- a) Go to CloudWatch console and select rule.

The screenshot shows the AWS CloudWatch Rules page. The 'Rules' section is highlighted with a red box. A table lists five CloudWatch Events rules:

Status	Name	Description
OK	codepipeline-16266824-edurekadevopsbucket-rule	Amazon CloudWatch Events rule to automatically start your pipeline when a change o
OK	codepipeline-96108730-edurekasamdevopssource-rule	Amazon CloudWatch Events rule to automatically start your pipeline when a change o
OK	codepipeline-edurek-latest-175121-rule	Amazon CloudWatch Events rule to automatically start your pipeline when a change o
OK	codepipeline-edurek-latest-186058-rule	Amazon CloudWatch Events rule to automatically start your pipeline when a change o
OK	codepipeline-edurek-master-183565-rule	Amazon CloudWatch Events rule to automatically start your pipeline when a change o

b) Configure following details:

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern Schedule

Build event pattern to match events by service

Service Name: CodePipeline
Event Type: CodePipeline Pipeline Execution State Change

Any state Specific state(s)

STARTED, FAILED, SUCCEEDED

Targets

Select Target to invoke when an event matches your Event Pattern or when scheduled.

SNS topic

Topic*: CodePipelineNotification

Add target*

Step 2: Configure rule details

Rule definition

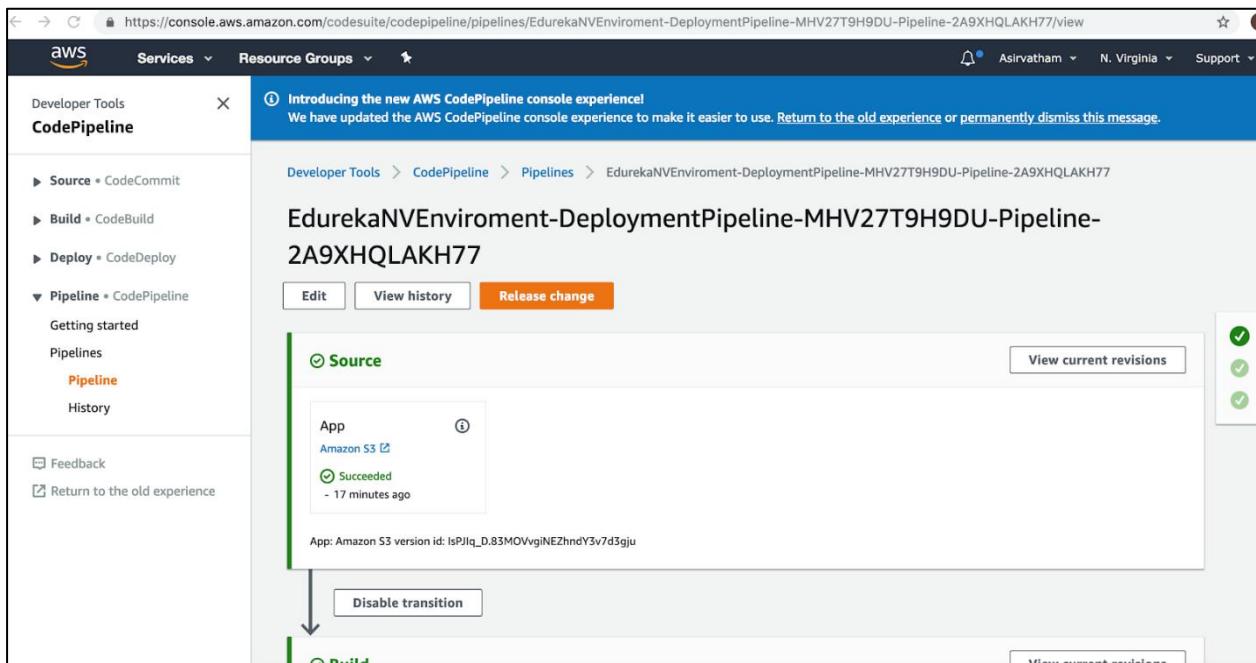
Name*: EdurekaPipeLineStartS
Description: EdurekaPipelineStartStop
State: Enabled

CloudWatch Events will add necessary permissions for target(s) so they can be invoked when this rule is triggered.

