Continuous Integration for a Node.js Application using Git, Jenkins, and AWS Elastic Beanstalk

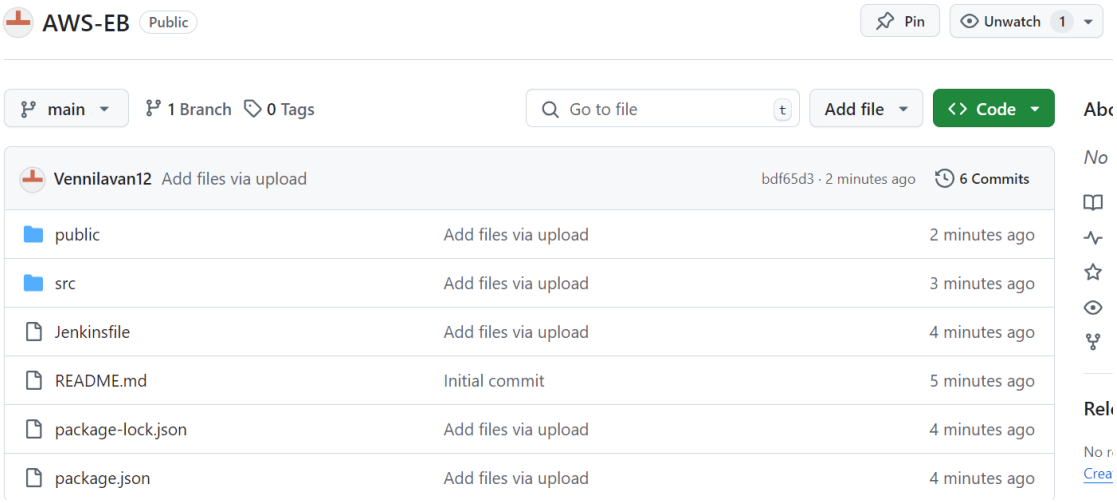
Solution

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

- Create a new repository into our github account.



- Clone the repository into local using the git clone method.

  git clone "Repo url"

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

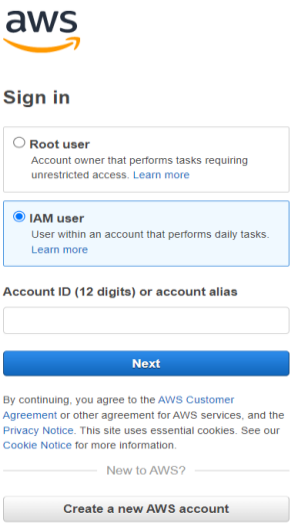
STEP 2: **Configure Jenkins to clone the Git repository, install dependencies, run tests, and package the application as a ZIP file**

Requirements :

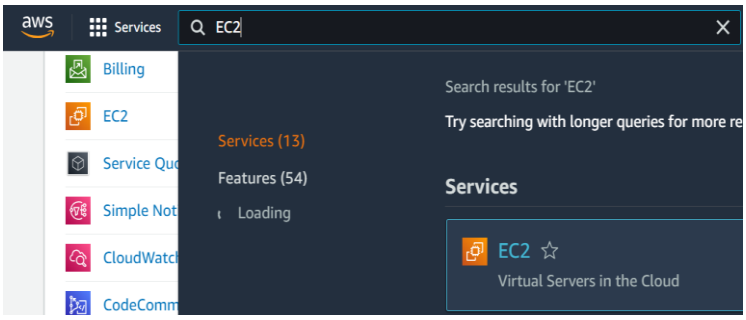
1. EC2
2. Jenkins

EC2

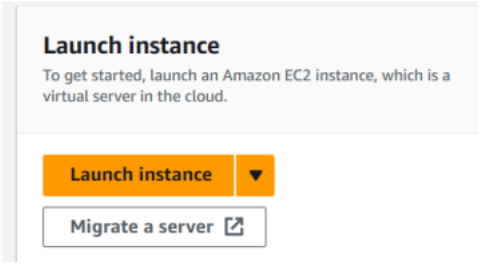
- Create an EC2 instance in AWS for installing Jenkins server.
- Login to AWS Management console



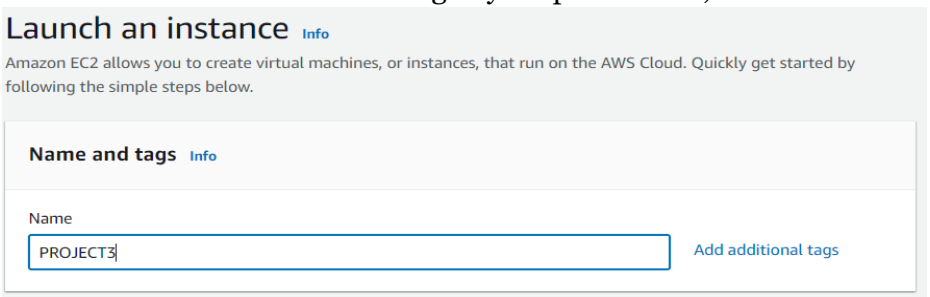
- Choose EC2 and click launch instance



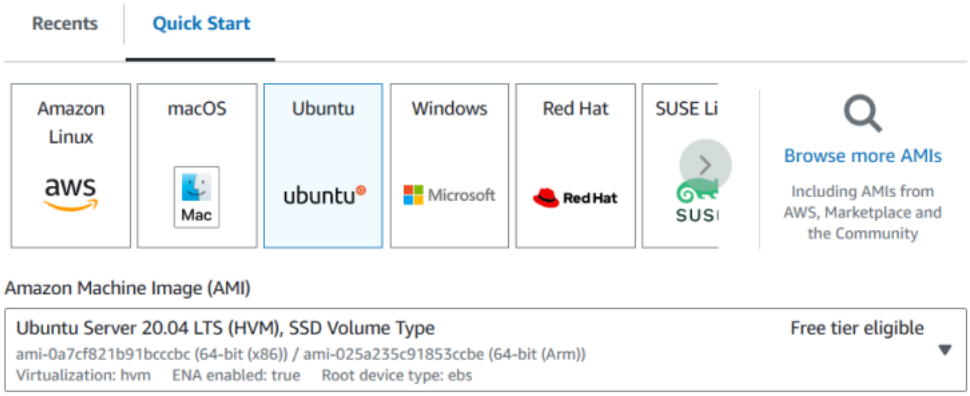
- Then under the EC2 management console we could able to find the Launch instance option, click that one for creating an instance:



