[**Continuous Integration for a Node.js Application using Git, Jenkins, and AWS Elastic Beanstalk**](https://github.com/Vennilavan12/zen-class-devops-documentation/blob/main/012%20-%20Projects%26SampleUseCases/project%203.md#project-3-continuous-integration-for-a-nodejs-application-using-git-jenkins-and-aws-elastic-beanstalk)

Imagine you are working on a team developing a Node.js-based web application, and you want to streamline the development, testing, and deployment processes. Your team decides to implement a Continuous Integration (CI) pipeline using Jenkins and leverage AWS Elastic Beanstalk for deployment and scaling.

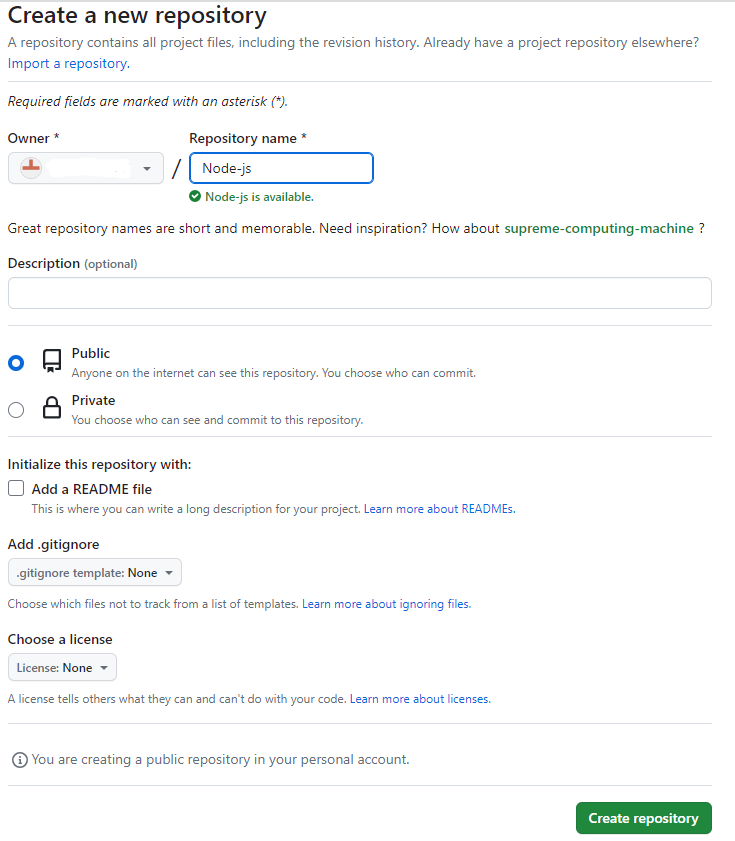
**Solution**

Project Requirements:

1. EC2
2. Git
3. Jenkins
4. AWS Elastic Beanstalk

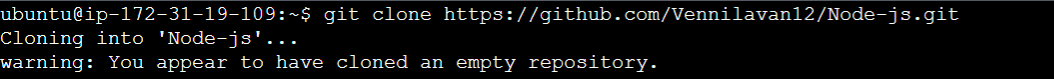
## **STEP 1:** [**Create a sample Node.js application with source code in a Git repository**](https://github.com/Vennilavan12/zen-class-devops-documentation/blob/main/012%20-%20Projects%26SampleUseCases/project%203.md#create-a-sample-nodejs-application-with-source-code-in-a-git-repository)

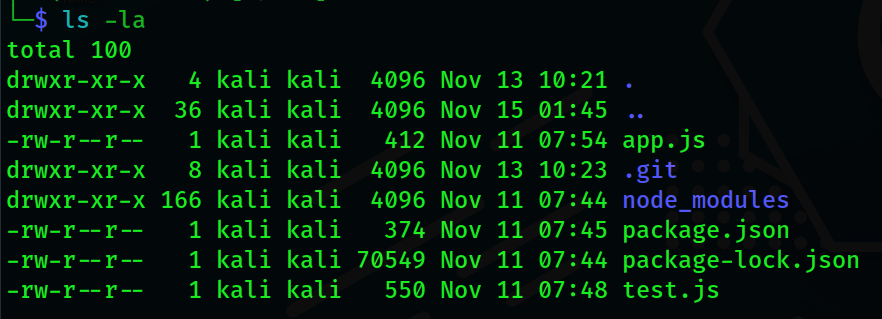
* Create a new repository into our github account.



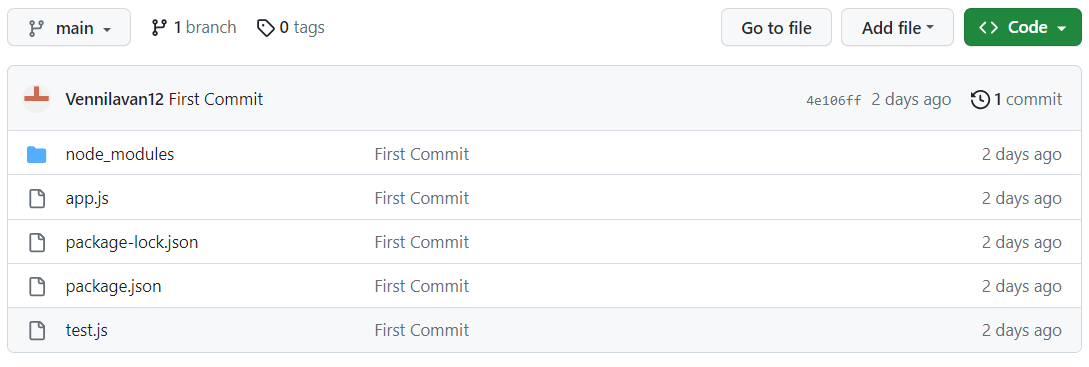
* Clone the repository into local using git clone method.

**git clone “Repo url”**





* Once the code has been ready we can move into next step.



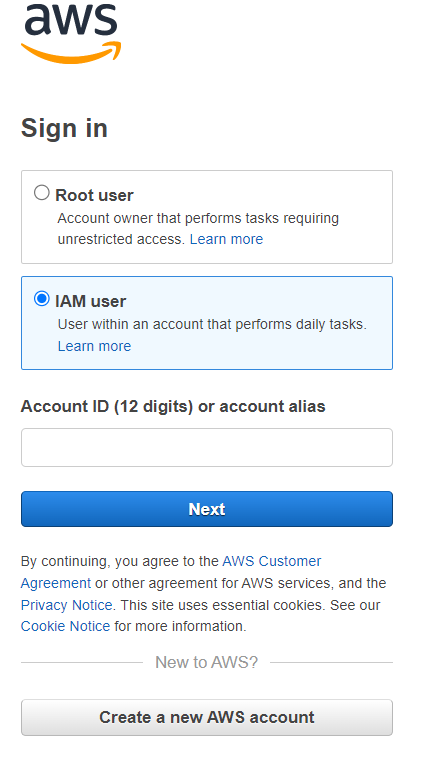
STEP 2: **Configure Jenkins to clone the Git repository, install dependencies, run tests, and package the application as a ZIP file**

Requirements :

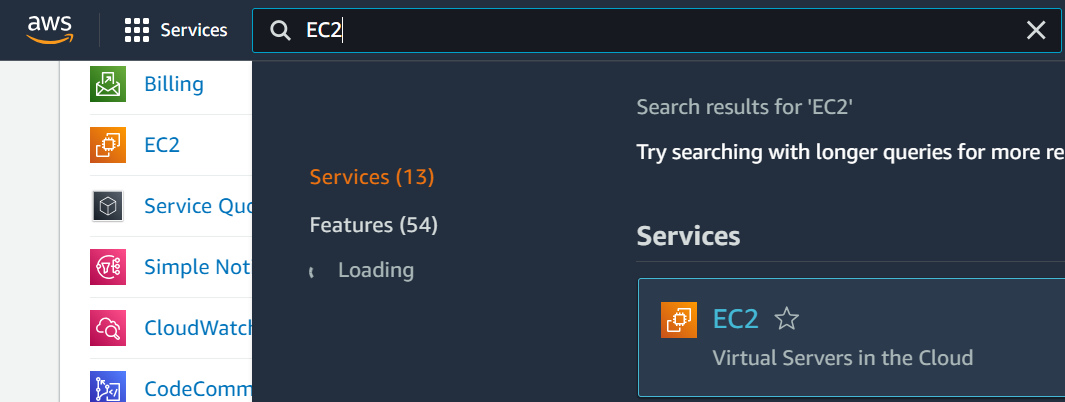
* 1. EC2
  2. Jenkins

**EC2**

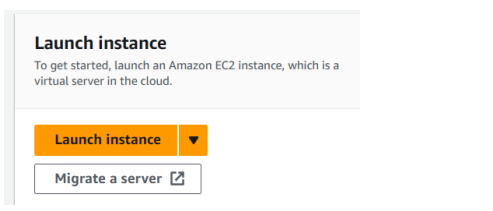
* Create EC2 instance in AWS for installing Jenkins server.
* Login to AWS Management console



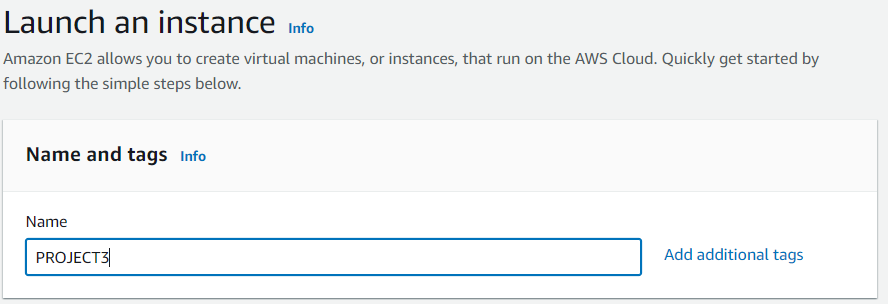
* Choose EC2 and click launch instance

****

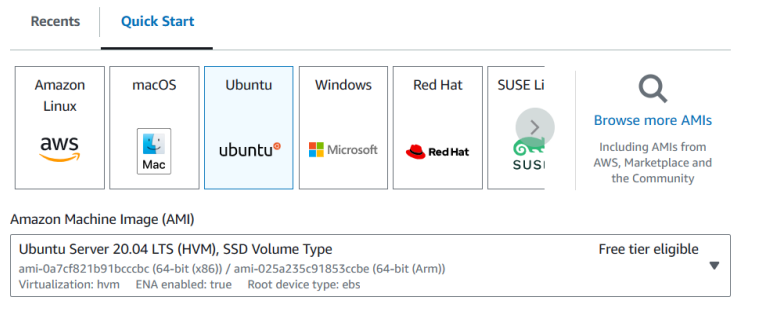
* Then under the EC2 management console we could able to find the Launch instance option, click that one for creating an instance:

****

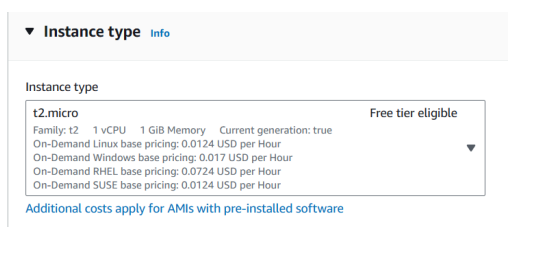
* Then name the instance according to your preferences,

****

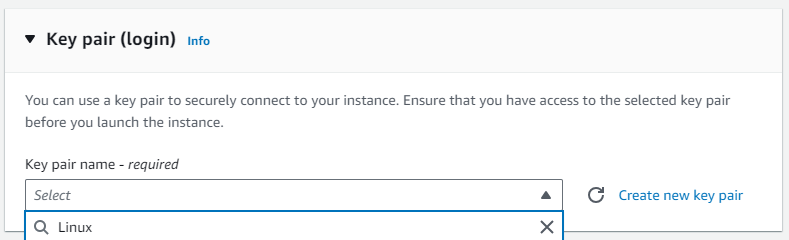
* Then we need to select an AMI Image for the instance, choose AMI image according to your preferences: Here I am selecting Ubuntu 20.04 AMI Image.

