CI/CD with AWS Elastic Beanstalk, Code commit and CodePipeline

**Prerequisites**

* Download sample node.js code

[**https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/tutorials.html**](https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/tutorials.html)

* Download git for windows

**Step 1. Create IAM role for ec2 and Beanstalk to communicate on your behalf**

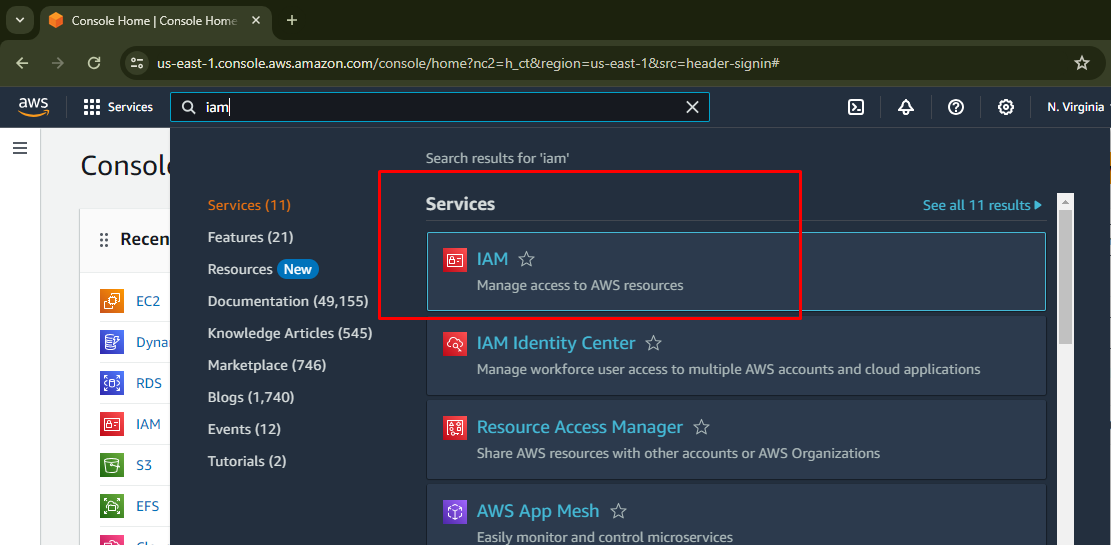
Configure IAM role

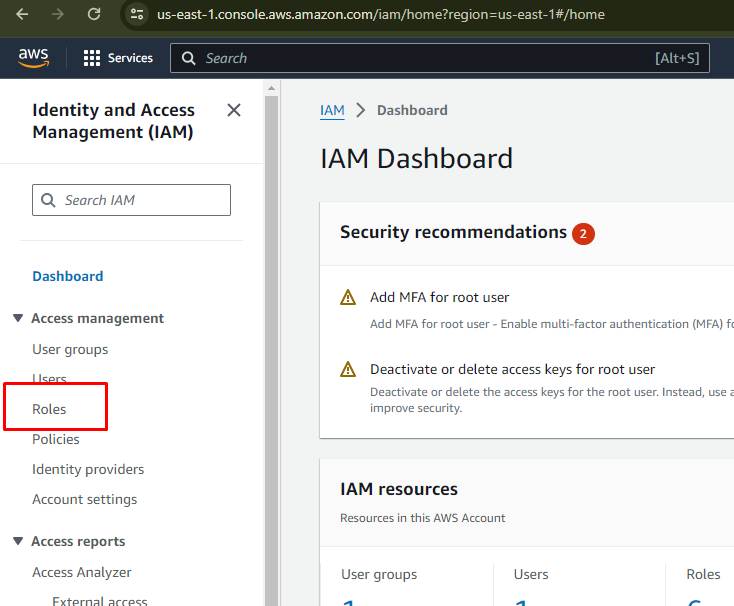
For Ec2: Select permissions Shown below

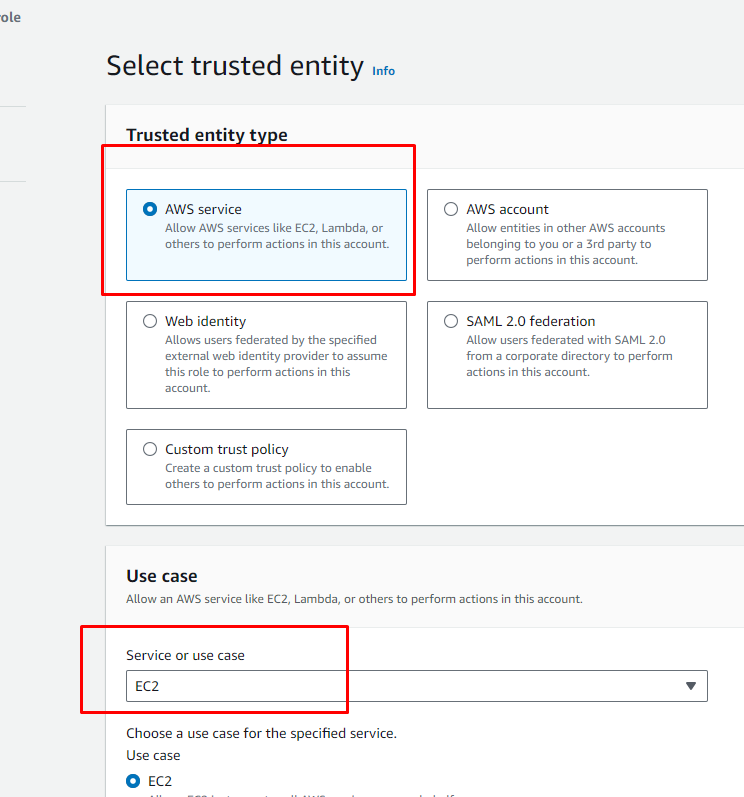
1. AWS Elastic beanstalk multicontainer

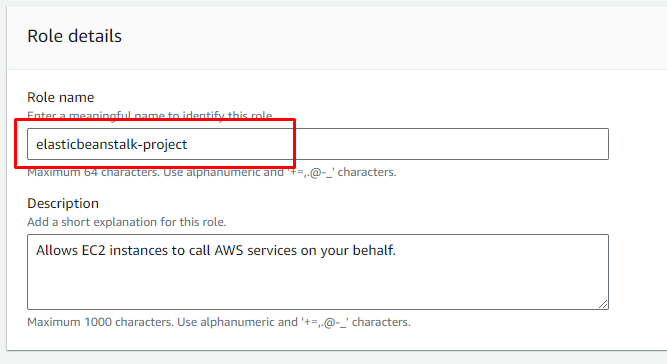
2.Aws elastic beanstalk webtier

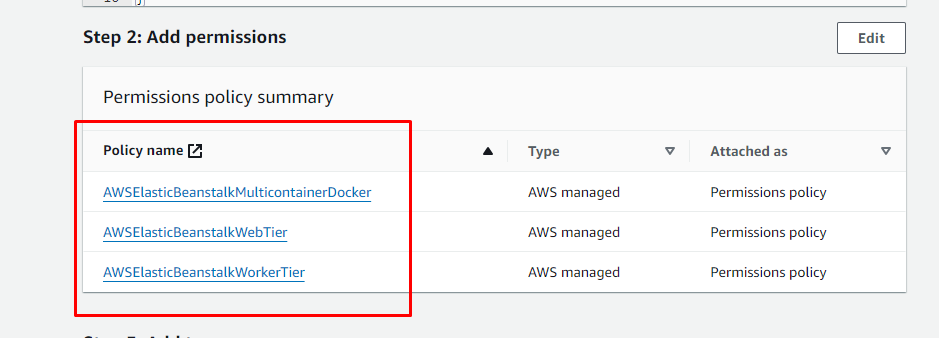
3.Aws Elastic beanstalk worker tier

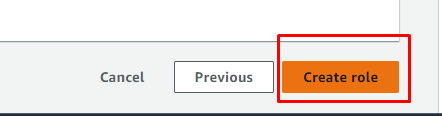