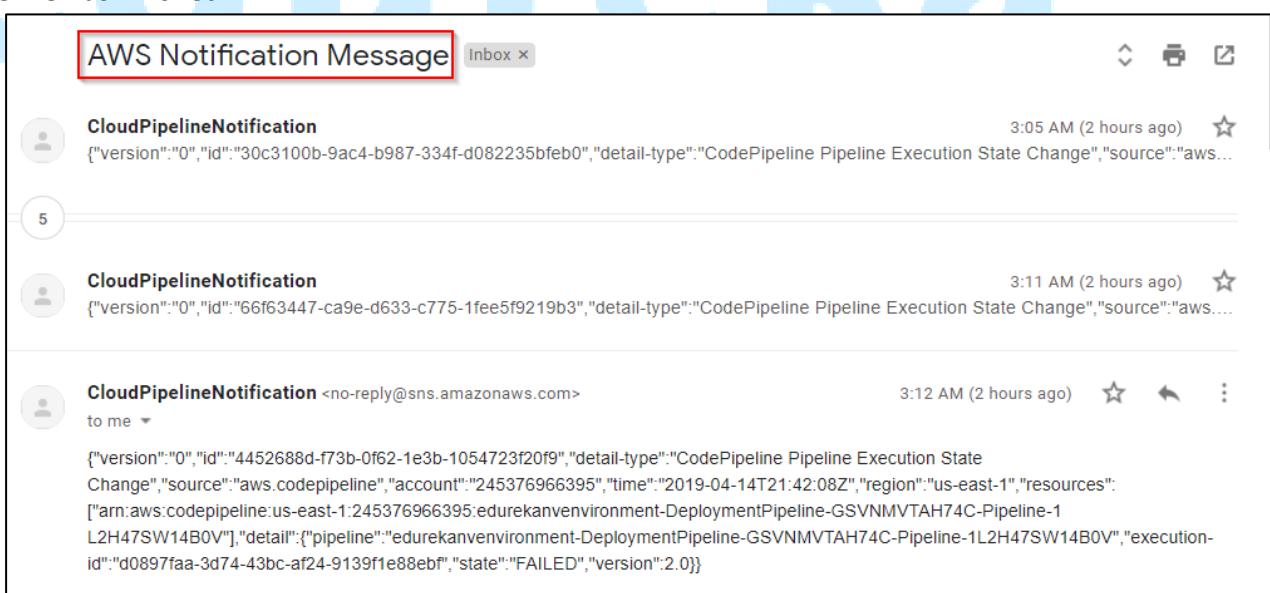
* Required

Cancel Back Create

- c) Let's now test the Pipeline SNS notification by initiating a new build by releasing a change.



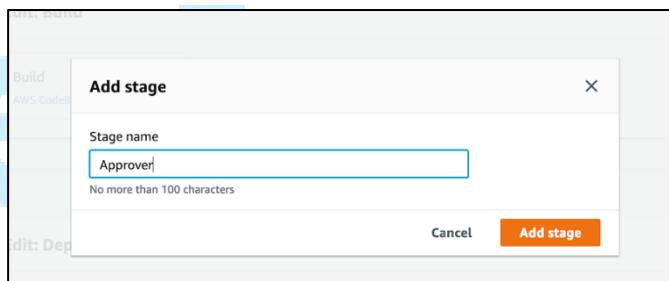
- d) You should see flow of emails, stating when the pipeline has started and later when the pipeline has finished



Step 8: Add *Approver* to the Pipeline

- a) Edit the pipeline and add a new stage after the build stage and before the deploy stage.

The screenshot shows the AWS CodePipeline console interface. On the left, there's a sidebar with options like Source, Build, Deploy, and Pipeline. The main area displays a pipeline named "EdurekaNVEnvironment-DeploymentPipeline". It consists of three stages: "Edit: Build" (using AWS CodeBuild) and "Edit: Deploy" (using Amazon ECS). Each stage has an "Edit stage" button. A blue callout box points to the "Edit stage" button for the "Edit: Deploy" stage.