****

* Then we need to select the instance type

****

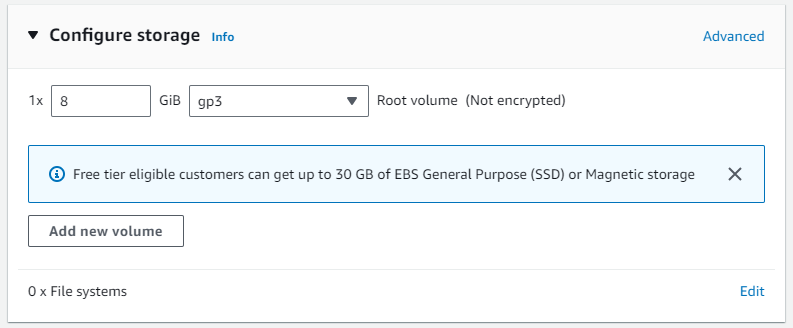
* Then we need to select the keypair for security authentication purpose of the instance

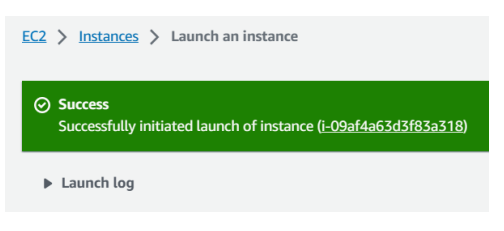
****

* Then under network settings, I am selecting default VPC, subnet no preference option, auto-assign public is enabled by default for default VPC & Subnet.

****

* Configure storage and launch a instance.

****

****

**Jenkins**

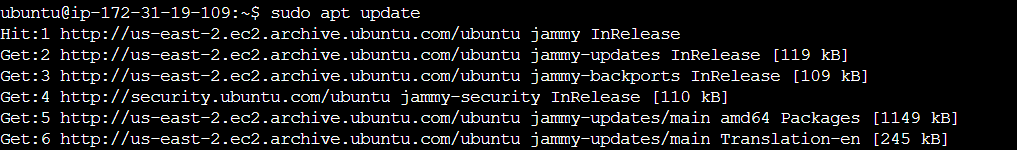
* Install a Jenkins in our EC2 machine.

## **Prerequisites**

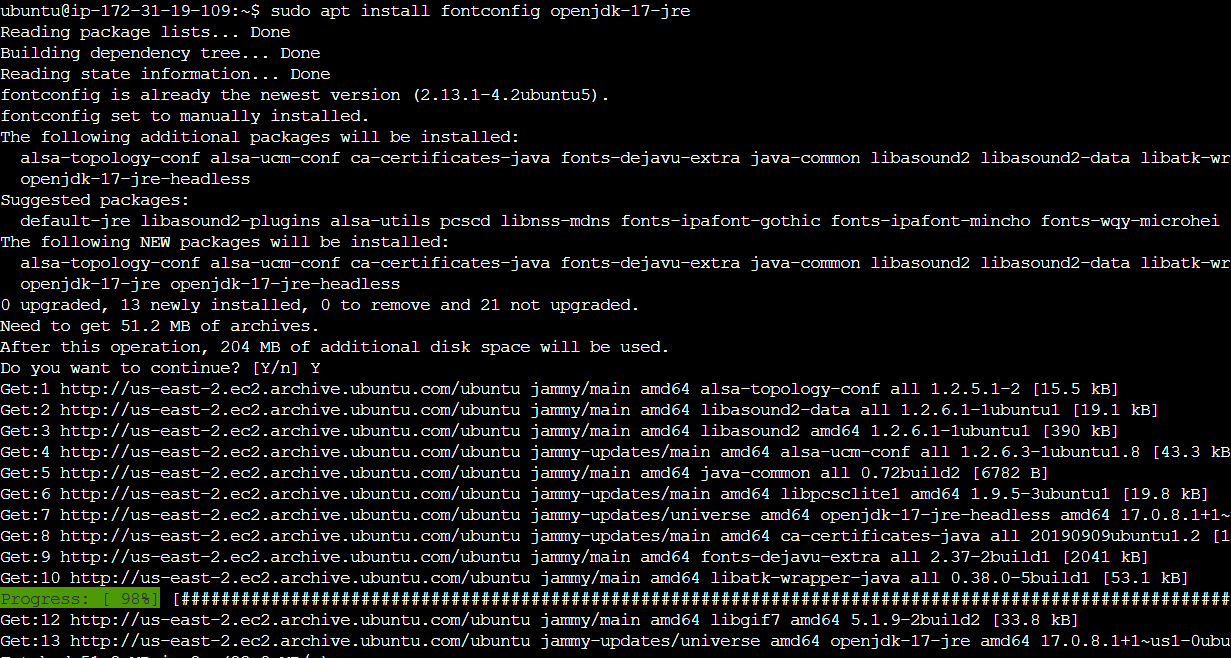
Minimum hardware requirements:

* 256 MB of RAM
* 1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)
* Jenkins requires Java to run, so we need to install java using this commands.

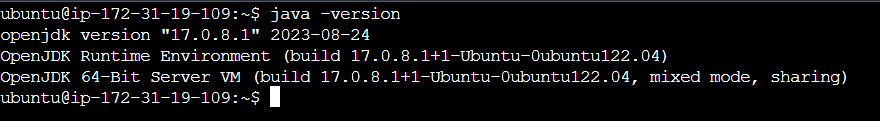
**sudo apt update**



**sudo apt install fontconfig openjdk-17-jre**



**java –version**



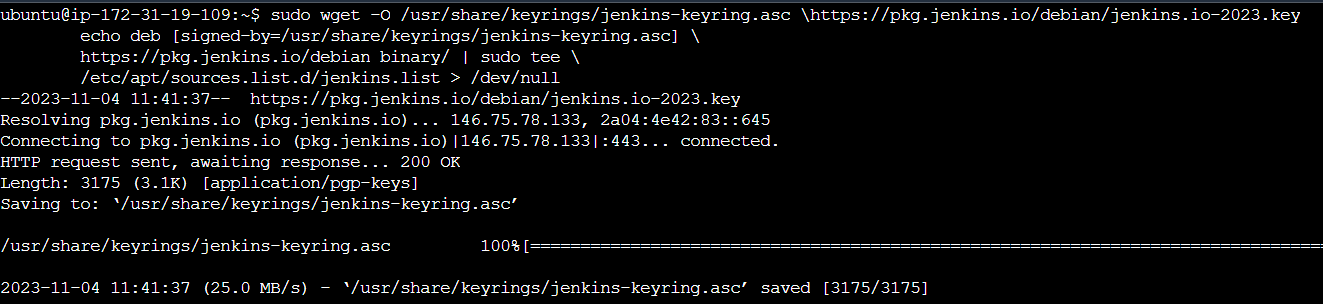
* Once java installation completes next start installing Jenkins using this commands.

**sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \https://pkg.jenkins.io/debian/jenkins.io-2023.key**

**echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \**

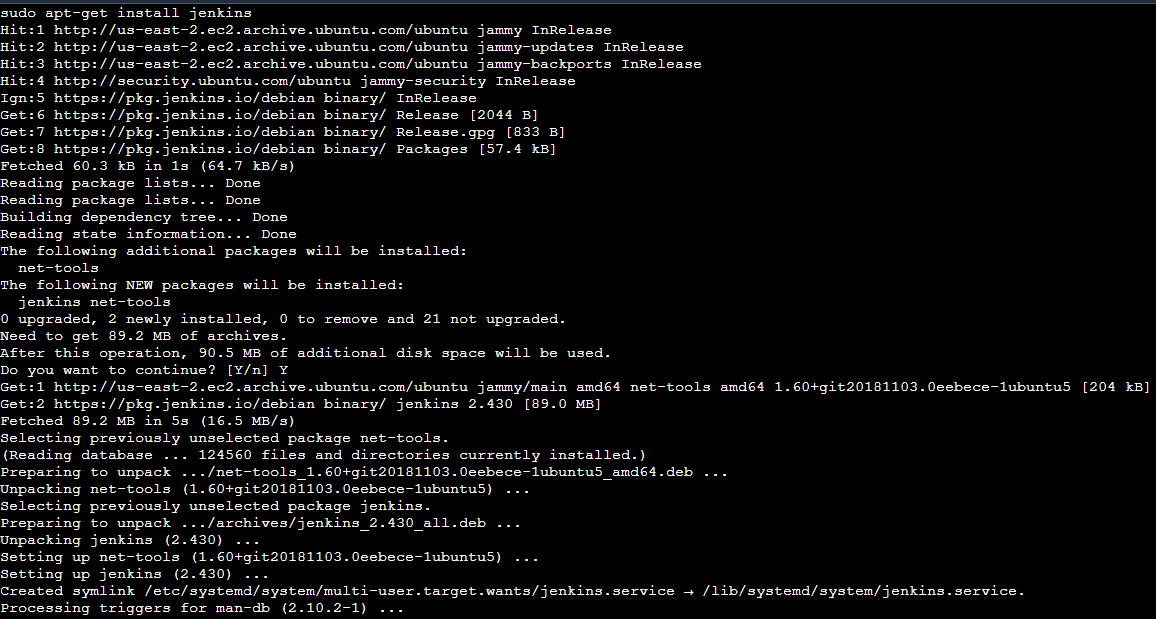
**https://pkg.jenkins.io/debian binary/ | sudo tee \**

**/etc/apt/sources.list.d/jenkins.list > /dev/null**

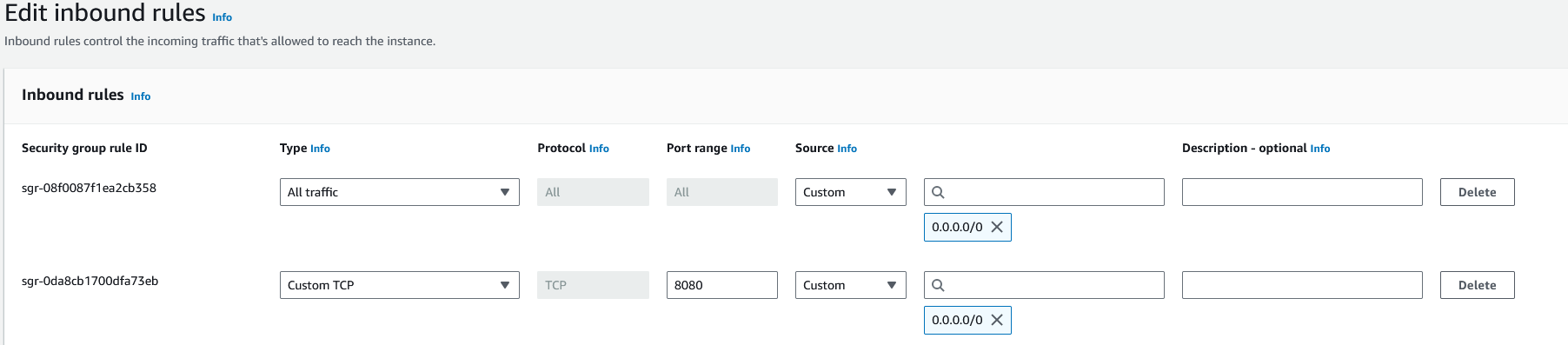


**sudo apt-get update**

**sudo apt-get install Jenkins**

****

* Once Jenkins installation finished we need to open port (8080) for accessing our Jenkins using aws security groups.
* Go to instance security groups and edit inbound rules and add 8080 with custom ip range and save.