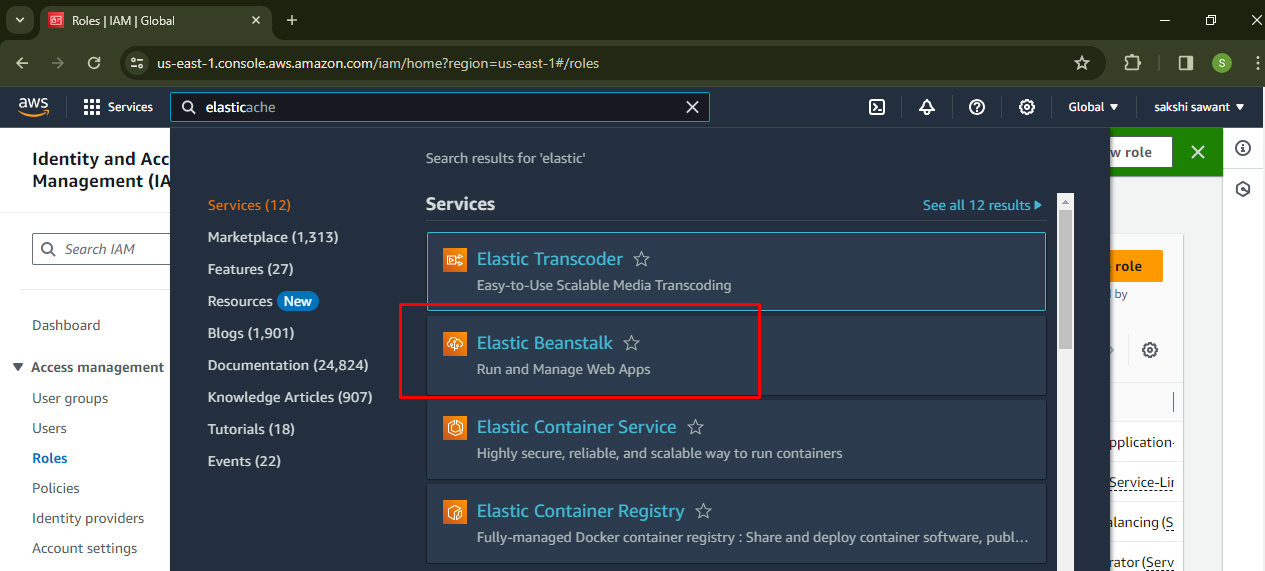


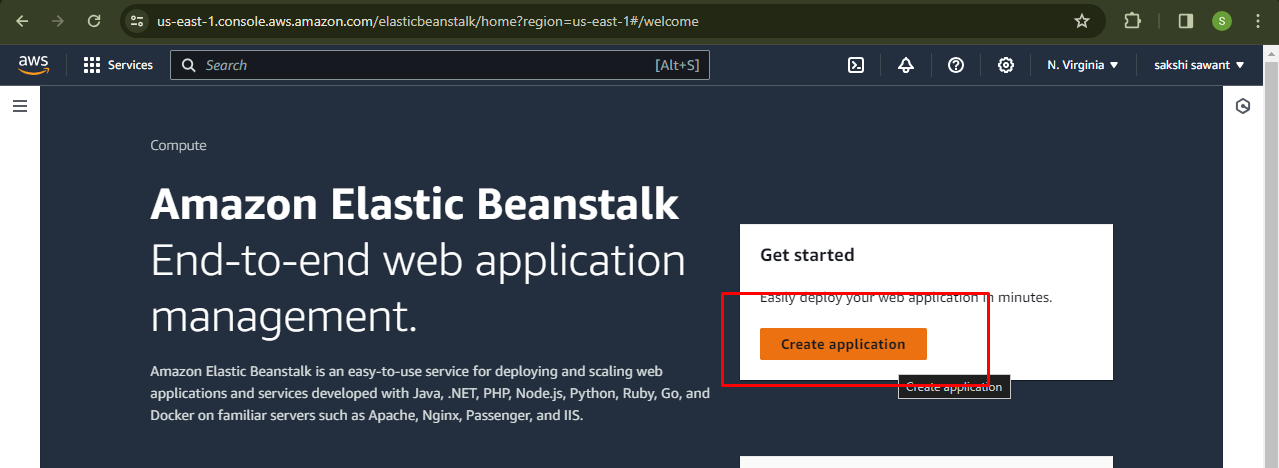


**Step2: Create a deployment environment Environment**

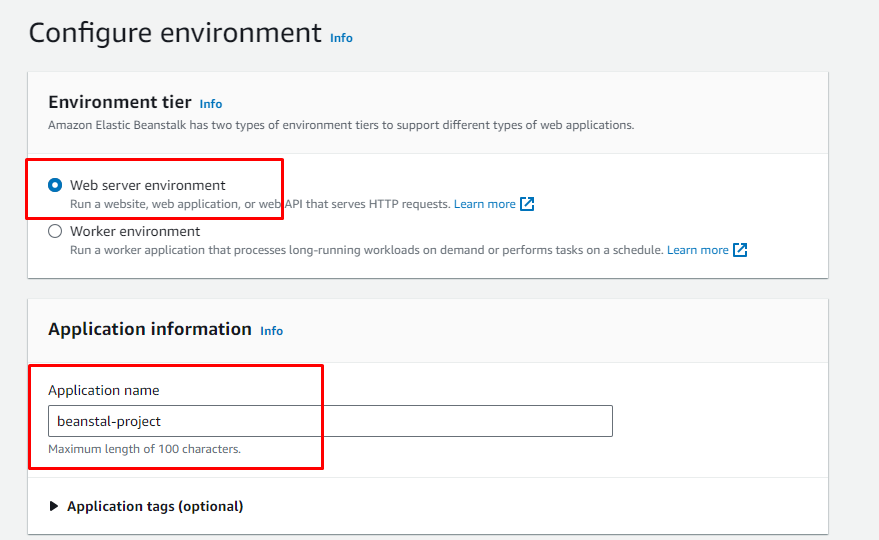
In continuous deployment, you need a deployment environment which can be either an EC2 server or a Docker container or Elastic Beanstalk (which can handle the environment configuration and bootstrapping automatically). In this case, I am going to use Elastic Beanstalk.

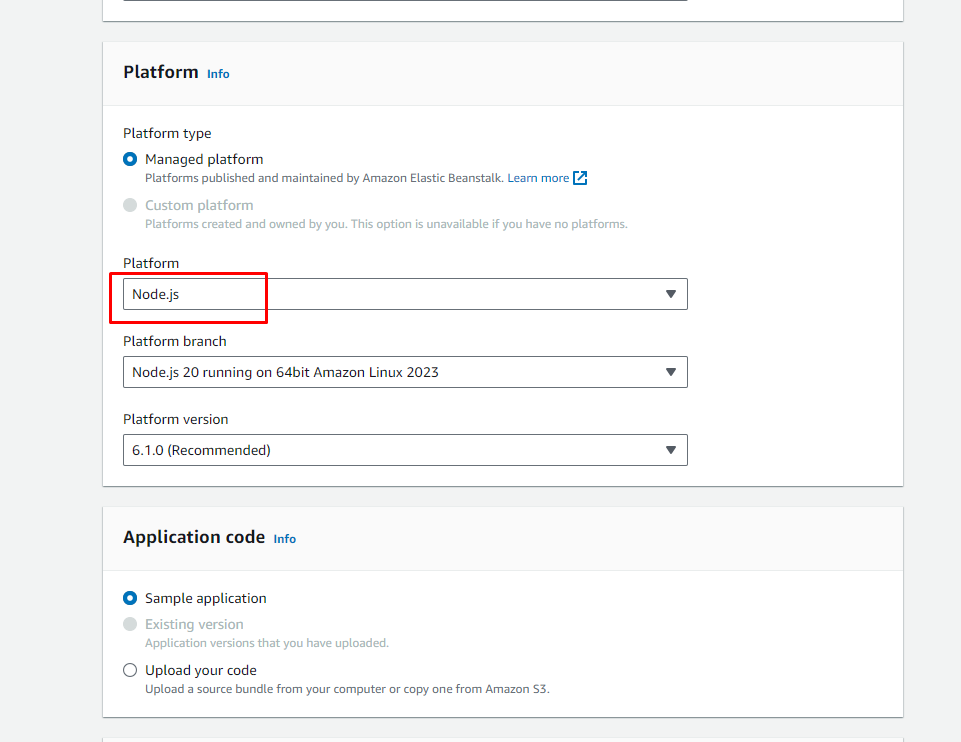
Elastic Beanstalk will host web application without the need to launch, configure, or operate. You don’t need to worry about the infrastructure, Elastic Beanstalk will take care of everything. I am going to deploy a Node js application on Elastic Beanstalk. So, let’s create an environment for Node js application with a single instance -