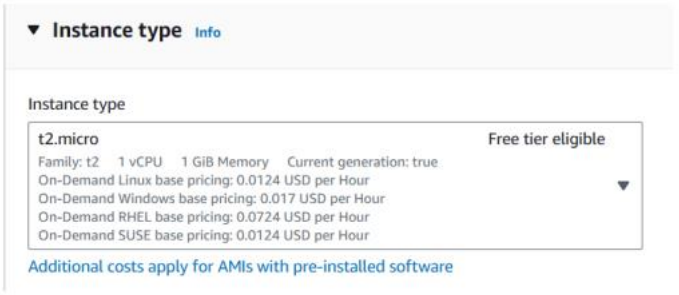
- Then name the instance according to your preferences,



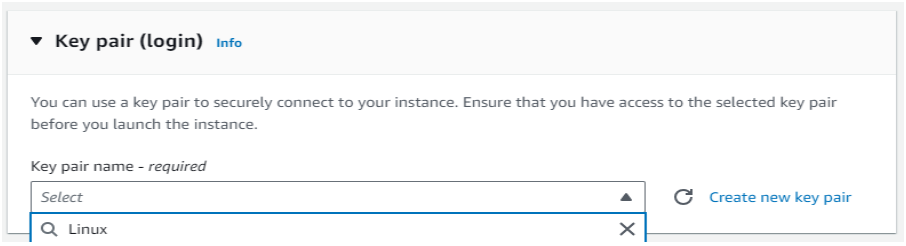
- Then we need to select an AMI Image for the instance, choose AMI image according to your preferences: Here I am selecting Ubuntu 20.04 AMI Image.



- Then we need to select the instance type

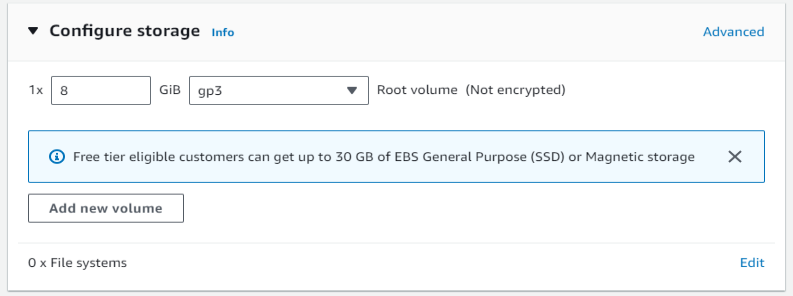


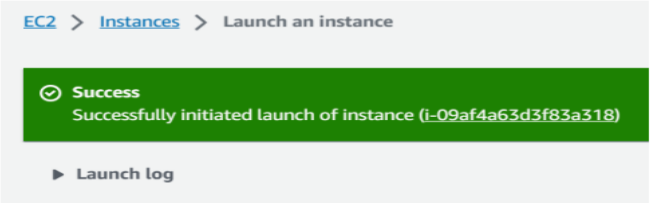
- Then we need to select the keypair for security authentication purpose of the instance



- 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 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

```
ubuntu@ip-172-31-19-109:~$ sudo apt update
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1149 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [245 kB]
```

sudo apt install fontconfig openjdk-17-jre

```
ubuntu@ip-172-31-19-109:~$ sudo apt install fontconfig openjdk-17-jre
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
fontconfig is already the newest version (2.13.1-4.2ubuntu5).
fontconfig set to manually installed.
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2 libasound2-data libatk-wr
  openjdk-17-jre-headless
Suggested packages:
  default-jre libasound2-plugins alsa-utils pcscd libnss-mdns fonts-ipafont-gothic fonts-ipafont-mincho fonts-wqy-microhei
The following NEW packages will be installed:
  alsa-topology-conf alsa-ucm-conf ca-certificates-java fonts-dejavu-extra java-common libasound2 libasound2-data libatk-wr
  openjdk-17-jre openjdk-17-jre-headless
0 upgraded, 13 newly installed, 0 to remove and 21 not upgraded.
Need to get 51.2 MB of archives.
After this operation, 204 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 alsa-topology-conf all 1.2.5.1-2 [15.5 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2-data all 1.2.6.1-1ubuntu1 [19.1 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libasound2 amd64 1.2.6.1-1ubuntu1 [390 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 alsa-ucm-conf all 1.2.6.3-1ubuntu1.8 [43.3 kB
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 java-common all 0.72build2 [6782 B]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libpcsclite1 amd64 1.9.5-3ubuntu1 [19.8 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre-headless amd64 17.0.8.1+1~
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 ca-certificates-java all 20190909ubuntu1.2 [1
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 fonts-dejavu-extra all 2.37-2build1 [2041 kB]
Get:10 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libatk-wrapper-java all 0.38.0-5build1 [53.1 kB]
Progress: [ 54%] [#################################################################################################################
Get:12 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libgif7 amd64 5.1.9-2build2 [33.8 kB]
Get:13 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 openjdk-17-jre amd64 17.0.8.1+1~us1-0ubu
```

java –version

```
ubuntu@ip-172-31-19-109:~$ java -version
openjdk version "17.0.8.1" 2023-08-24
OpenJDK Runtime Environment (build 17.0.8.1+1-Ubuntu-0ubuntu122.04)
OpenJDK 64-Bit Server VM (build 17.0.8.1+1-Ubuntu-0ubuntu122.04, mixed mode, sharing)
ubuntu@ip-172-31-19-109:~$
```