****

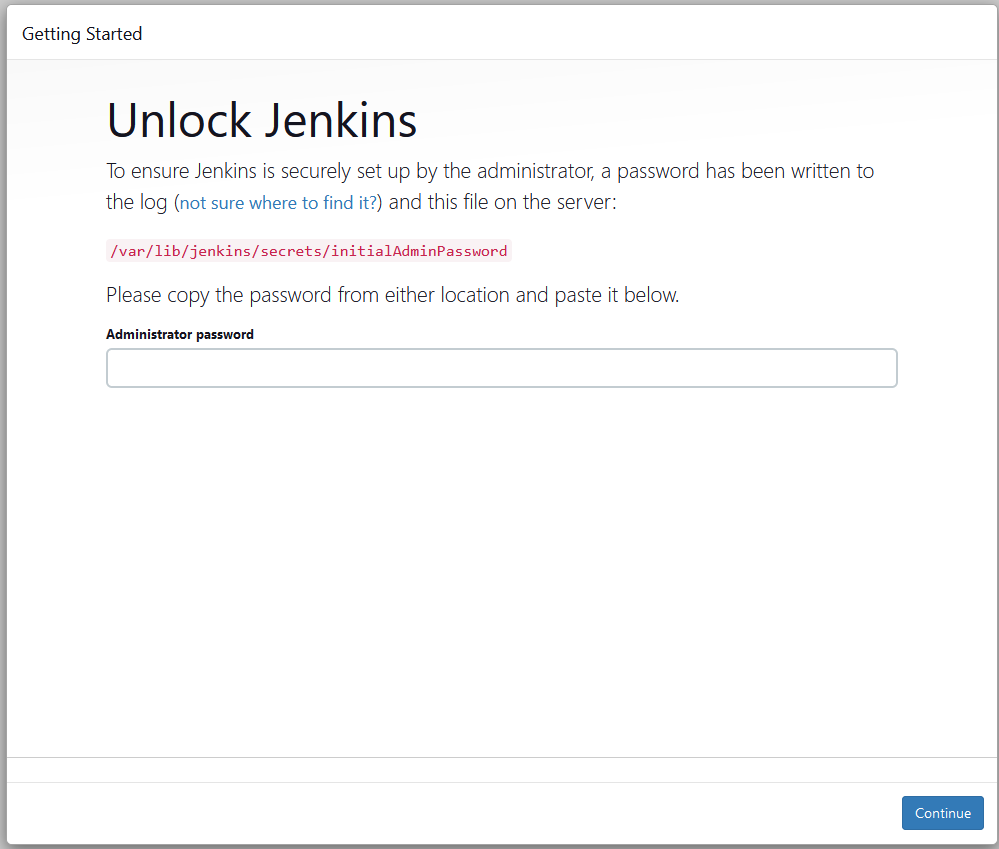
* Next Accessing our Jenkins server using our instance public ip follows with port 8080.

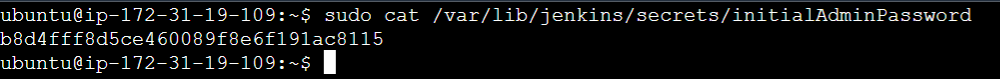
[http://IP:8080](http://ip:8080)



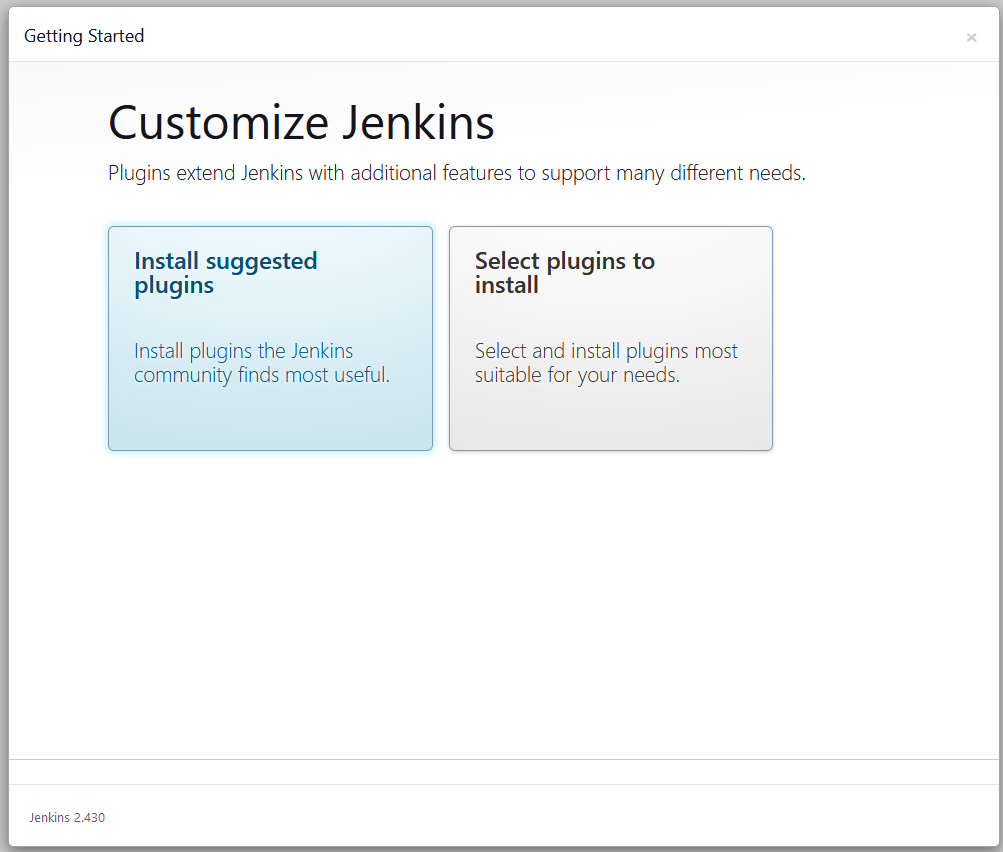
* Getting started Jenkins page will open paste our admin password.
* Use this command to get Jenkins admin password