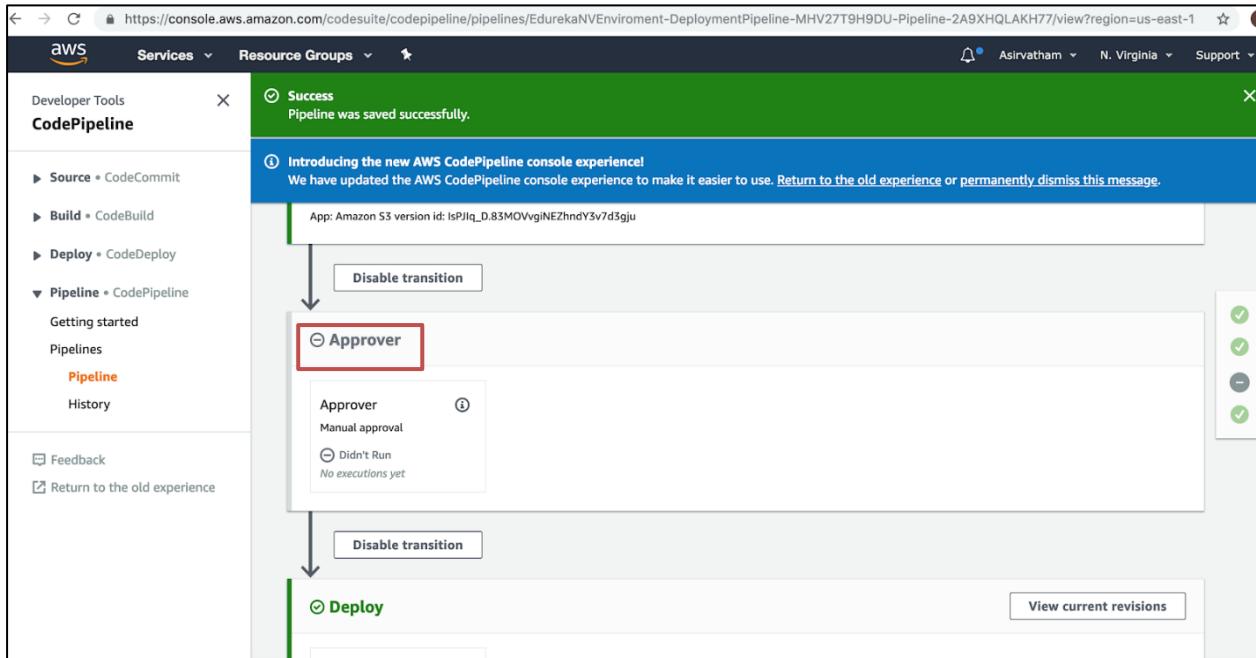
- b) Edit the Approver stage, click on add action and fill up the details as mentioned below:

The screenshot shows the 'Edit action' dialog box from the AWS CodePipeline console. The action name is set to 'Approver'. The action provider is 'Manual approval'. An SNS topic ARN is specified as 'arn:aws:sns:us-east-1:672403442155:CodePipelineNotification'. The URL for review is left empty. Comments are also left empty. The 'Save' button is visible at the bottom right.