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

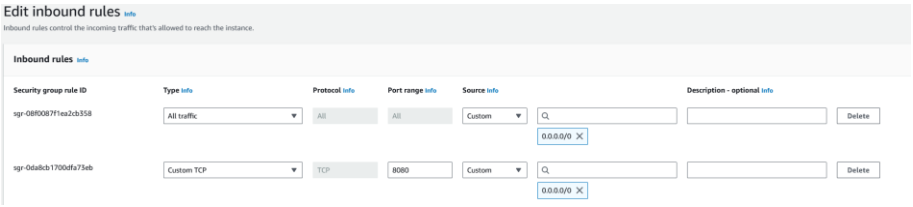
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \https://pkg.jenkins.io/debian/jenkins.io-2023.key
echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \ https://pkg.jenkins.io/debian binary/ | sudo tee \ /etc/apt/sources.list.d/jenkins.list > /dev/null

```
ubuntu@ip-172-31-19-109:~$ 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
--2023-11-04 11:41:37--  https://pkg.jenkins.io/debian/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.78.133, 2a04:4e42:83::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.78.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc         100%[=============================================================================>]

2023-11-04 11:41:37 (25.0 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]
```

sudo apt-get update
sudo apt-get install Jenkins

```
sudo apt-get install jenkins
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Ign:5 https://pkg.jenkins.io/debian binary/ InRelease
Get:6 https://pkg.jenkins.io/debian binary/ Release [2044 B]
Get:7 https://pkg.jenkins.io/debian binary/ Release.gpg [833 B]
Get:8 https://pkg.jenkins.io/debian binary/ Packages [57.4 kB]
Fetched 60.3 kB in 1s (64.7 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 21 not upgraded.
Need to get 89.2 MB of archives.
After this operation, 90.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian binary/ jenkins 2.430 [89.0 MB]
Fetched 89.2 MB in 5s (16.5 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 124560 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../archives/jenkins_2.430_all.deb ...
Unpacking jenkins (2.430) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.430) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
```

- 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.

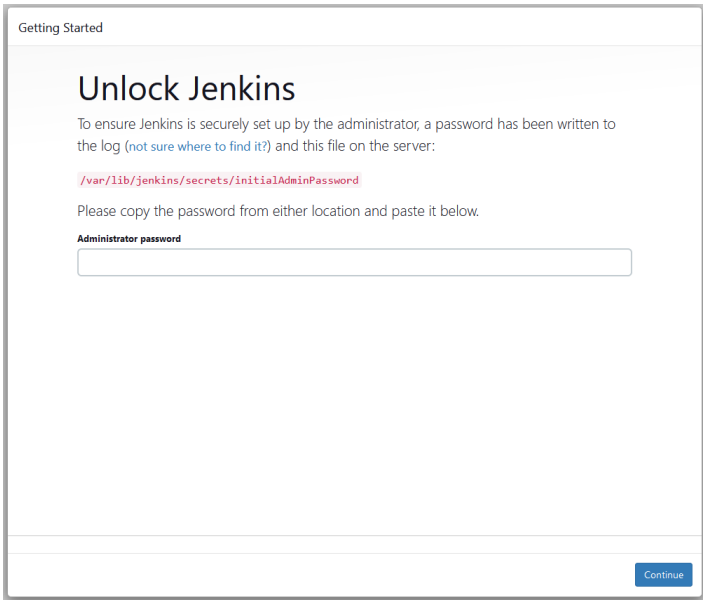


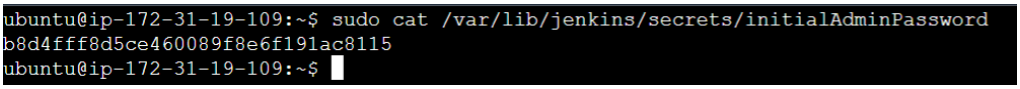
- Next Accessing our Jenkins server using our instance public ip follows with port 8080.

http://IP:8080

- Getting started, the Jenkins page will open and paste our admin password.
- Use this command to get Jenkins admin password

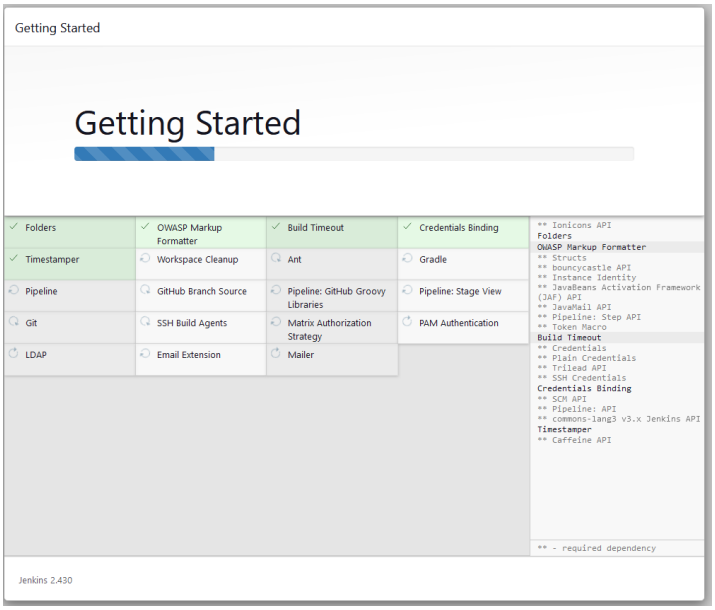
sudo cat /var/lib/jenkins/secrets/initialAdminPassword



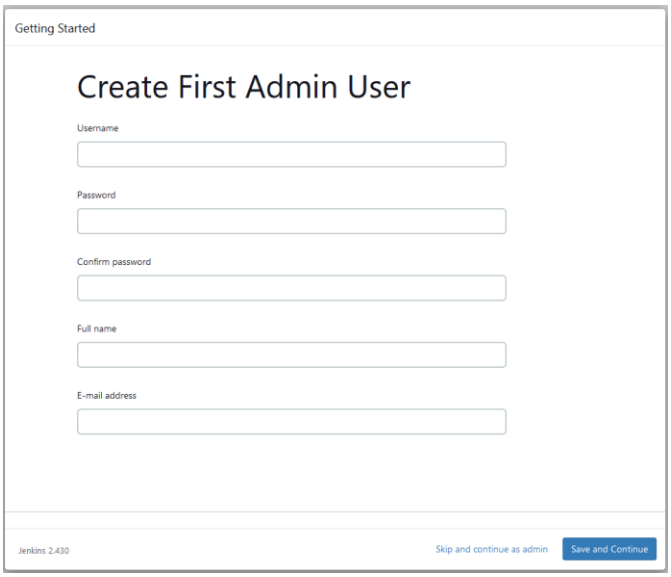


- Copy and paste the password from terminal Jenkins page next click install suggested plugins.