**sudo cat /var/lib/jenkins/secrets/initialAdminPassword**

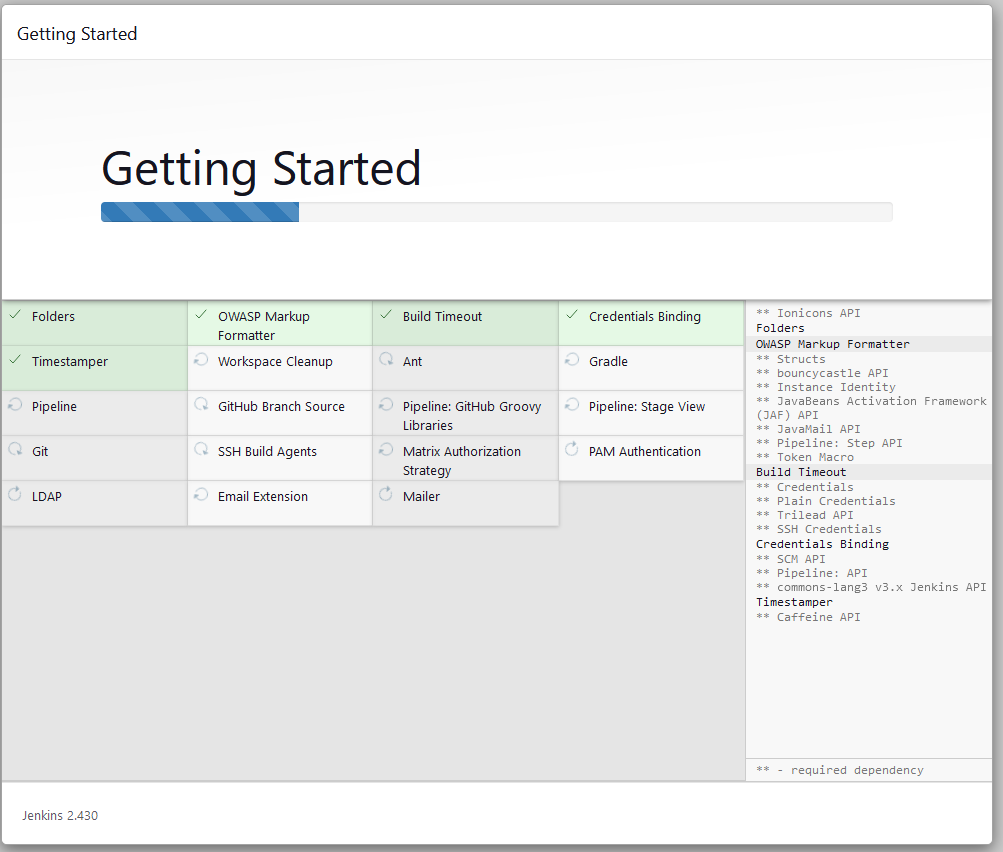
****

****

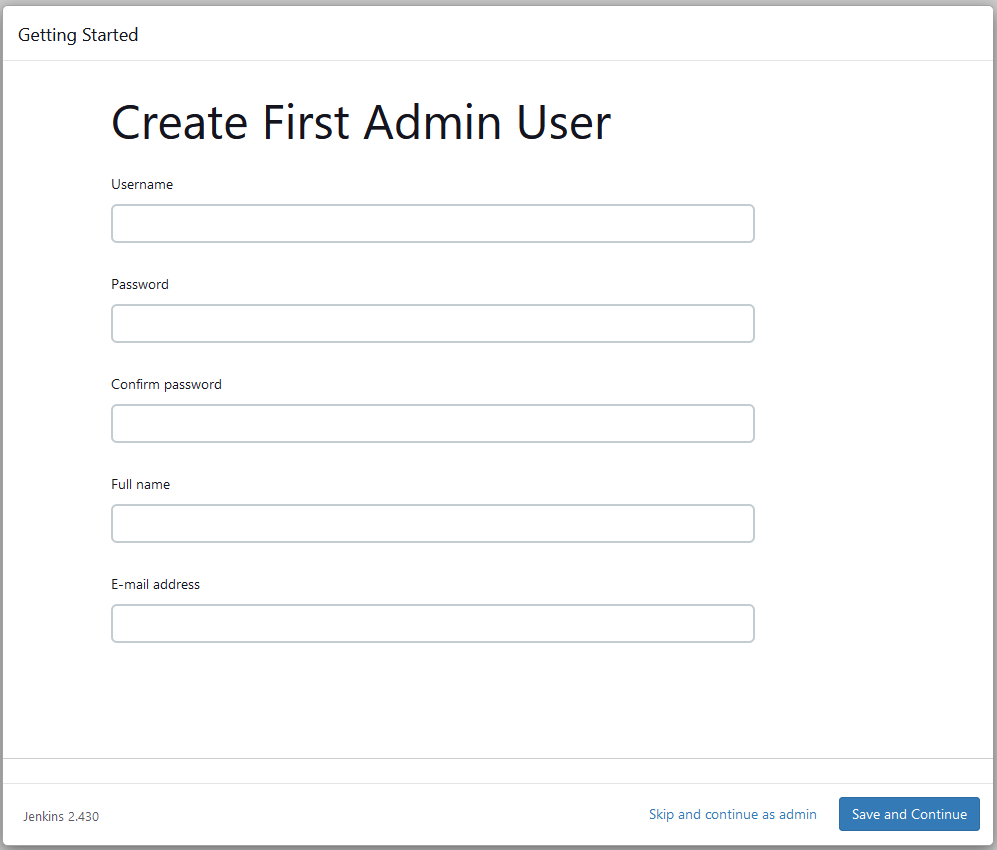
* Copy and paste the password from terminal Jenkins page next click install suggested plugins.



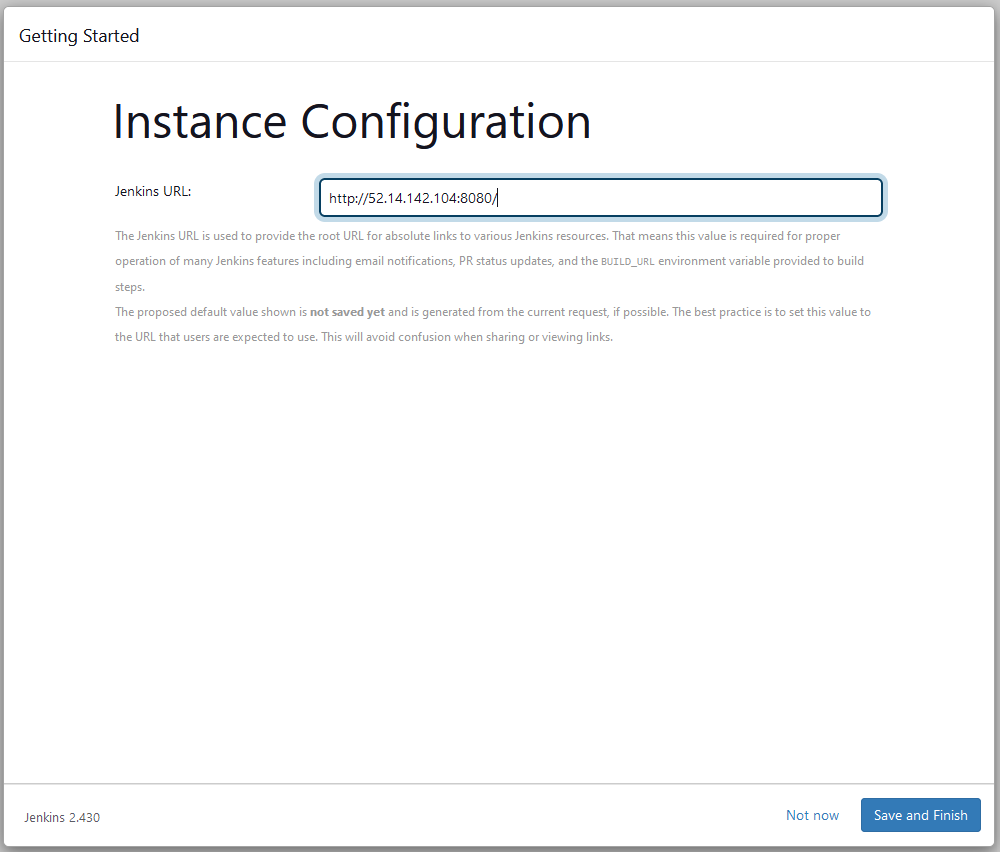
* It will install all plugins



* Create a user to work with Jenkins or continue as admin.

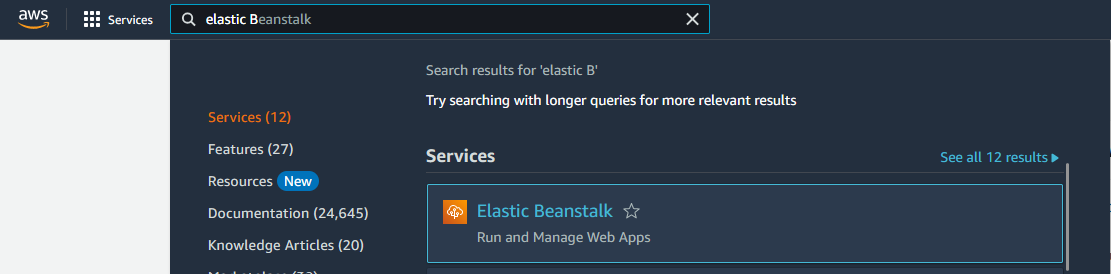


* Click save and continue to enter Jenkins dashboard.

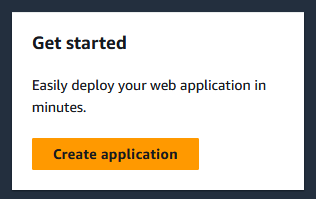


**AWS Elastic Beanstalk**

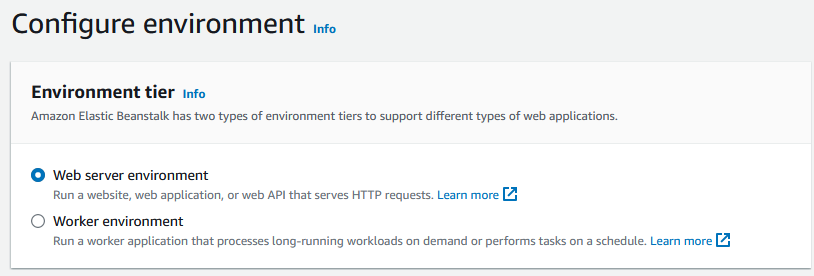
* Create and configure Elastic Beanstalk application in aws for deploy our nodejs application.
* First we need to login aws management console.
* Search and select elastic beanstalk service.

****

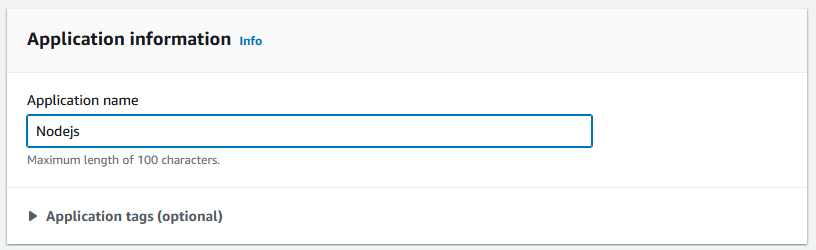
* Click Create application button

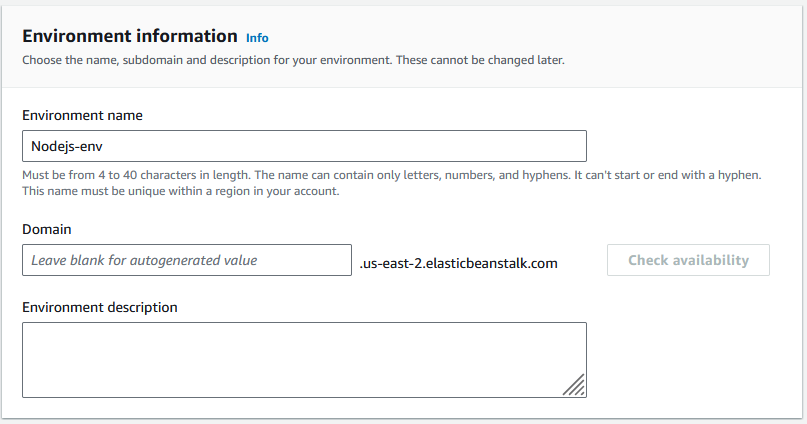


* First we need to configure our environment as a web server environment.

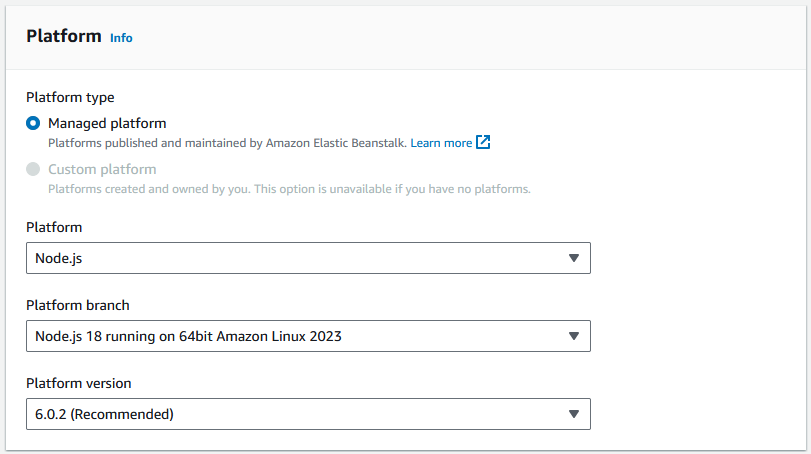


* Give name to our Application and also give environment name.