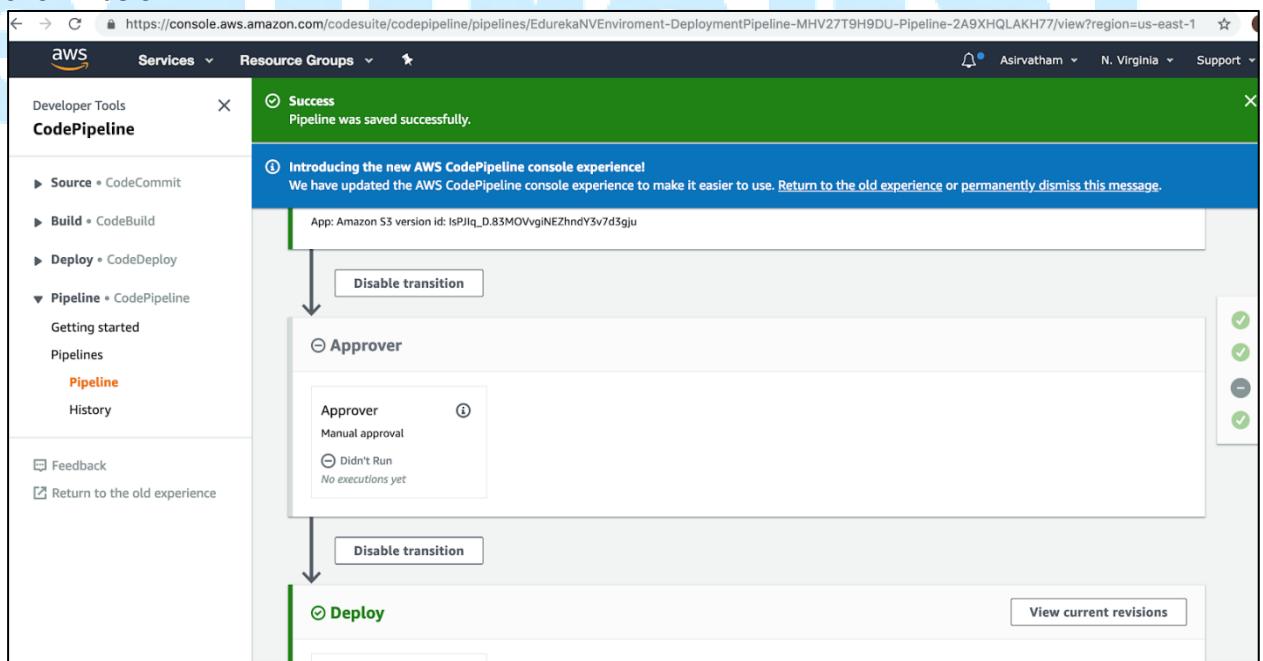
edureka!

AWS DevOps Project Solution

- c) After completing you will now see a new stage added.

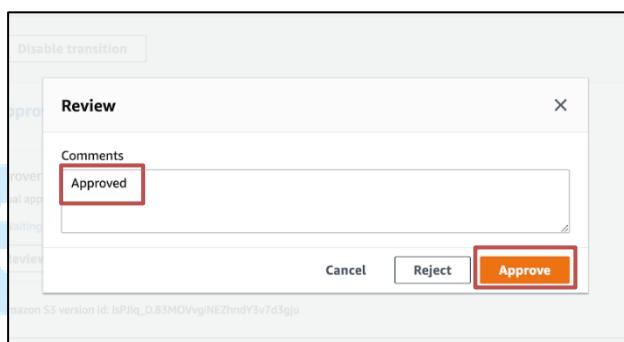


- d) Test the approver stage by triggering the build again. The Build will run and will pause while waiting for the approval. The Build will start to run once the approval is given as shown below.



AWS DevOps Project Solution

The screenshot shows the AWS CodePipeline console interface. On the left, a sidebar navigation includes 'Source' (CodeCommit), 'Build' (CodeBuild), 'Deploy' (CodeDeploy), and 'Pipeline' (CodePipeline). Under Pipeline, 'Getting started', 'Pipelines', and 'History' are listed. A red box highlights the 'Review' button in the 'Approver' section of the pipeline stage. The pipeline consists of three stages: 'Approver', 'Disable transition', and 'Deploy'. The 'Approver' stage is currently active, showing a manual approval step. The 'Deploy' stage is shown below it.



ekal!

The screenshot shows the AWS CodePipeline console interface after the approval step has been completed. The 'Approver' stage now displays a green checkmark and the message 'Approved - Just now'. The pipeline continues through the 'Disable transition' and 'Deploy' stages. The 'Deploy' stage is currently active, showing an 'Amazon ECS' task with the status 'In progress - Just now'. The pipeline history sidebar on the left shows the progression from Source to Deploy.

Step 9: Now let's build the same replication environment in the other region. (Follow the steps same as we did for creating pipeline in Virginia region)

- Use the same template and deploy an infrastructure using the same CloudFormation template.

The screenshot shows the AWS CloudFormation 'Create stack' wizard. In the 'Prerequisite - Prepare template' step, the 'Template is ready' option is selected. In the 'Specify template' step, the 'Upload a template file' option is selected, and a file named 'ecs-refarch-continuous-deployment.yaml' is chosen. The 'JSON or YAML formatted file' input field contains the path to the uploaded file.

- Edit the IAM policies like the same way we did for the N.Virginia region.