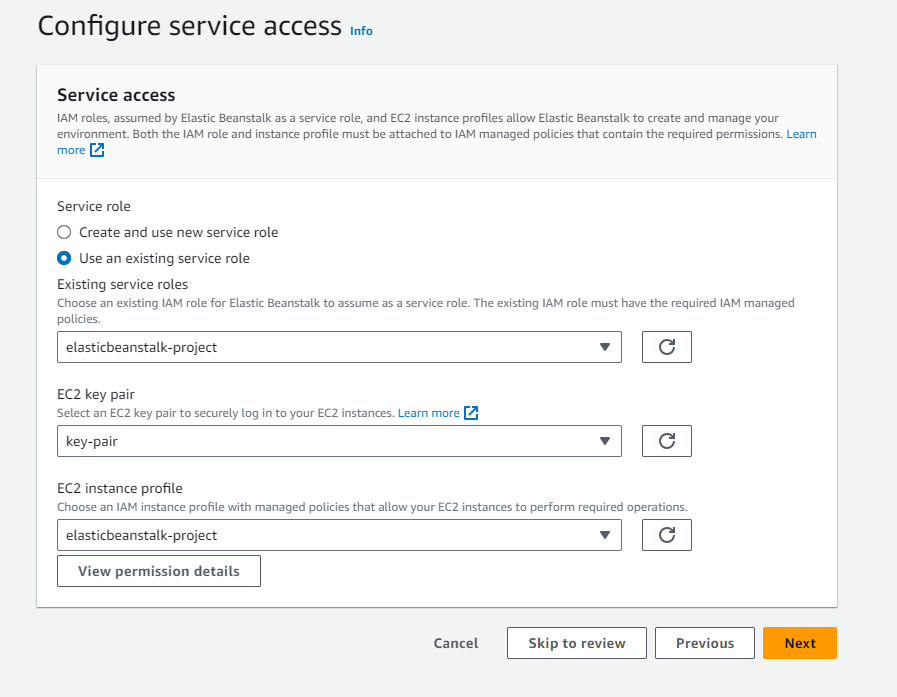


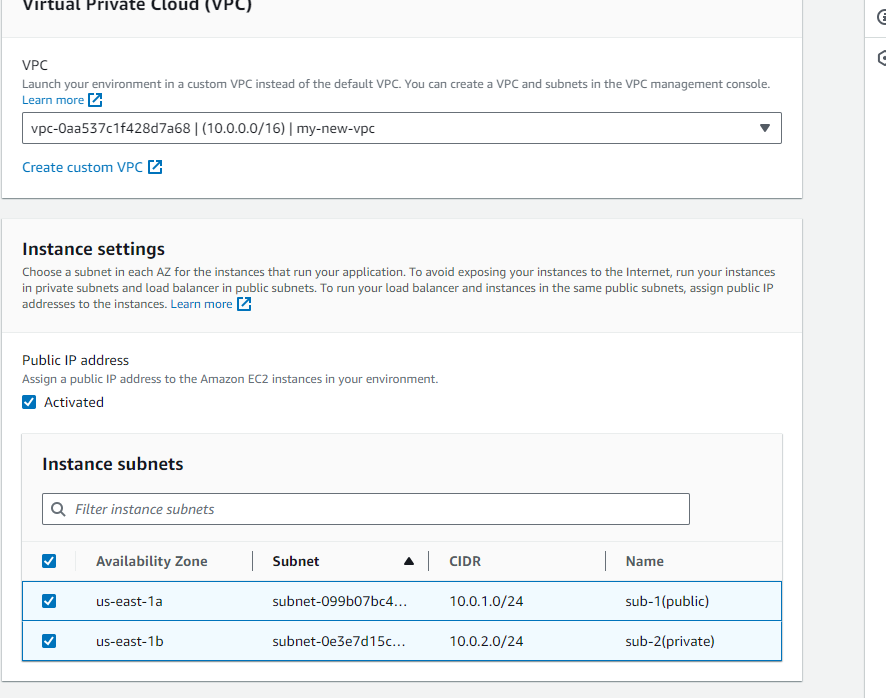


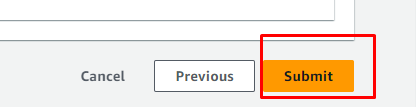
give a name for your application Select Node js. as a platform and other environment parameters like single instance, Service role,vpc, security group and keys in the advanced options.