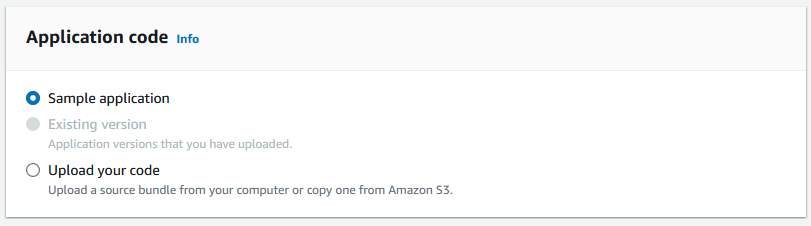




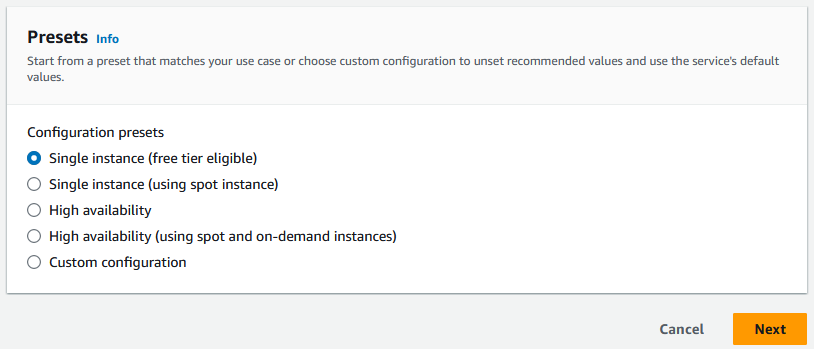
* Select Platform for our application and choose version for our application.



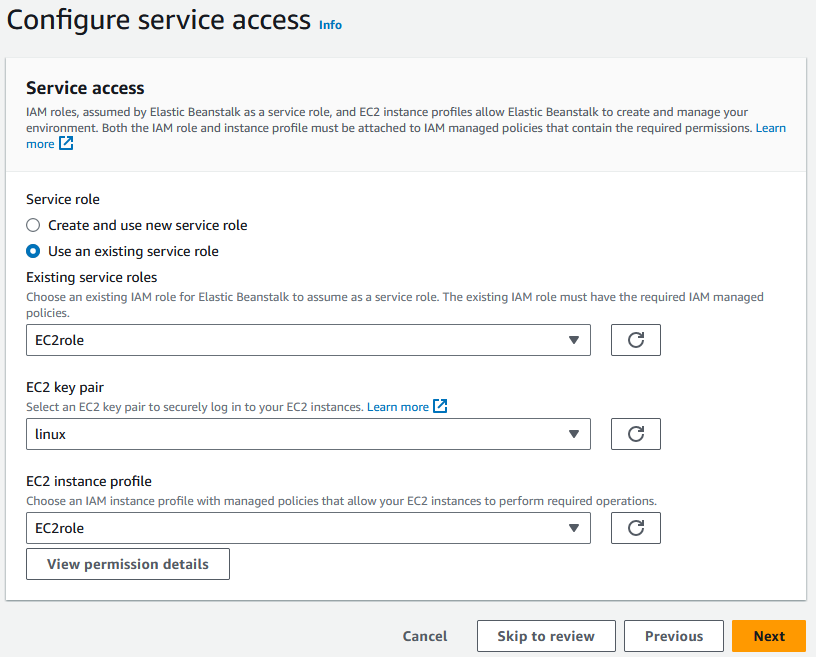
* Choose sample code for our application.



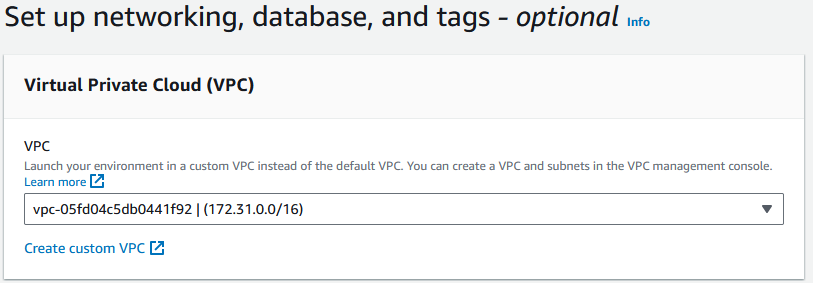
* Choose presets for our application as per needs and click next.



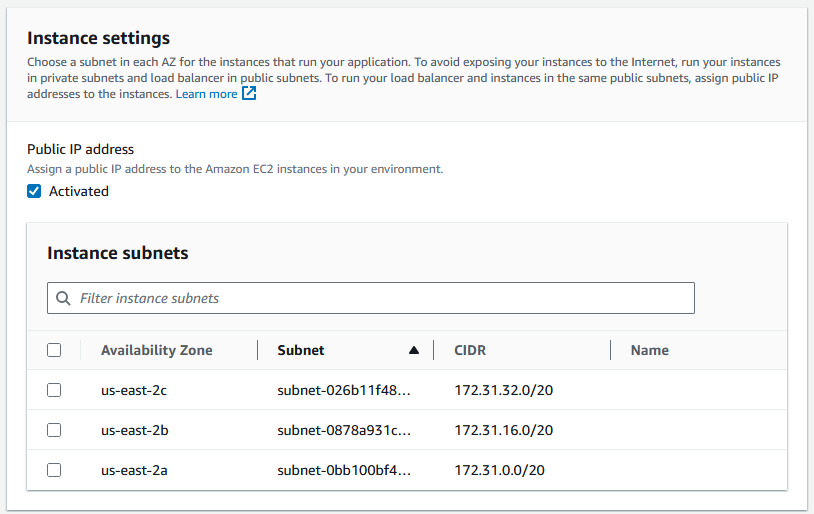
* Configure service step we need to choose our service role with permissions of **EC2, Elastic Beanstalk and S3.**
* We can attach those permissions as a role polices in IAM console.



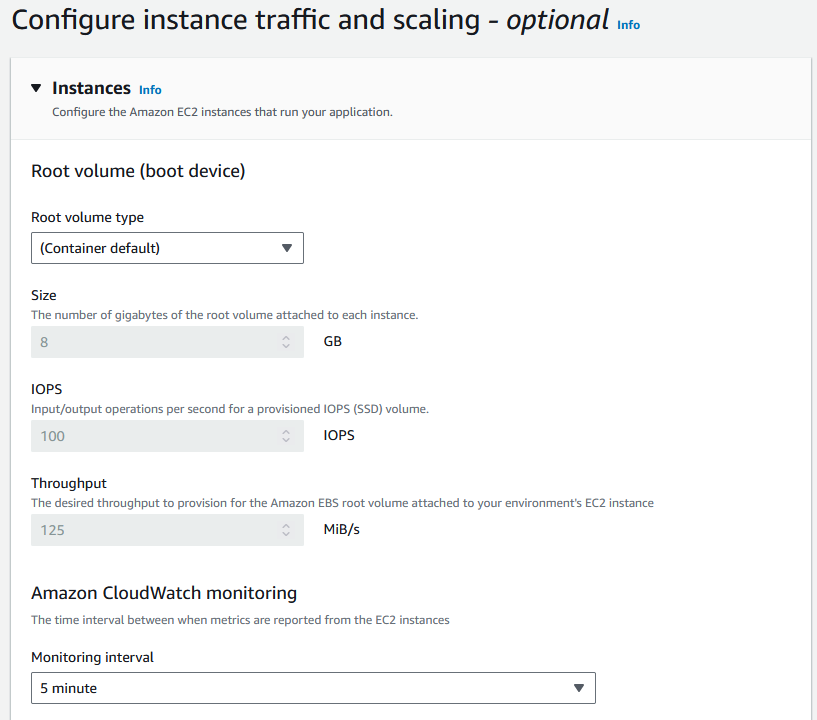
* Next we need to configure networks for this choose our VPC.



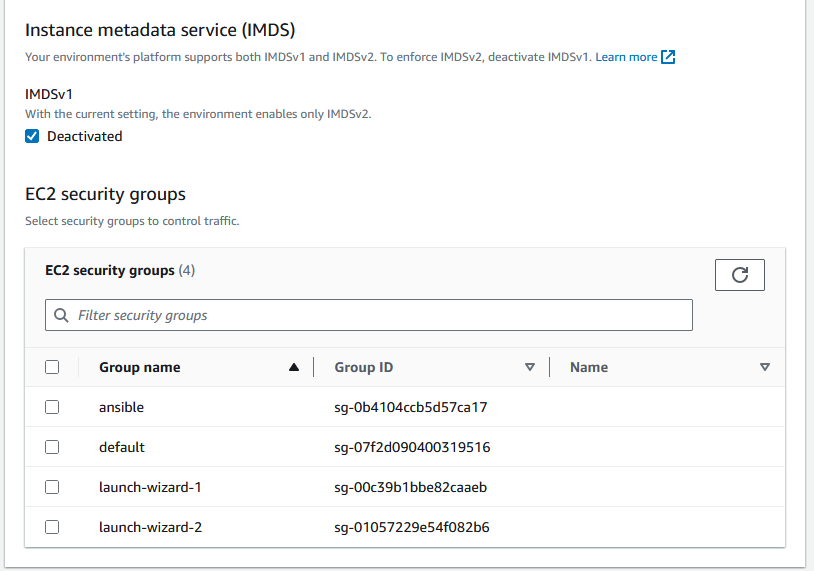
* Select public ip checkbox as well as select subnets for launch our application.



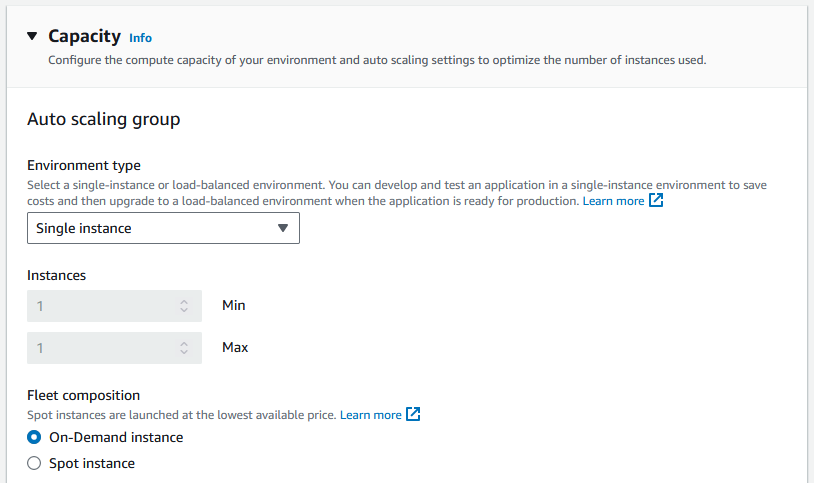
* Choose others that are optional and click next.
* Next configure instance details as go with default values.



* Choose our security group.