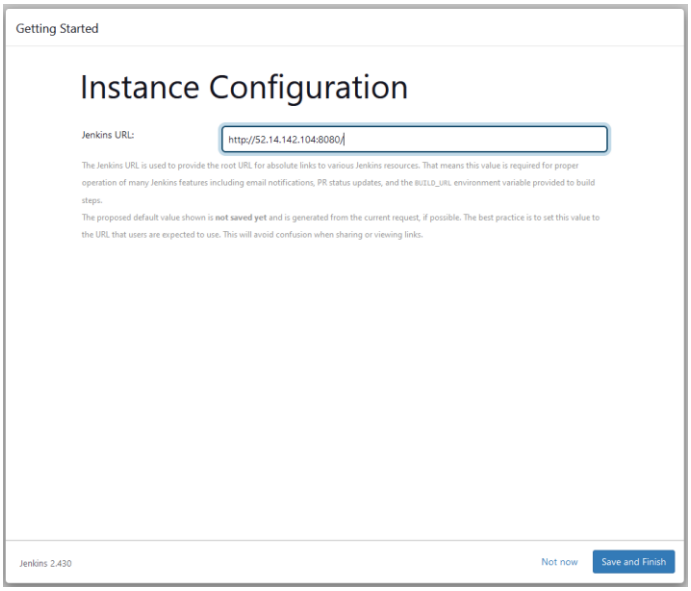
- It will install all plugins



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

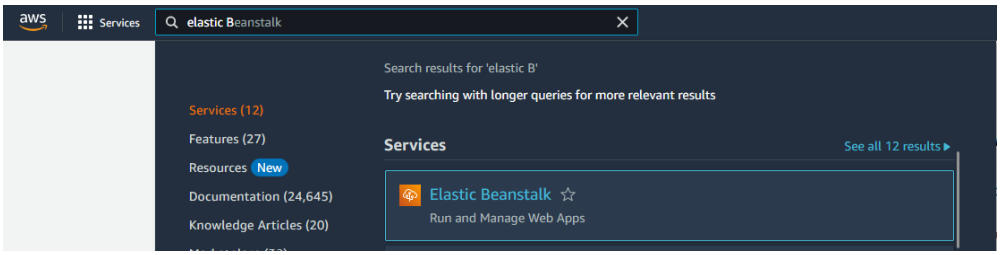


- Click save and continue to enter Jenkins dashboard.

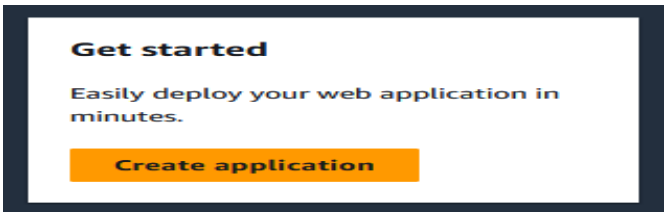


AWS Elastic Beanstalk

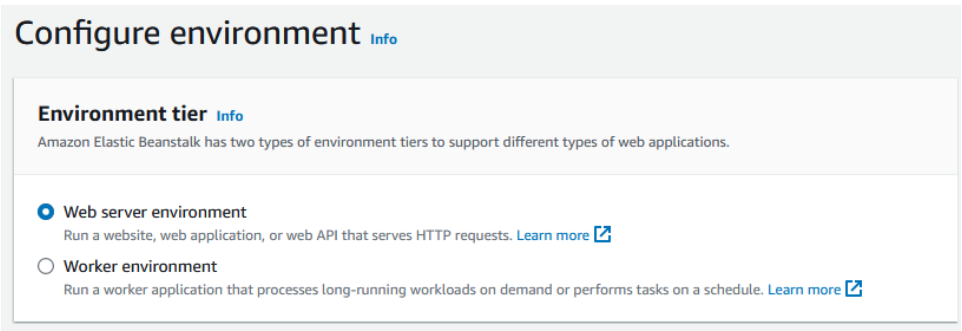
- Create and configure Elastic Beanstalk application in aws to deploy our nodejs application.
- First we need to login aws management console.
- Search and select elastic beanstalk service.