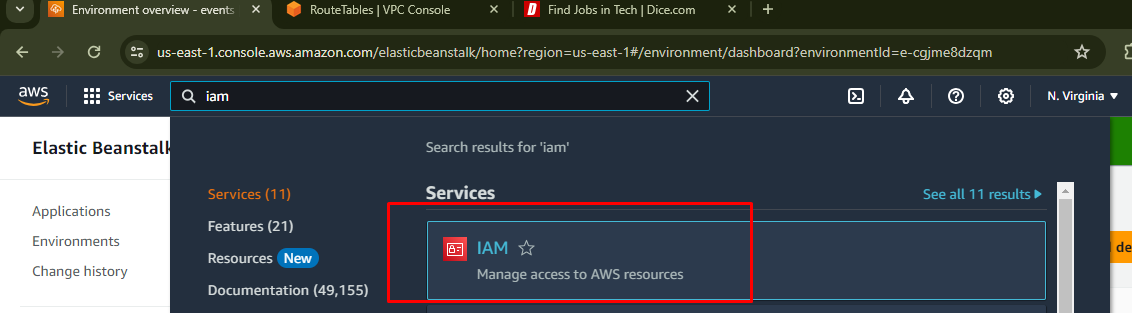


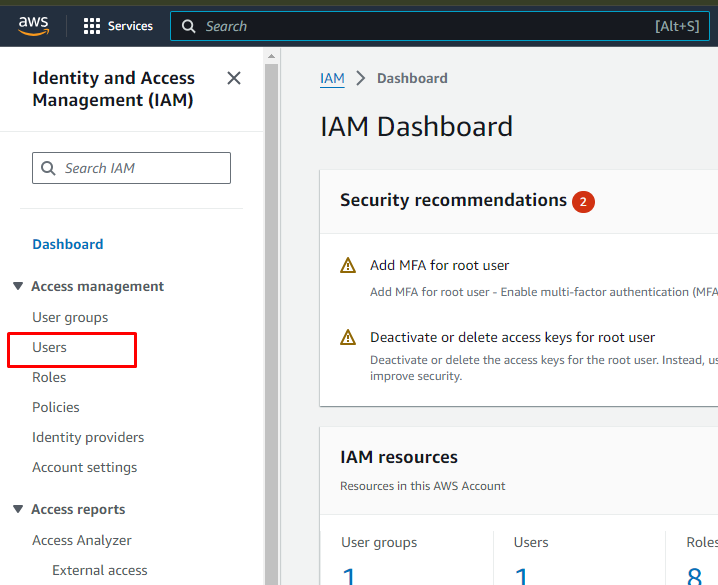


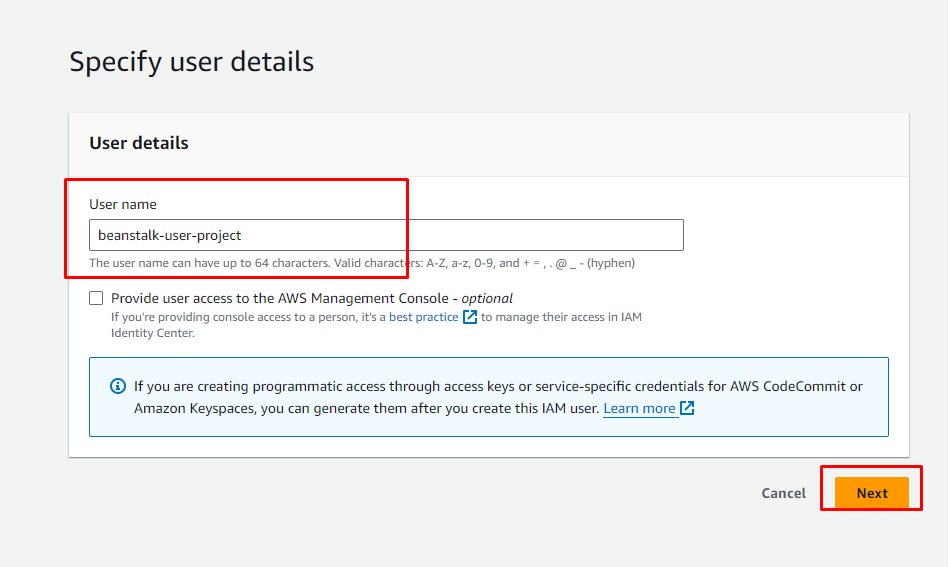


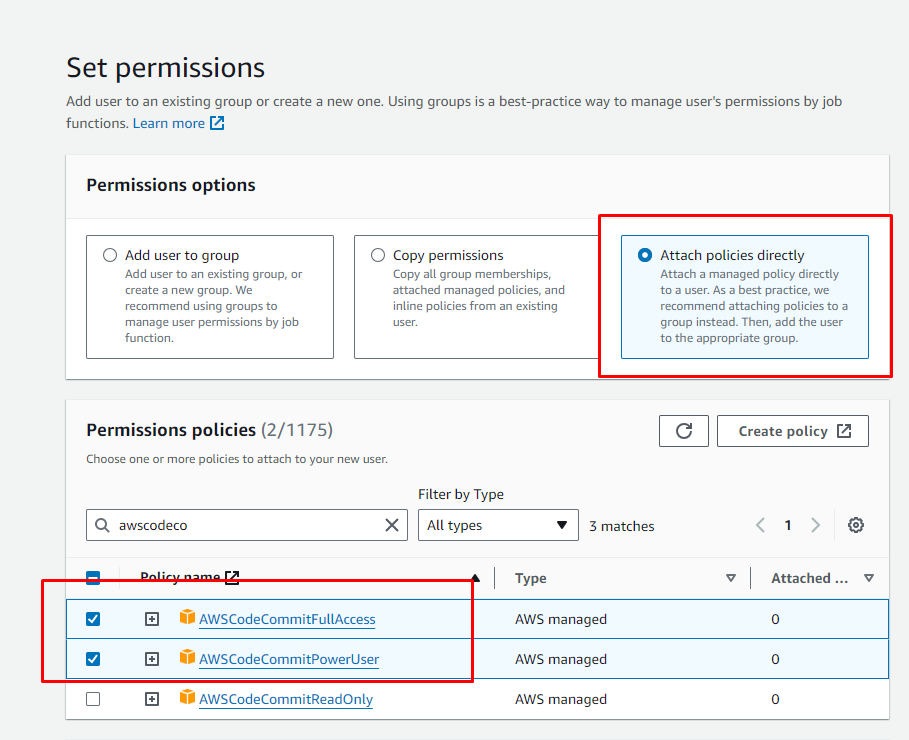
Once you have the environment, you can see the health check status and check the configurations of the environment. It will show the node js. configuration items you’ve selected during setup.