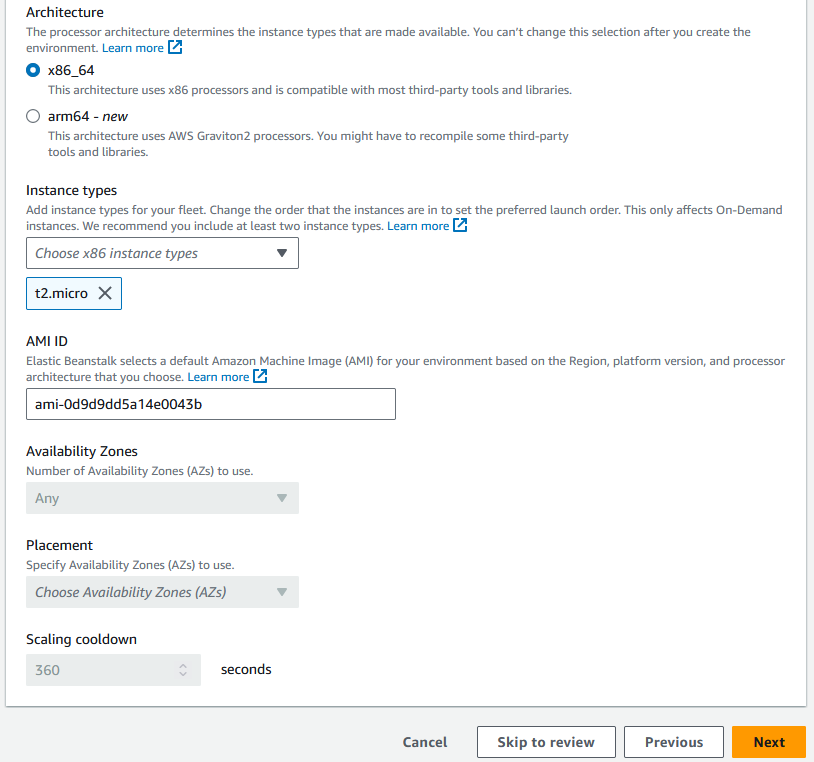


* Select our capacity for instance choose single or multiple instance.

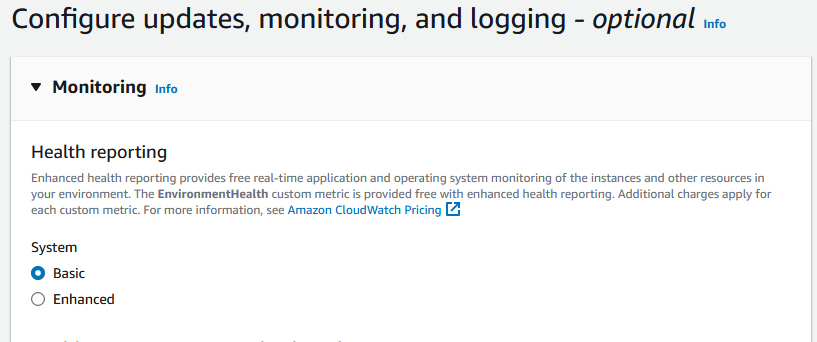


* Also choose architecture and instance type (ex: t2.micro)

and AMI ID and click next.

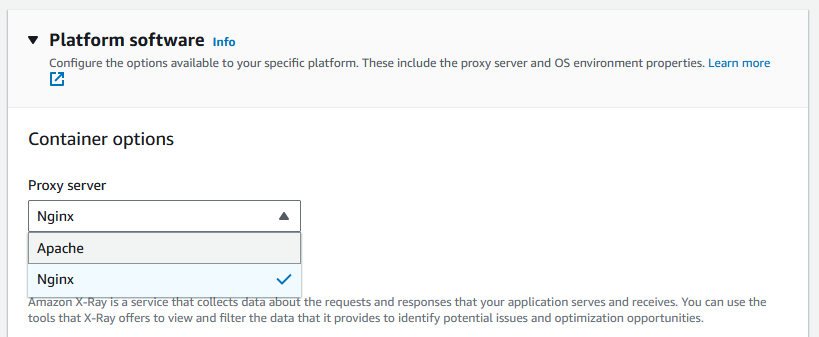


* Next configure monitoring process select basic monitoring.



* Choose our server proxy as Nginx or Apache and choose

other options as default and click next.

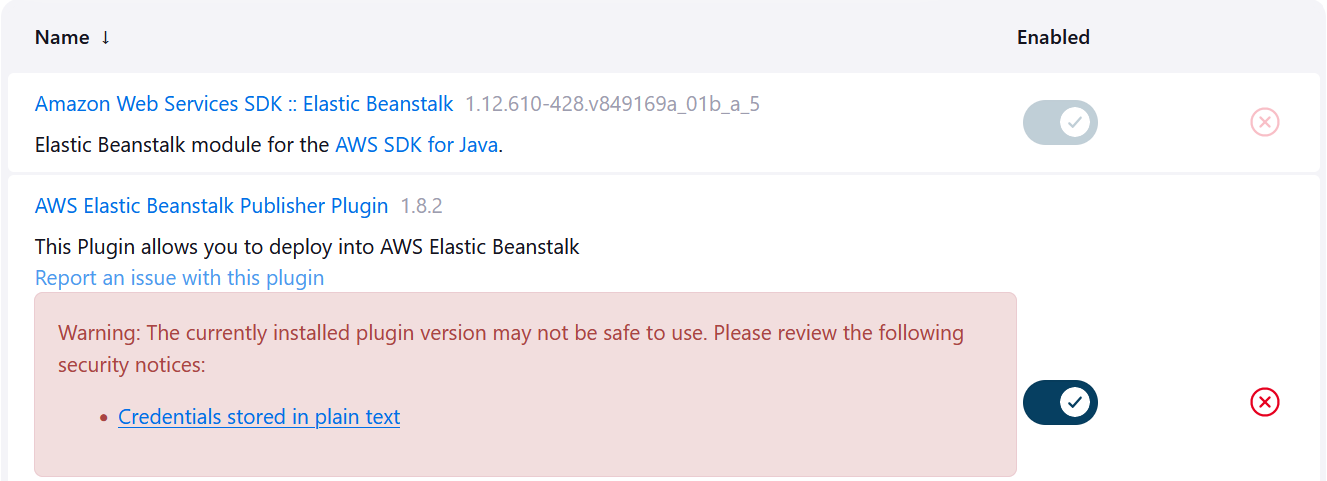


* Final step should be review and submit it will create a nodejs application on beanstalk.

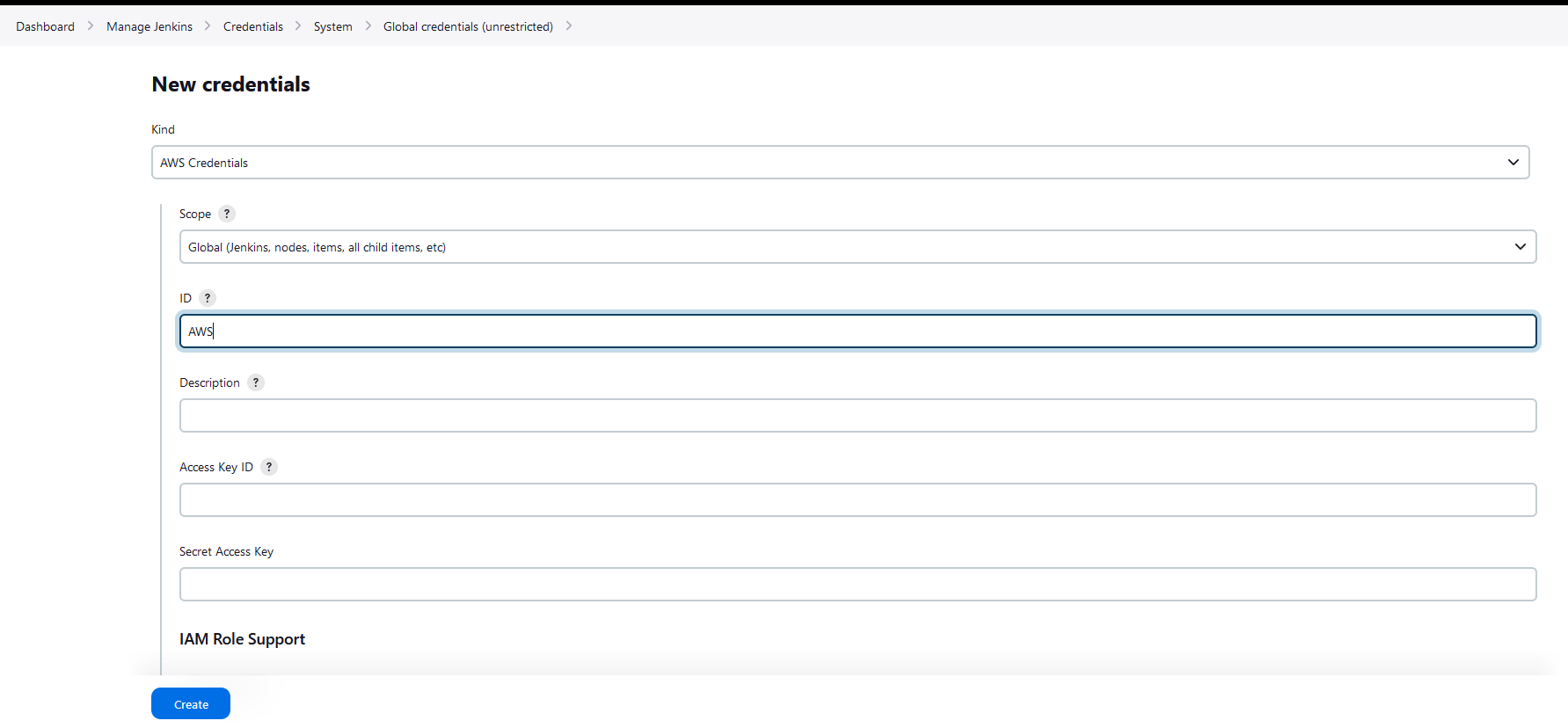
**Jenkins Configuration for Nodejs application deployment**

* First Login to Jenkins dashboard and install required plugins.
* **Manage Jenkins >> Plugins >> Available Plugins**