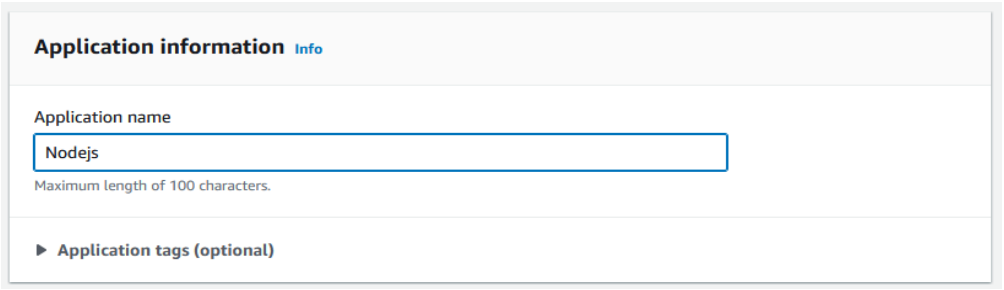
- Click Create application button

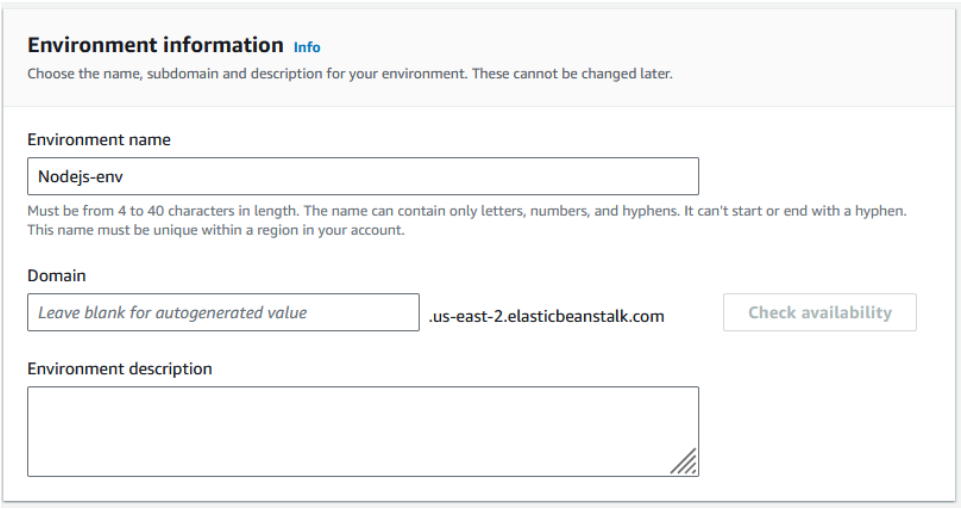


- First we need to configure our environment as a web server environment.

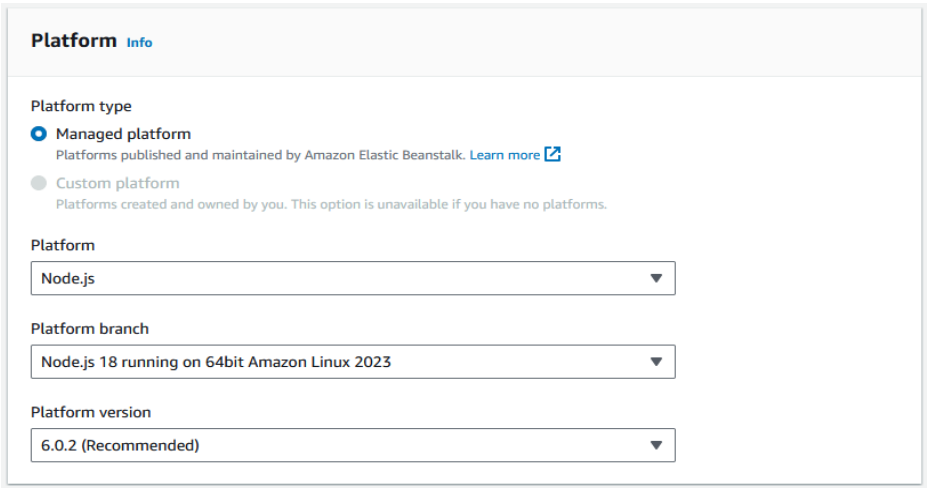


- Give name to our Application and also give environment name.

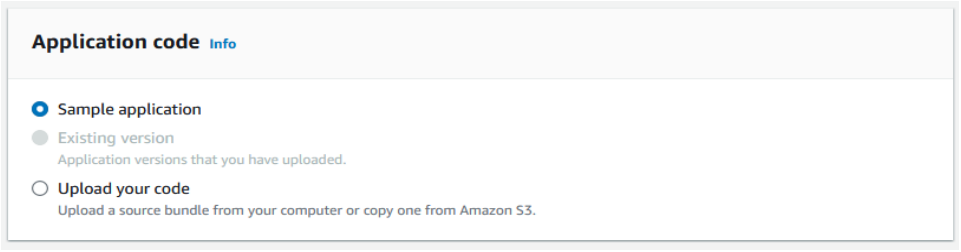




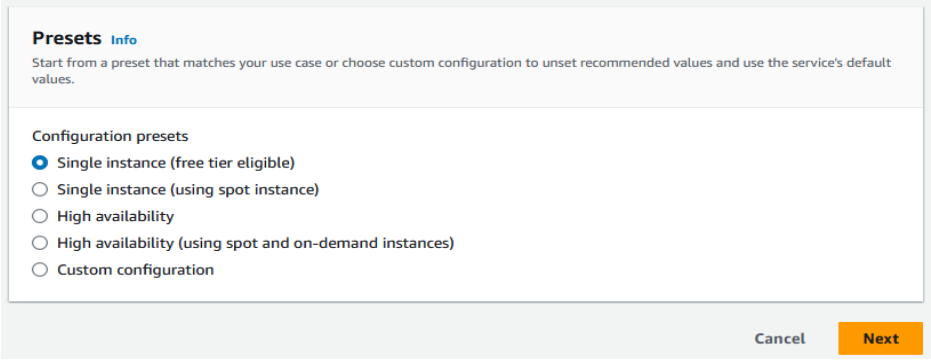
- Select Platform for our application and choose version for our application.



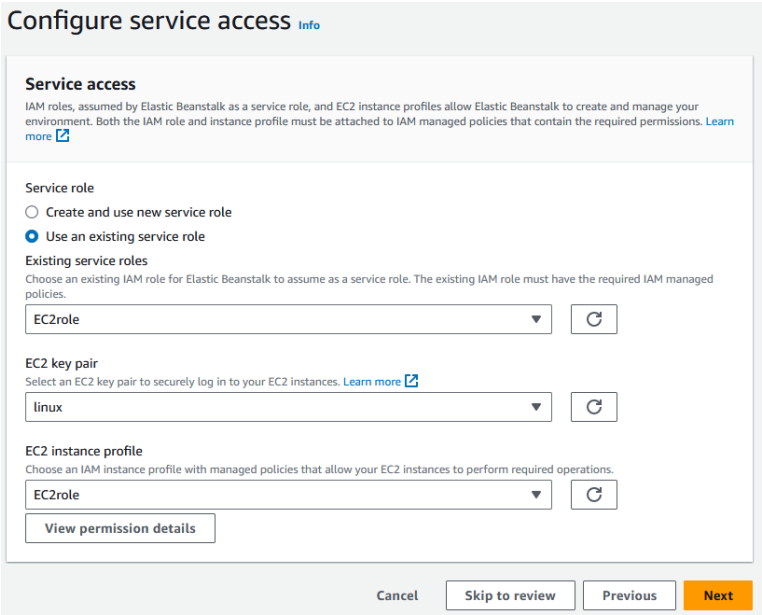
- Choose sample code for our application.