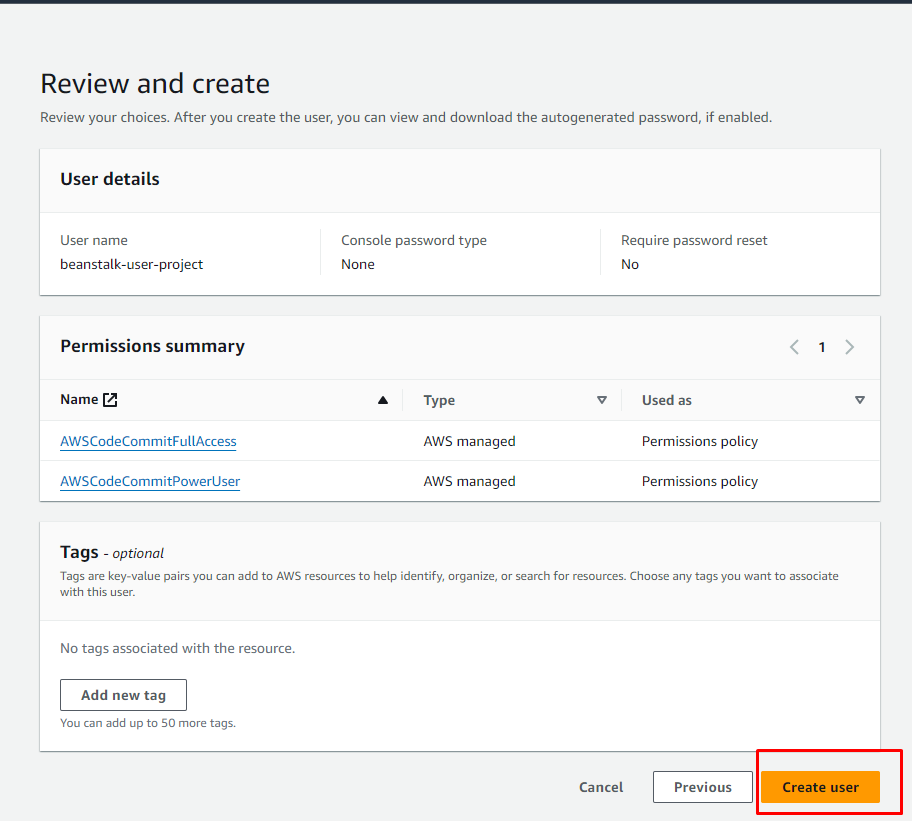
**Step 3 : Create an Iam user**

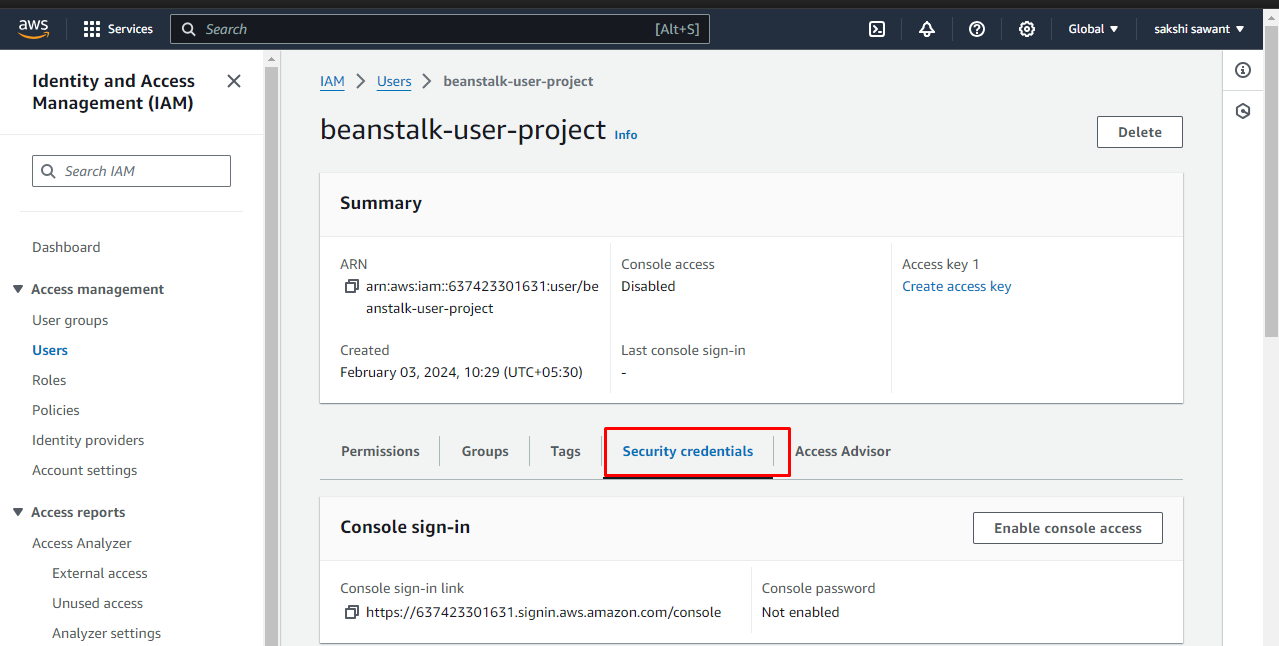


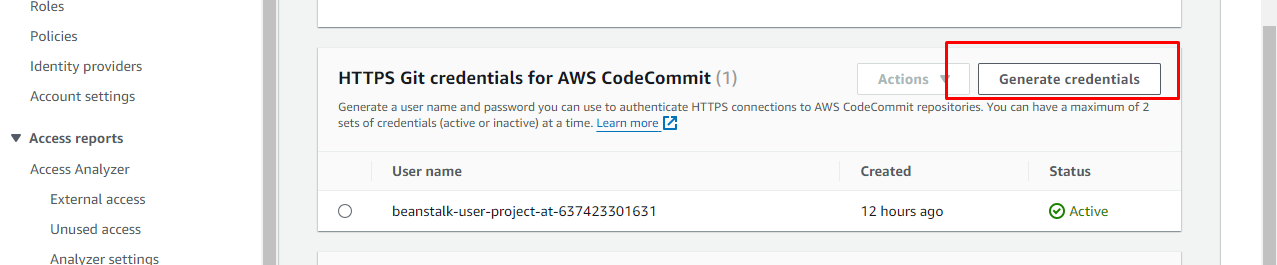








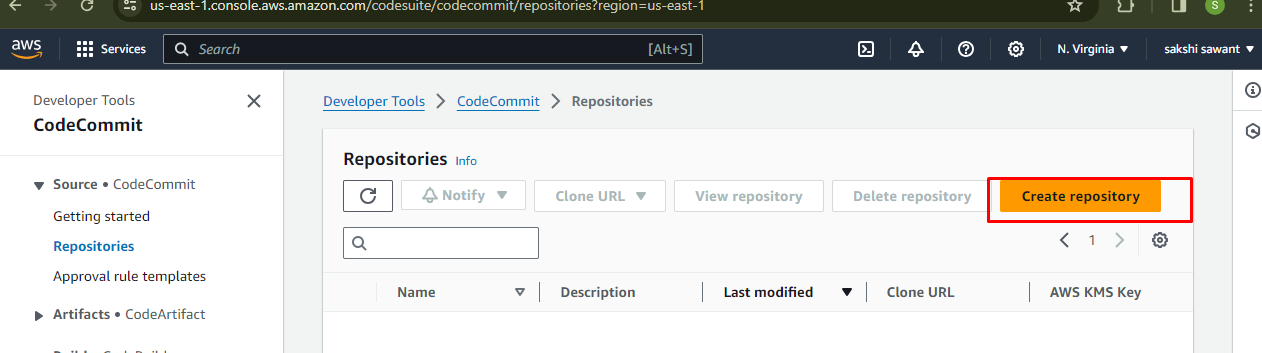


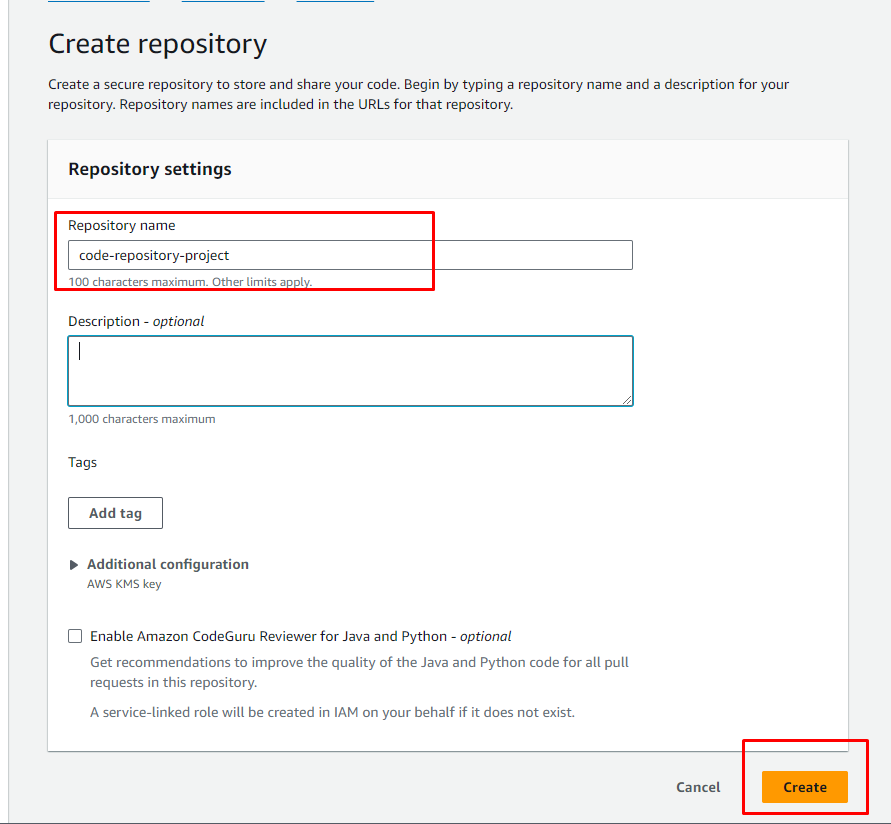


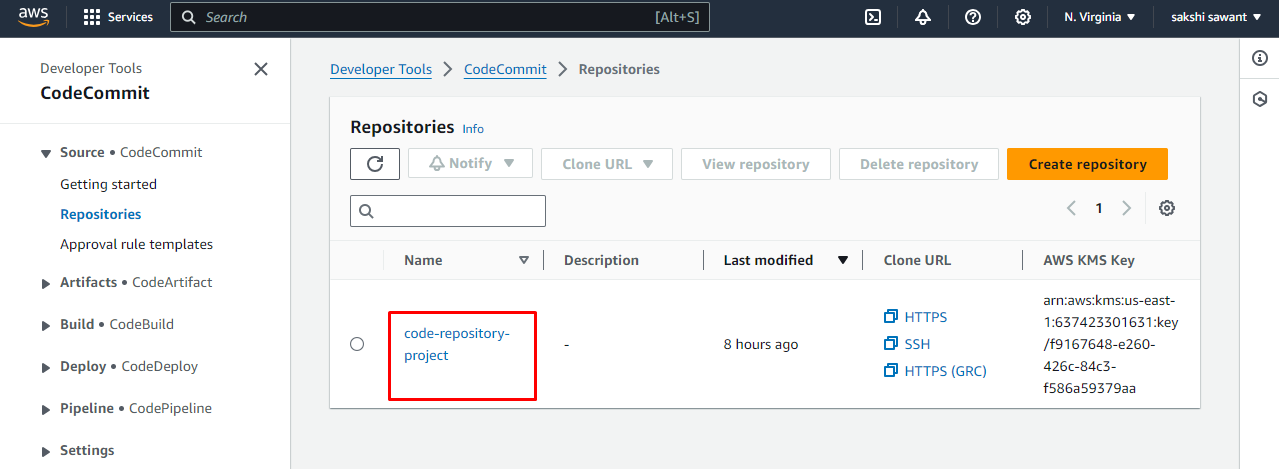
**Step 4 : Create code commit Repository**