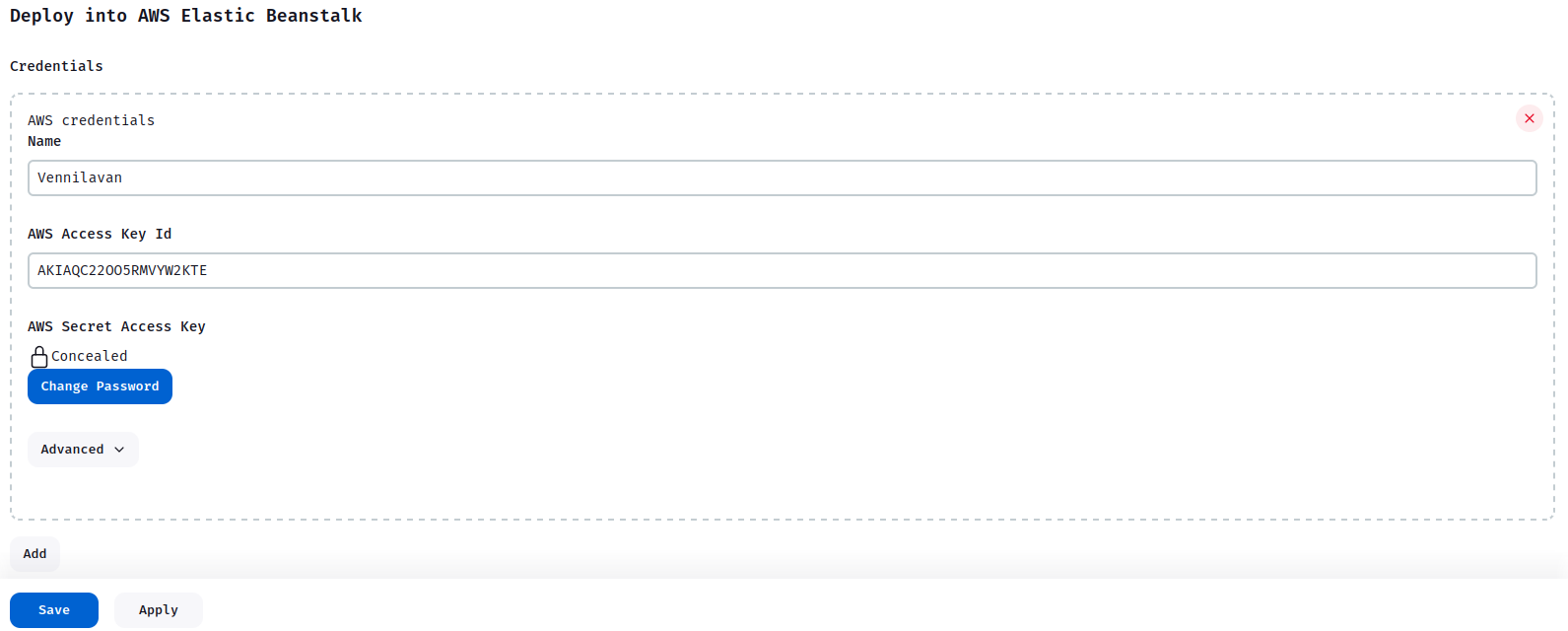
1. NodeJS Plugin
2. AWSEB Deployment Plugin
3. Publish to AWS beanstalk plugin



* Next we need to configure those plugins so follow the steps.
* Create credentials in the system and use your aws credentials and use the same ID name in jenkins file Deploy stage.



* **Manage Jenkins >> System >> Deploy into AWS Elastic Beanstalk**



* Configure Name, AWS Access Key Id and AWS Secret Access Key.
* Install node and zip in our ec2 machine using below commands.

**sudo apt install npm**

**sudo apt install zip**

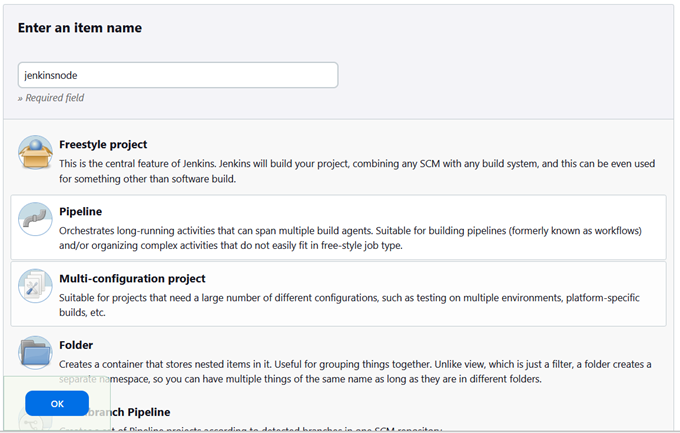
**sudo apt install awscli**

* Install aws cli and configure IAM credentials in terminal using below command.

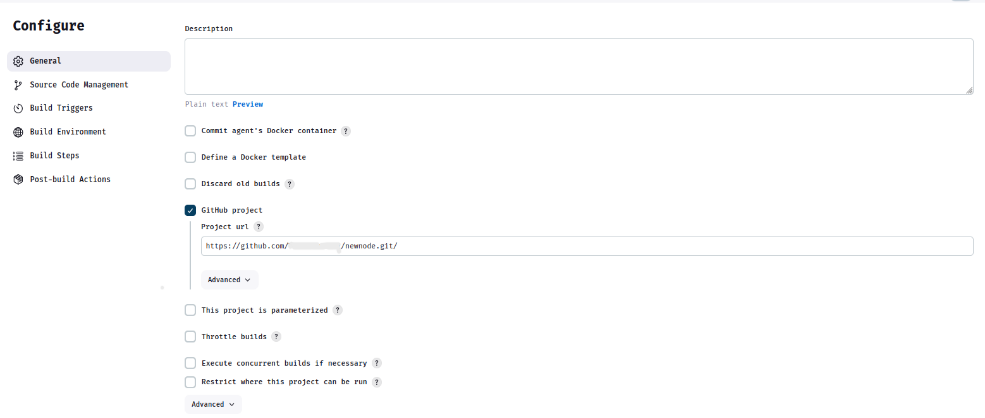
**aws configure**

**JOB Creation in Jenkins**

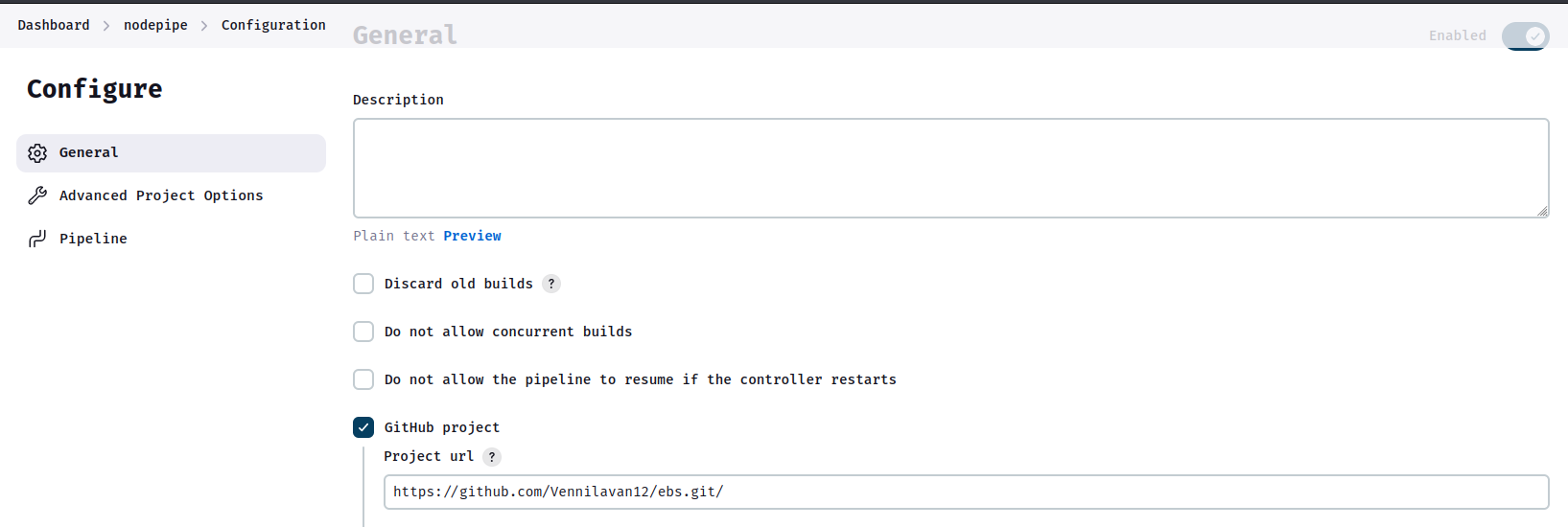
* In this project we use a **Pipeline** as a Project in Jenkins.
* Create a **Pipeline** Project and config our application details.
* **Dashboard >> New item**

****

* Give the name to the project and select Pipeline project and click ok to create it.
* Next we need to configure our project with our github code.

****

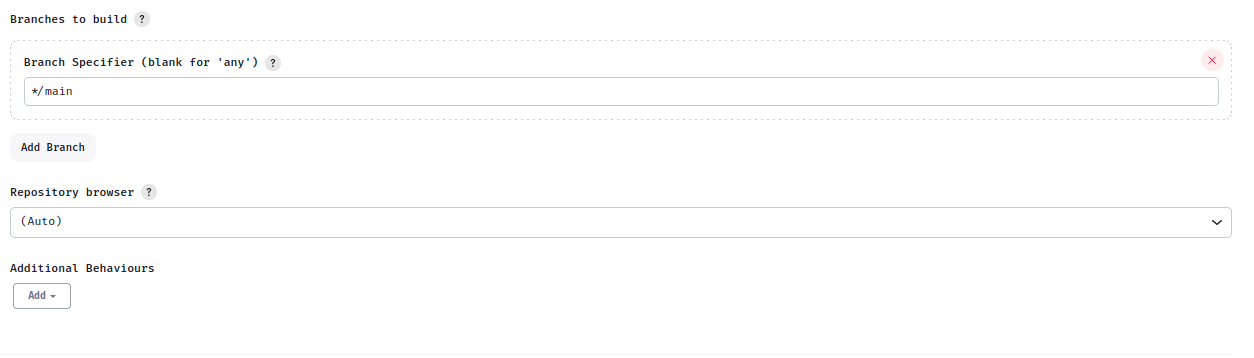
* Select Github Project and paste our github link.
* On Source code management select Git and paste the same link and select or branch as main.