The screenshot shows the AWS IAM Role configuration page for the role 'Edurekaohio-DeploymentPipe-CodePipelineServiceRole-CG1ETDLOKP4A'. The 'Permissions' tab is selected, showing three attached policies: 'AmazonS3FullAccess', 'AmazonSNSFullAccess', and 'AWSLambdaBasicExecutionRole'. The 'AmazonS3FullAccess' policy is highlighted with a red box.

AWS DevOps Project Solution

- c) In the CodePipeline point the source to the appropriate s3 bucket in the same region.

Action name
Choose a name for your action

No more than 100 characters

Action provider
 Amazon S3

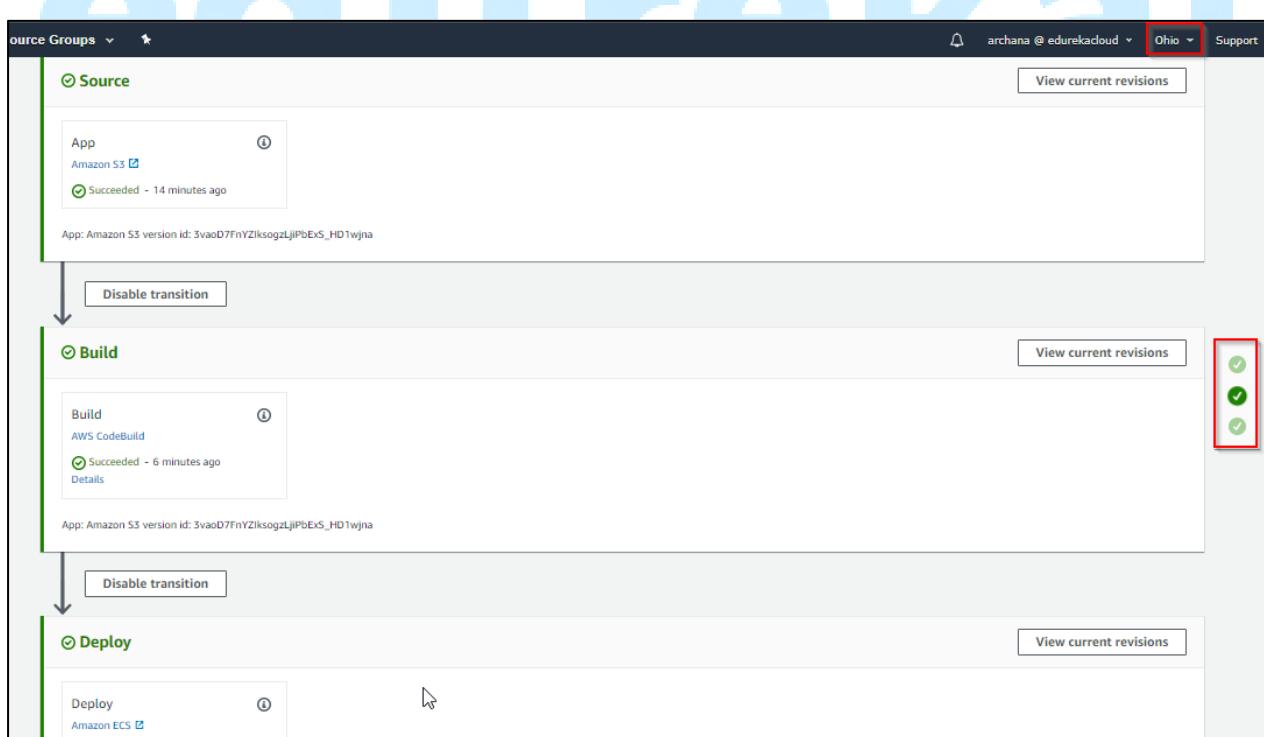
Bucket

S3 object key

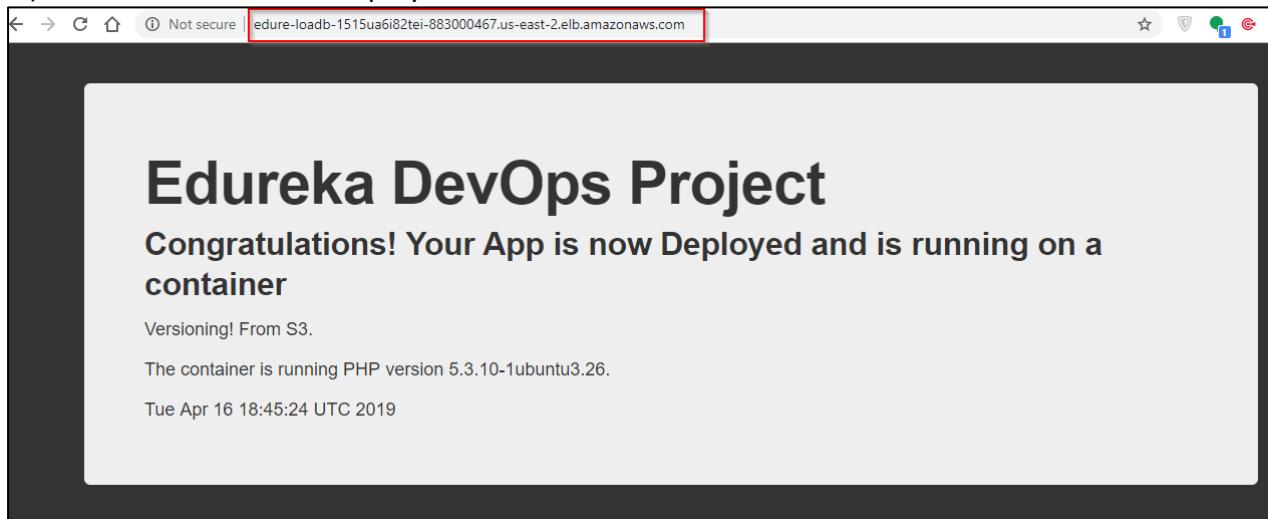
Change detection options
Choose a detection mode to automatically start your pipeline when a change occurs in the source code.
 Amazon CloudWatch Events (recommended)
Use Amazon CloudWatch Events to automatically start my pipeline when a change occurs
 AWS CodePipeline
Use AWS CodePipeline to check periodically for changes

Output artifacts
Choose a name for the output of this action.

No more than 100 characters



- d) Check the successful deployment in via DNS name of Load Balancer



- e) Create CloudWatch and SNS notifications the same way we did for the N.Virginia region.

SNS:-

The screenshot shows the AWS SNS "Subscription" details page for a subscription with ARN `arn:aws:sns:us-east-2:245376966395:CloudPipelineNotification2:fb0439f9-09d1-4347-9d6e-cb44eaddee1e2`. The "Details" section includes:

ARN	Status
<code>arn:aws:sns:us-east-2:245376966395:CloudPipelineNotification2:fb0439f9-09d1-4347-9d6e-cb44eaddee1e2</code>	<input checked="" type="checkbox"/> Confirmed