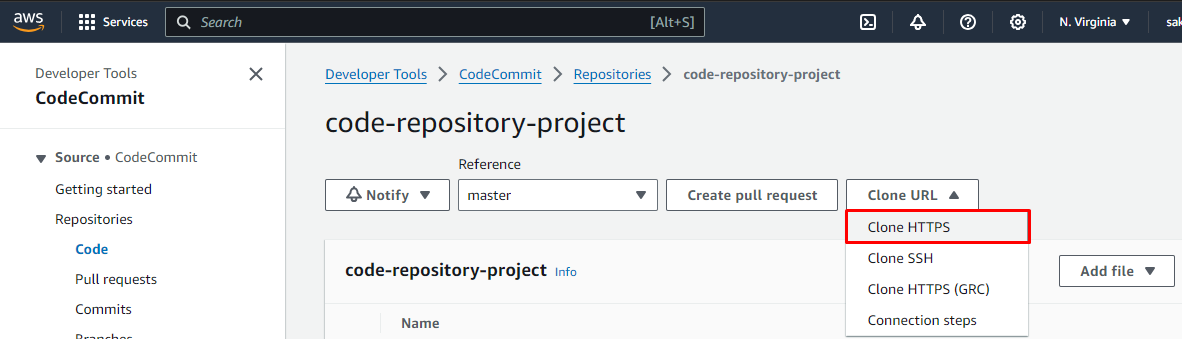
**Clone the code from codecommit repository  
I have a sample node js application which I am going to deploy on Elastic Beanstalk and along with that, I will configure CodePipeline. This pipeline will take the source code from your codecommit repository and perform actions on it. If there is any change in the source code it will directly deploy that change to servers.**

**Here, you have three options for source code viz., GitHub repository, Amazon S3. You can select any one out of. For this particular case, I am going to use code commit .**



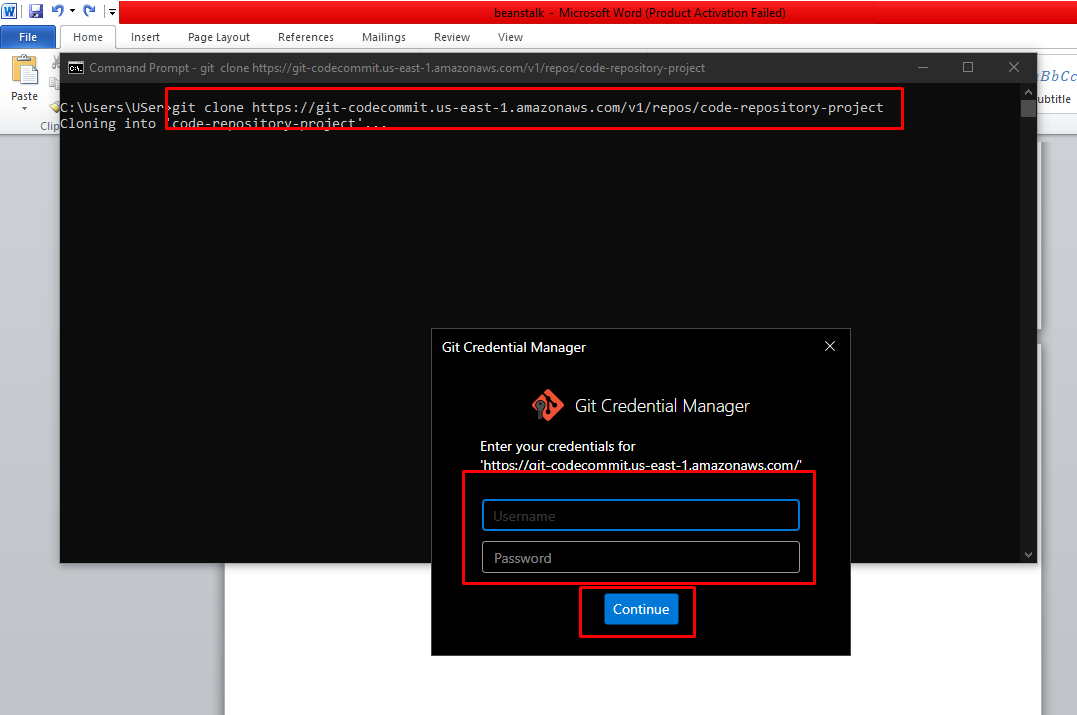




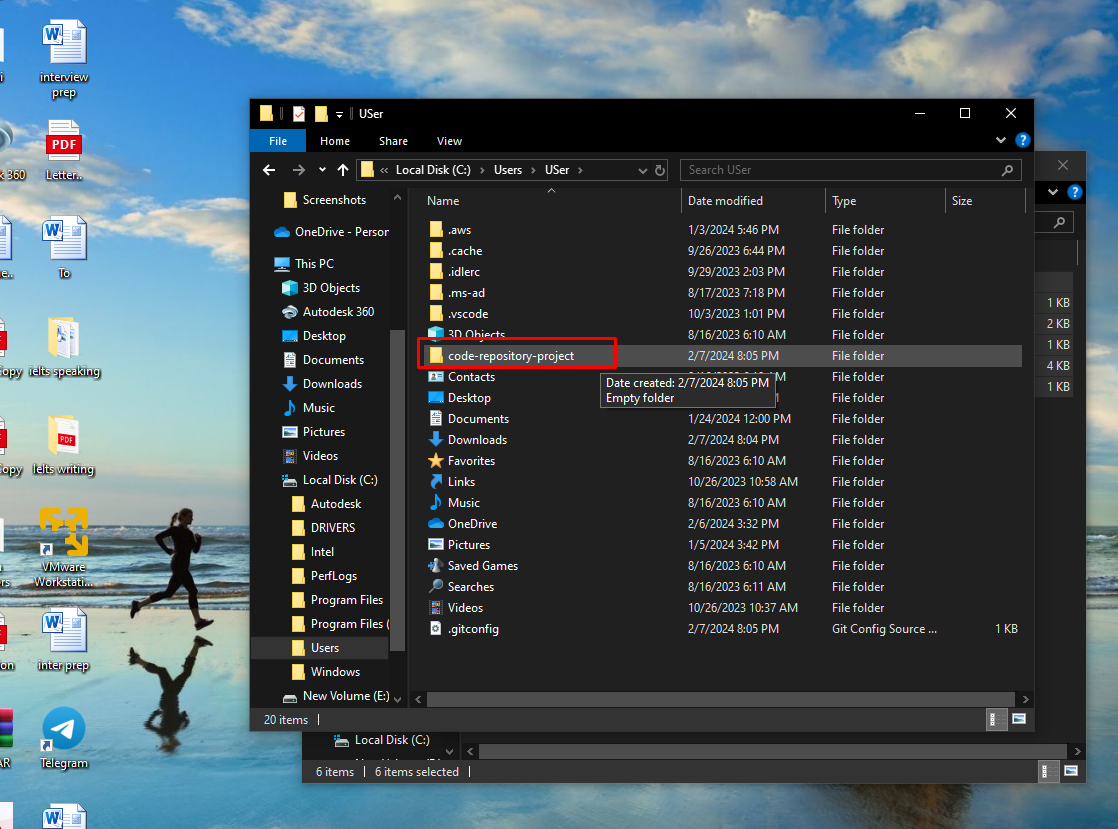


Step 5: **Clone the code to code commit repository**

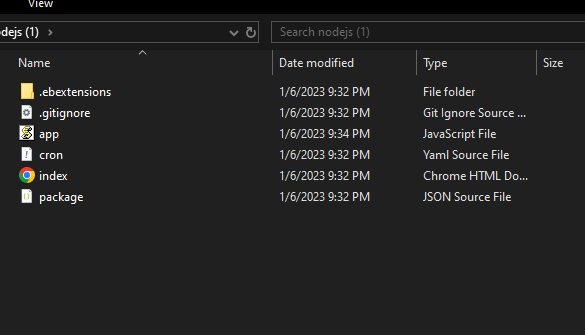
* Open your command prompt
* check git is present on your computer
* Command : git –version
* If yes then perform the next command
* Command: git clone url\_you *just*\_copied
* Now enter the Credentials you generated after creating a user for codecommit.