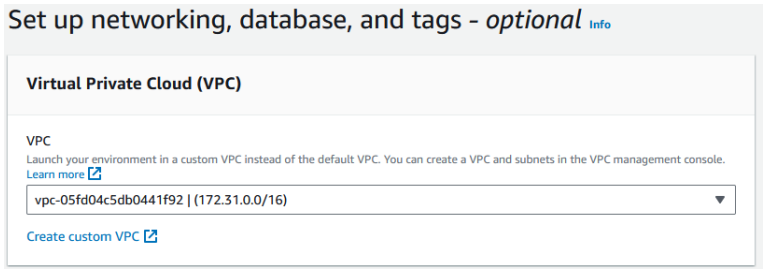
- Choose presets for our application as per needs and click next.



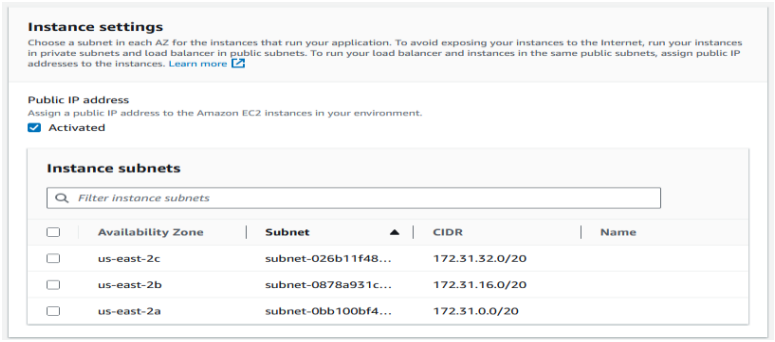
- Configure service step we need to choose our service role with permissions of **EC2, Elastic Beanstalk and S3.**
- We can attach those permissions as a role polices in IAM console.



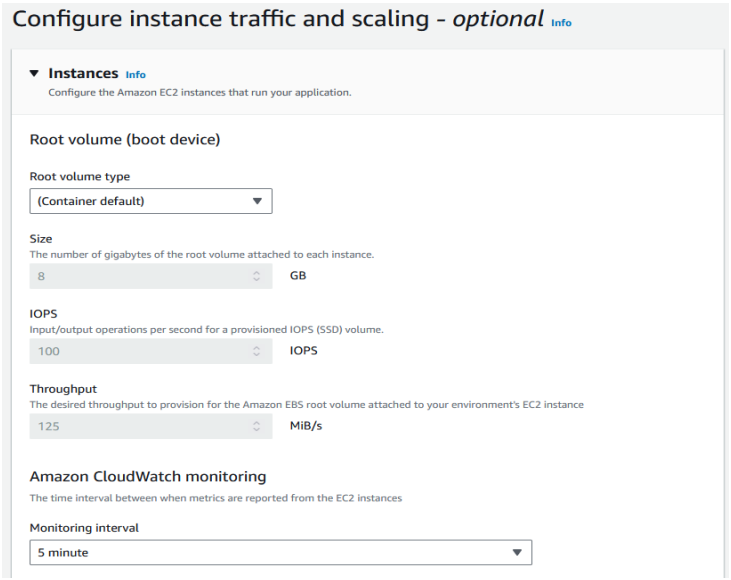
- Next we need to configure networks for this to choose our VPC.



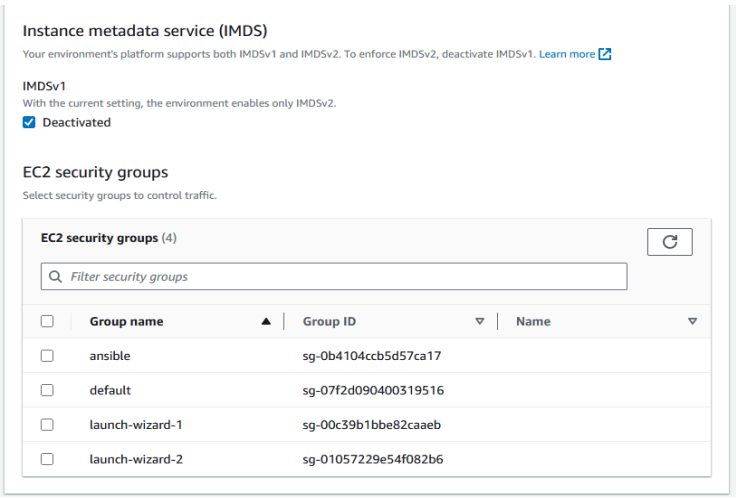
- Select public ip checkbox as well as select subnets for launch our application.



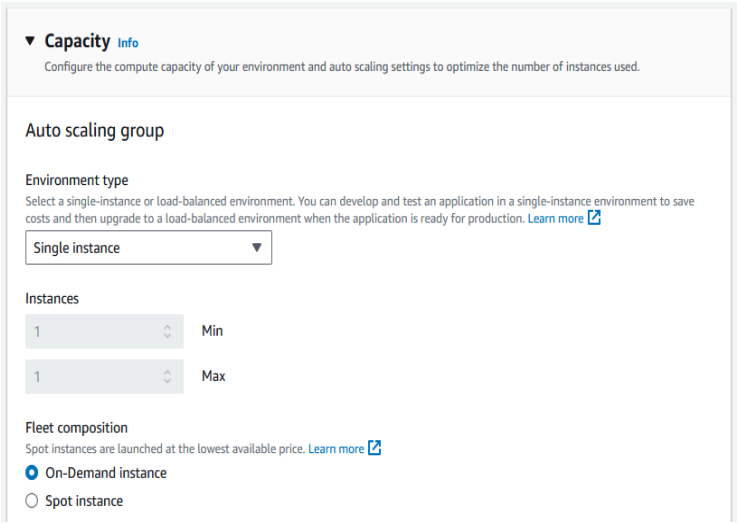
- Choose others that are optional and click next.
- Next configure instance details as go with default values.