Endpoint	Protocol
[REDACTED].co	EMAIL
Topic	
<code>CloudPipelineNotification2</code>	

CloudWatch:-

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

Event Pattern i Schedule i

Build event pattern to match events by service

Service Name: **CodePipeline**
Event Type: **CodePipeline Pipeline Execution State Change**

Any state Specific state(s)

Event Pattern Preview:

```
{
  "source": [
    "aws.codepipeline"
  ],
  "detail-type": [
    "CodePipeline Pipeline Execution State Change"
  ]
}
```

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

SNS topic

Topic*: **CloudPipelineNotification2**

Add target*

f) Add the Approver Stage to the Pipeline.

Edit action

Action name
Choose a name for your action
approver
No more than 100 characters

Action provider
Manual approval

Configure the approval request.

SNS topic ARN - optional
arn:aws:sns:us-east-2:245376966395:CloudPipelineNotification2

URL for review - optional
Type the URL you want to provide to the reviewer as part of the approval request. The URL must begin with 'http://' or 'https://'.
[Empty input field]

Comments - optional
Comments you type here display for the reviewer in email notifications or the console.
[Empty input field]

Expected results:

The Pipeline should run in both the regions and you will get separate notifications for approving the change from the 2 different SNS notifications.

Conclusion:

We have successfully created the infrastructure, monitored it completely and ensured high availability by successfully deploying the application in two different regions.

edureka!