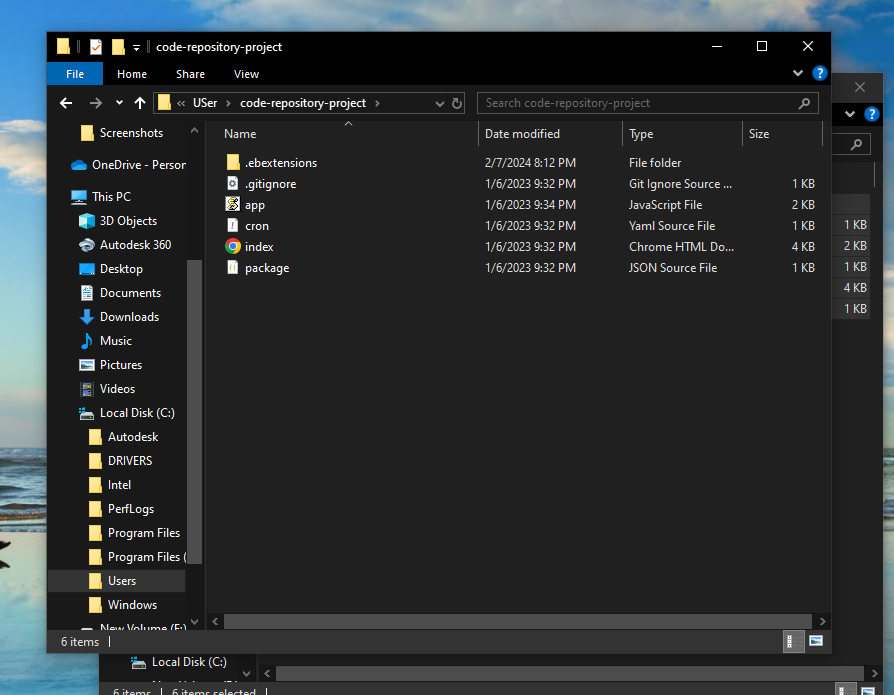
* One folder now has been created on your local same name as you gave to code commit repository



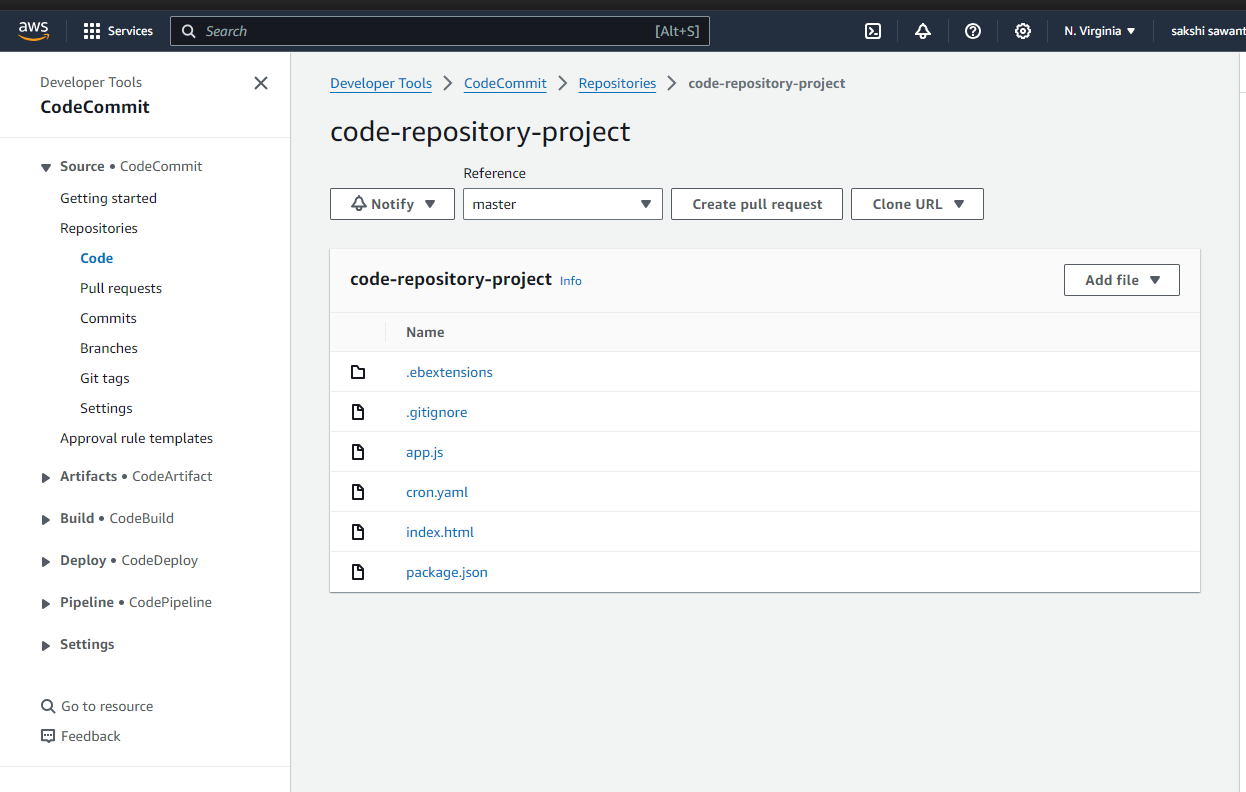
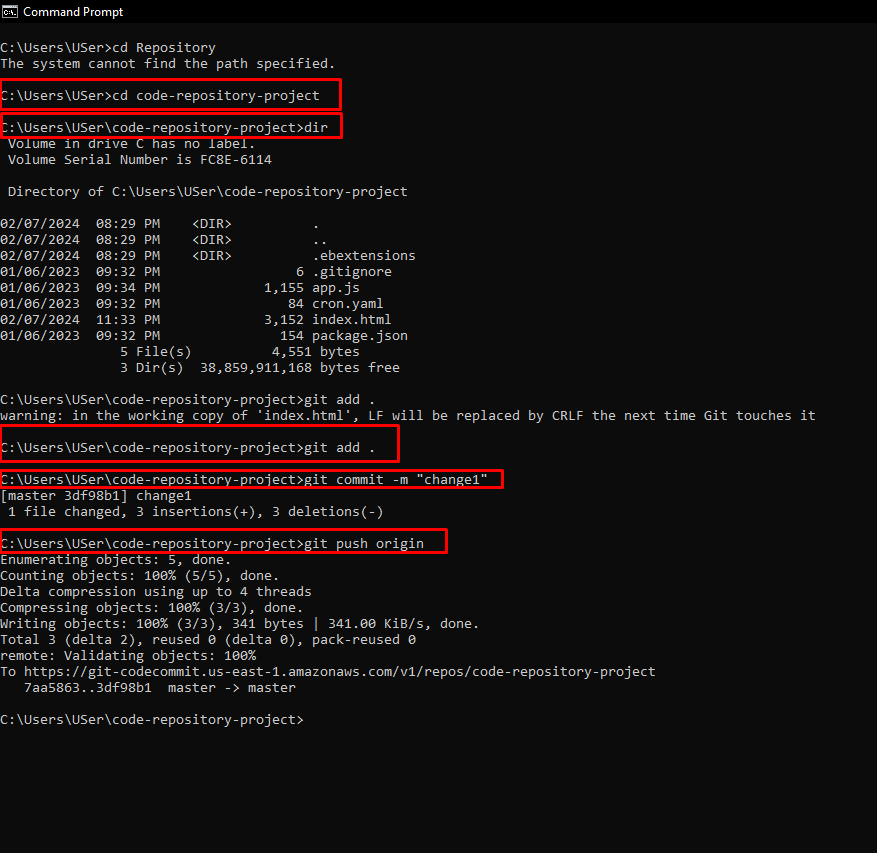
now we will open the folder where we kept our node js file in our local system and copy all the files