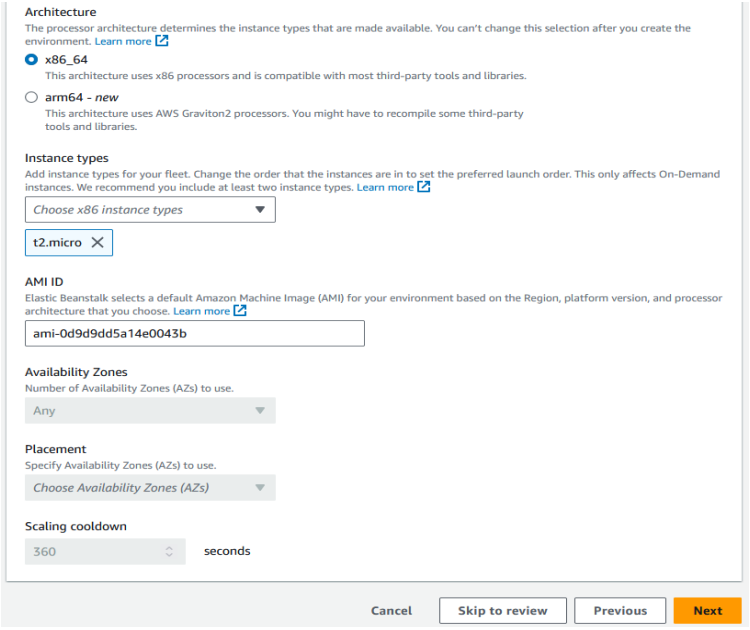
- Choose our security group.



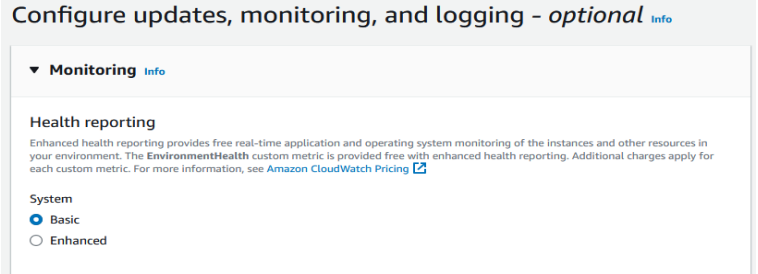
- Select our capacity for instance choose single or multiple instance.



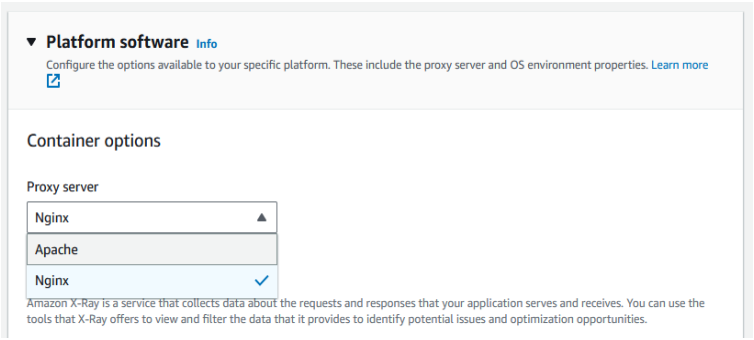
- Also choose architecture and instance type (ex: t2.micro) and AMI ID and click next.



- Next, configure the monitoring process and select basic monitoring.



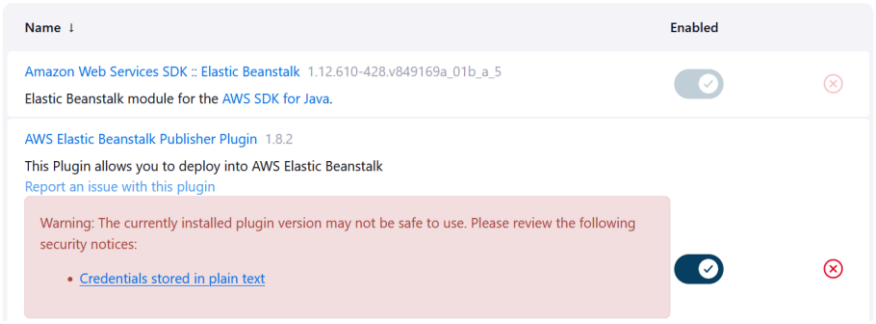
- Choose our server proxy as Nginx or Apache and choose other options as default and click next.



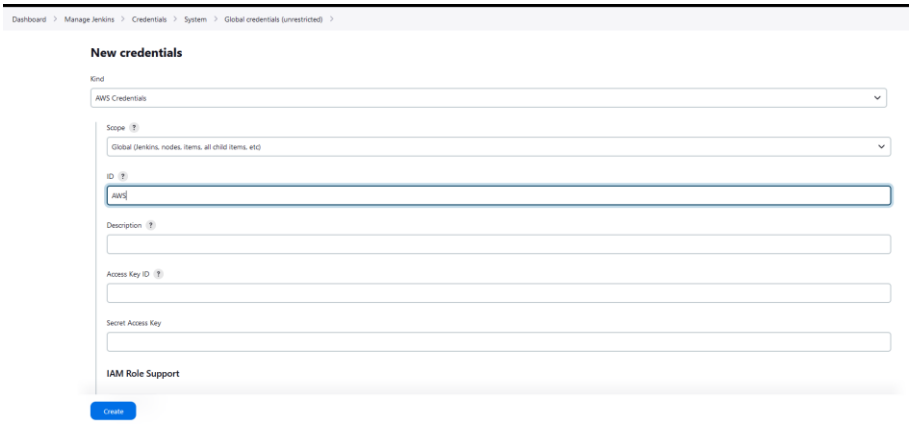
- Final step should be review and submit. It will create a nodejs application on beanstalk.