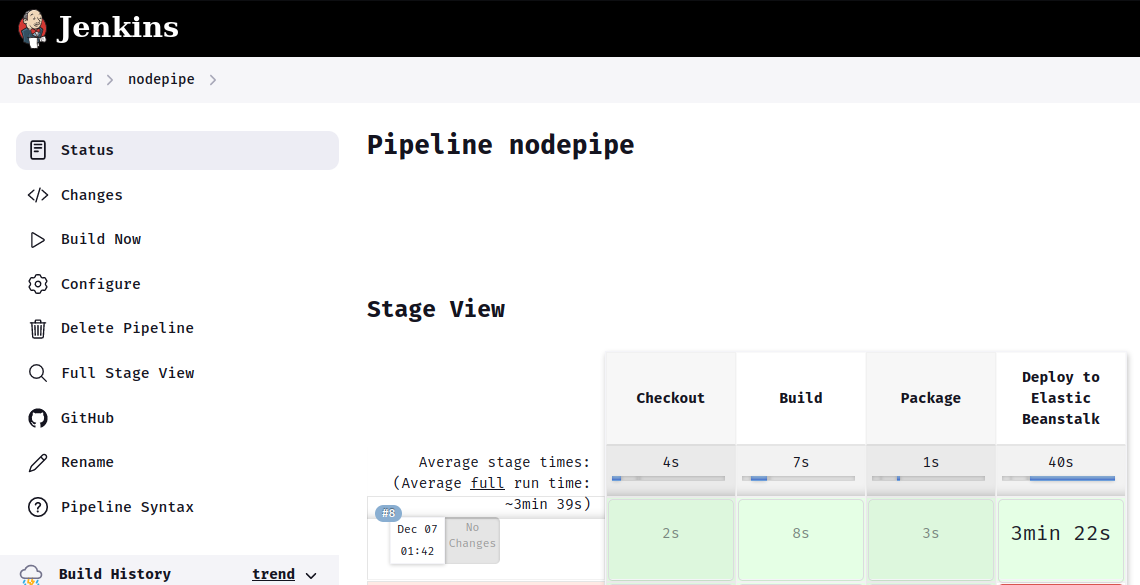


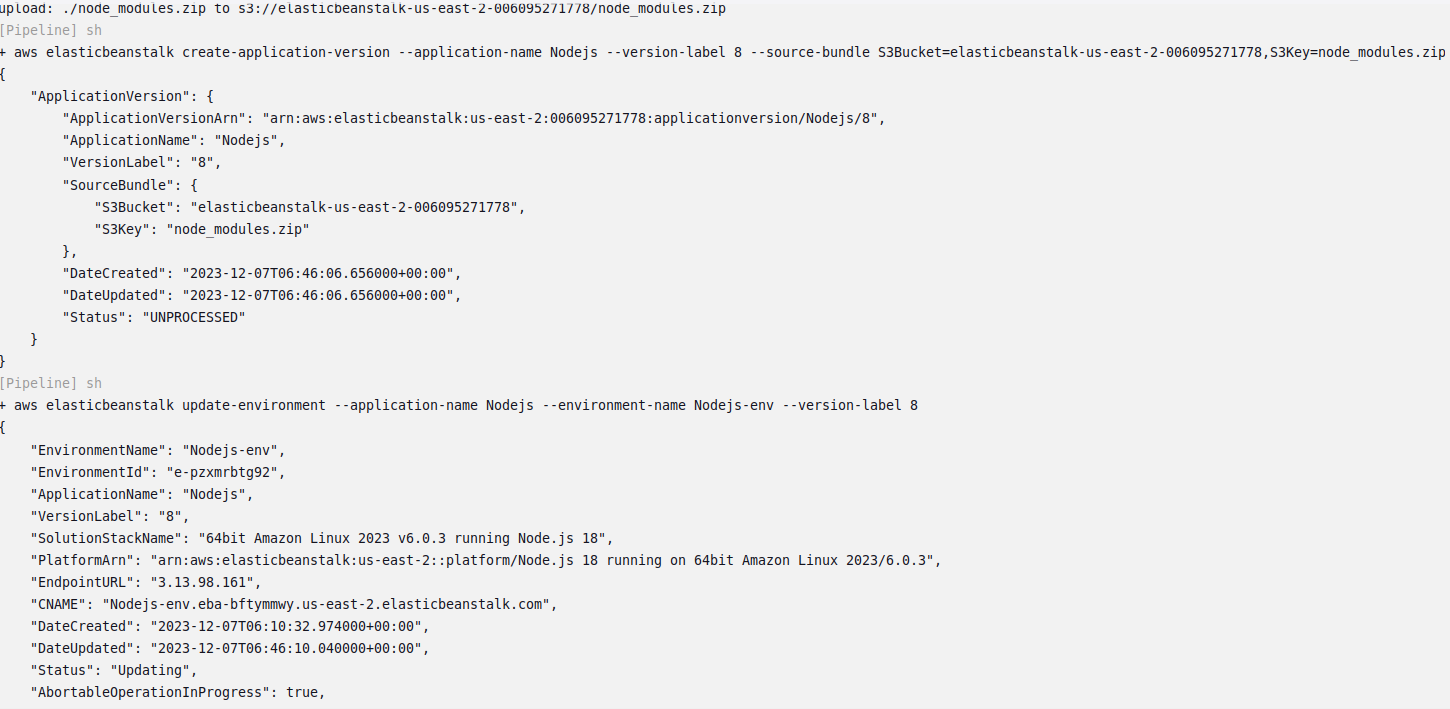
* Select our Build Triggers as GitHub hook triggers.

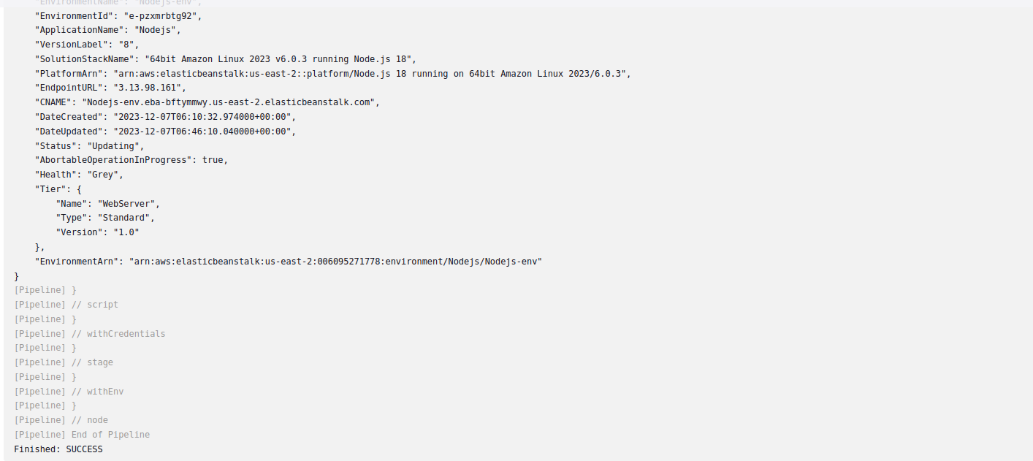


* Next we need to select pipeline script from scm and choose your repo link and Jenkinsfile.

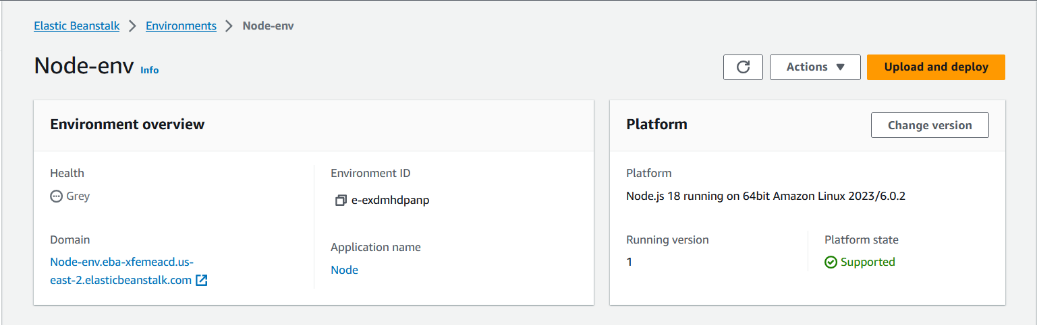


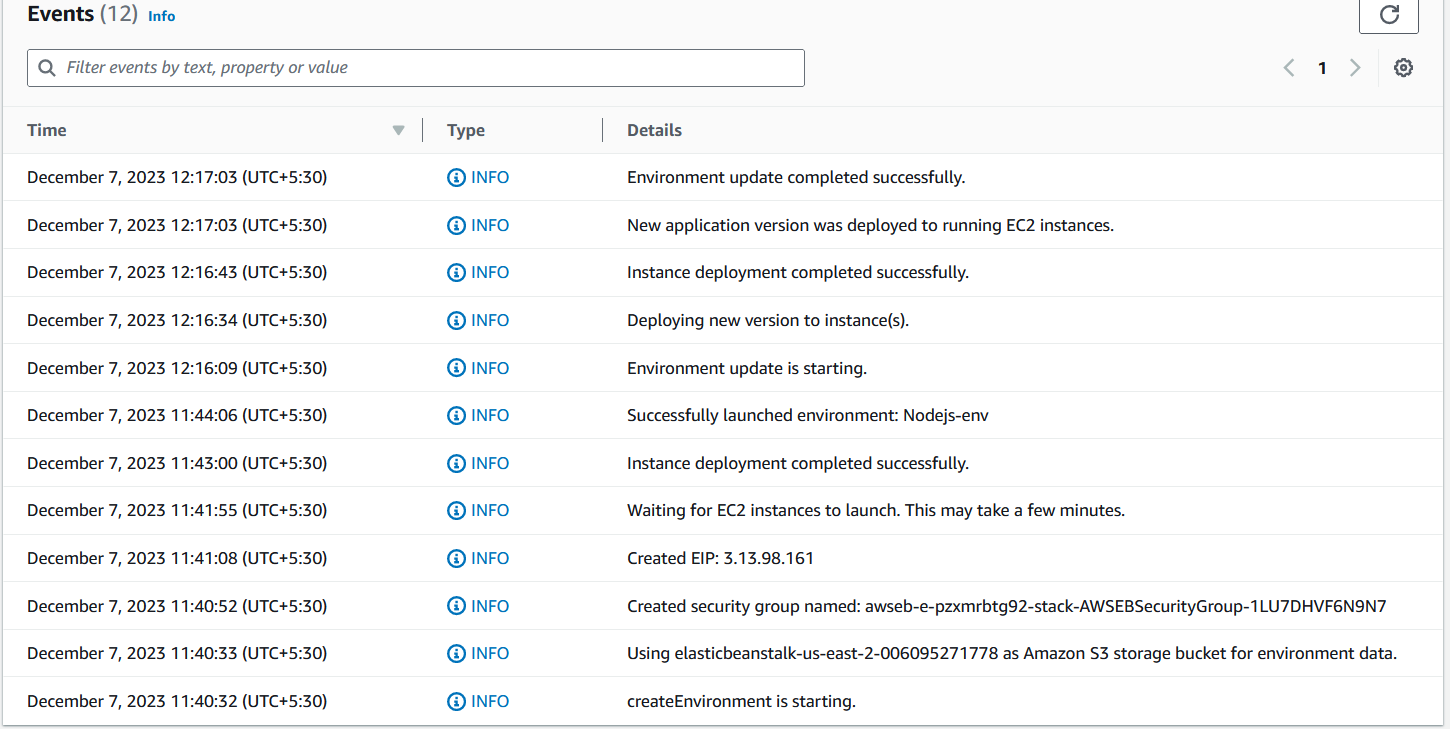




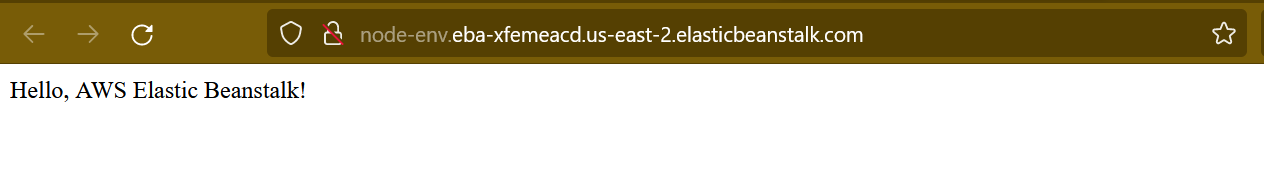


* Go to the AWS dashboard and check Elastic Beanstalk environment.

****

****

* Check the Beanstalk environment to reach our website.

****

Reference :Github url: <https://github.com/Vennilavan12/ebs.git>