* Here in the new code commit folder we will paste it



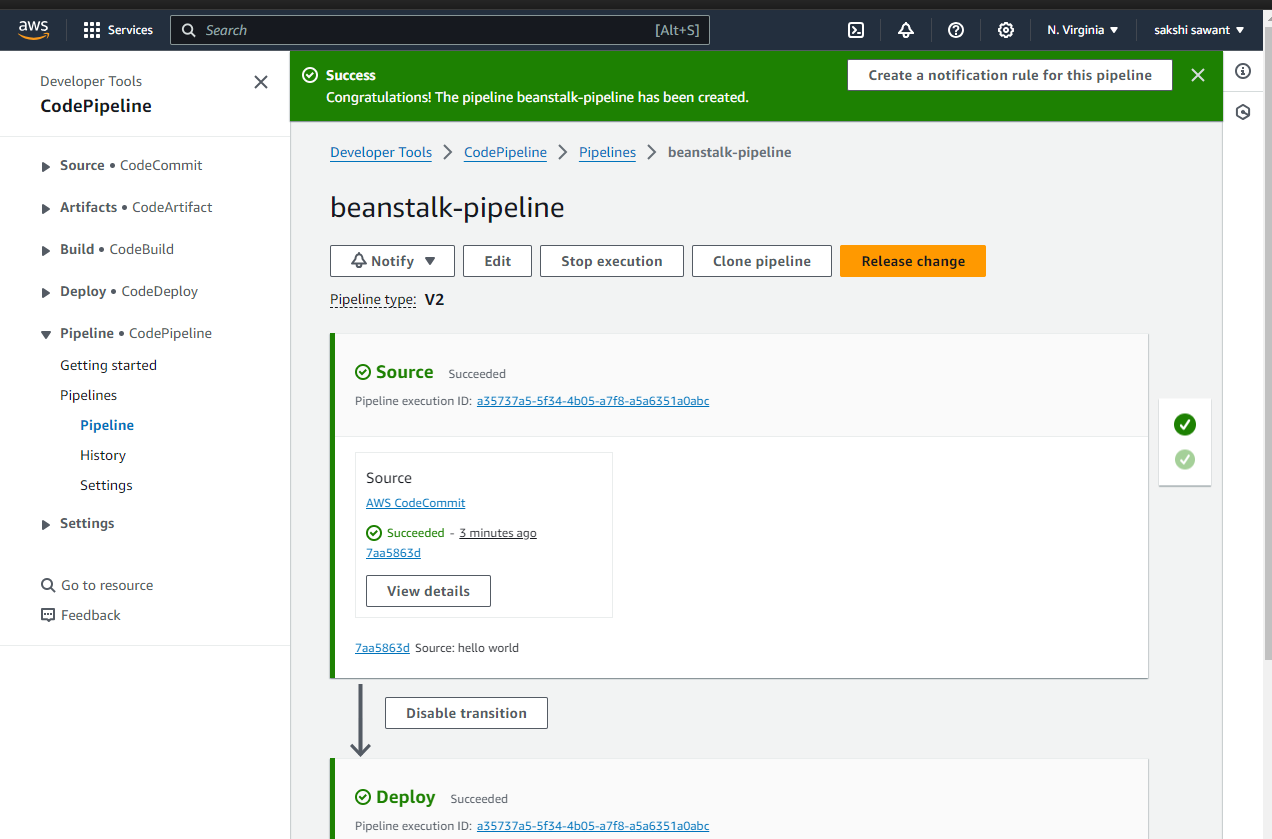
* Open command prompt and perform the command shown below to push your local code to code commit repository

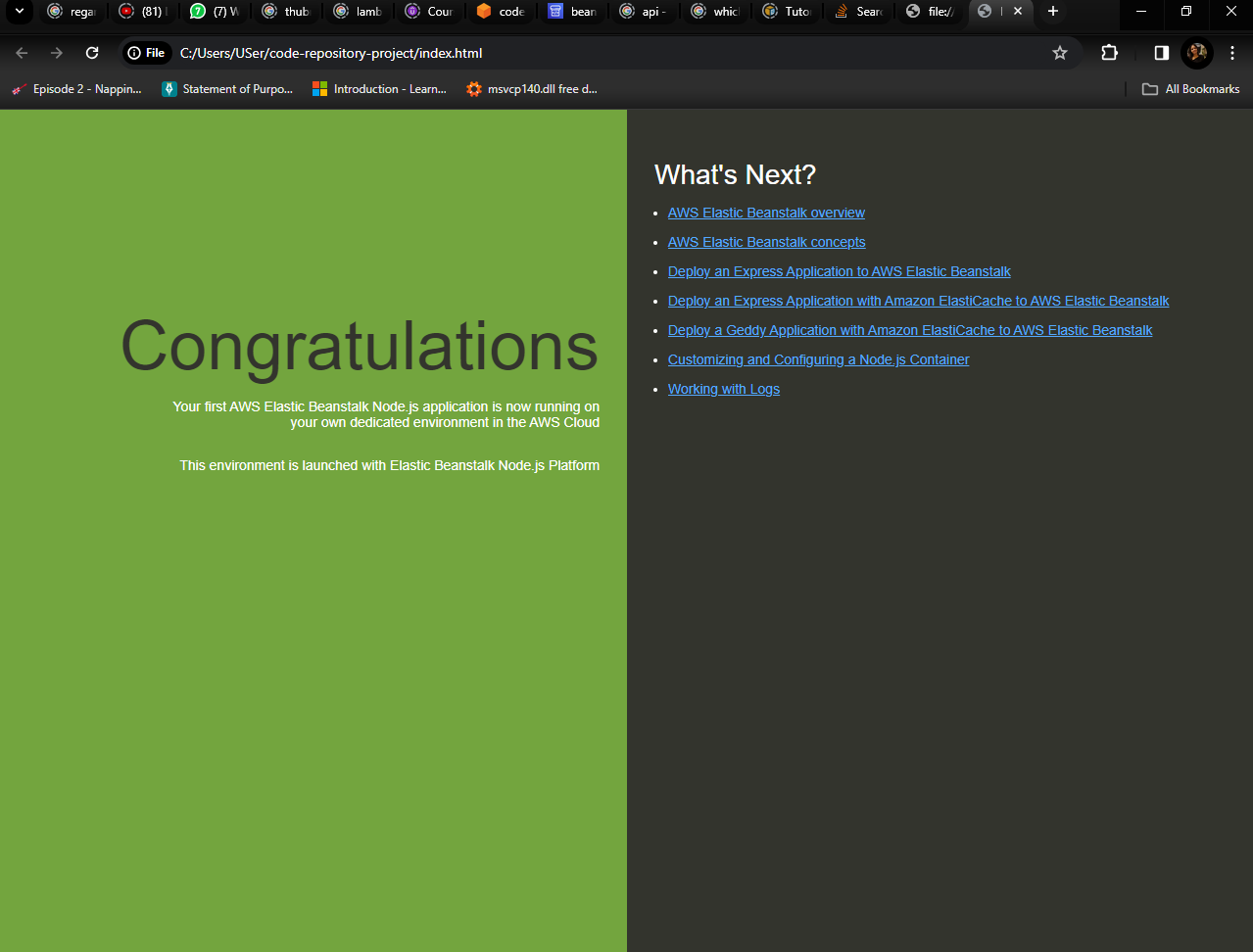


Step 6: Create a CodePipeline

Create a CodePipeline, which will be used to build, test, and deploy your code every time there is a code change based on release configuration which you have defined. This enables you to rapidly deliver your features and updates to deploy server.

a. Go to CodePipeline service, click on “create pipeline” and name it accordingly





Step 7 : Commit the changes and update your app

You can revise your code and commit the changes to the repository. CodePipeline will detect your updated sample code and then automatically initiate deploying it to your EC2 instance via Elastic Beanstalk.

Let’s update your index.html page, commit changes, and after some time, you will see the pipeline being updated. It will automatically pull the updated code and will start reflecting the changes.