Jenkins Configuration for Nodejs application deployment

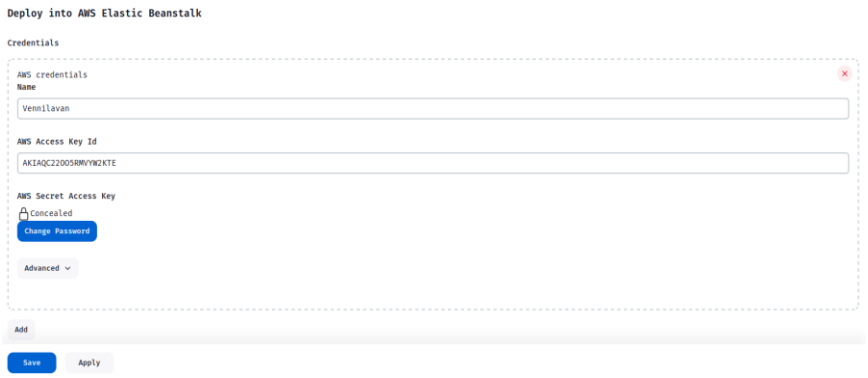
- First Login to Jenkins dashboard and install required plugins.
- **Manage Jenkins >> Plugins >> Available Plugins**
    1. NodeJS Plugin
    2. AWSEB Deployment Plugin
    3. Publish to AWS beanstalk plugin



- 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

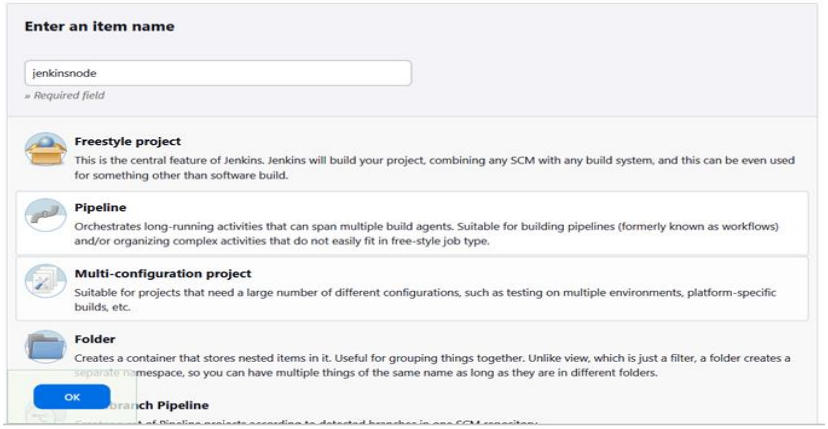
  sudo apt install awscli

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

  aws configure

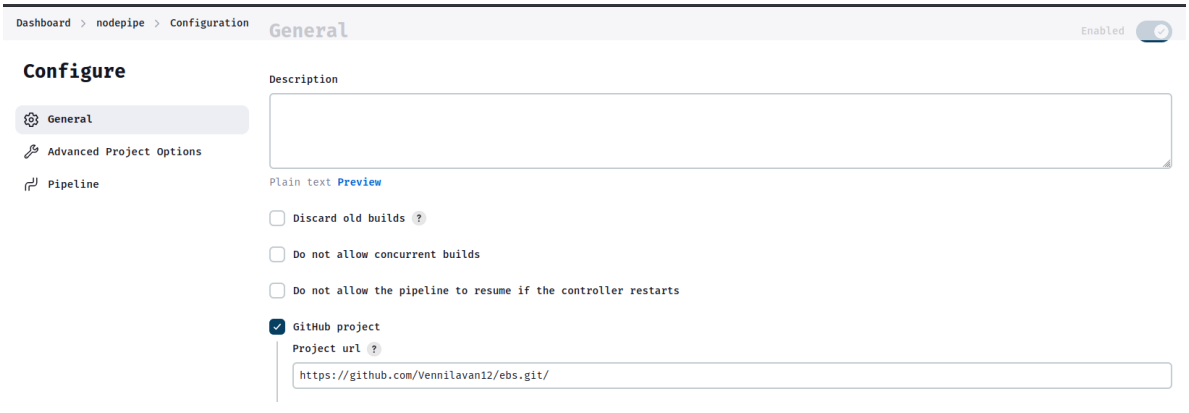
JOB Creation in Jenkins

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



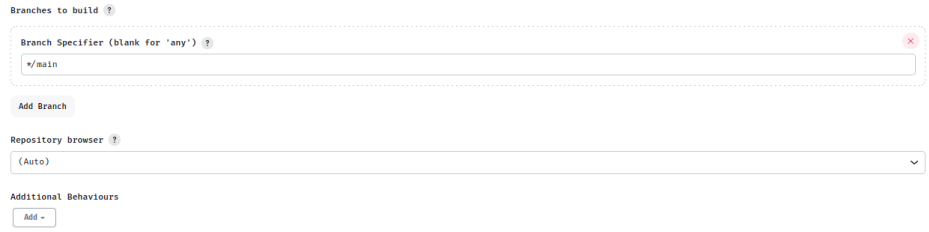
- Give the name to the project and select Pipeline project and click ok to create it.
- Next we need to configure our project with our github code.

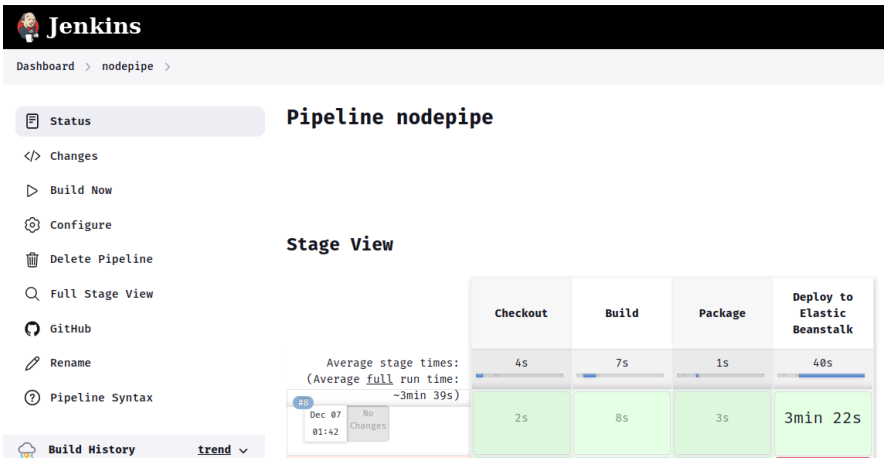
- Select Github Project and paste our github link.
- On Source code management select Git and paste the same link and select or branch as main.



- Select our Build Triggers as GitHub hook triggers.

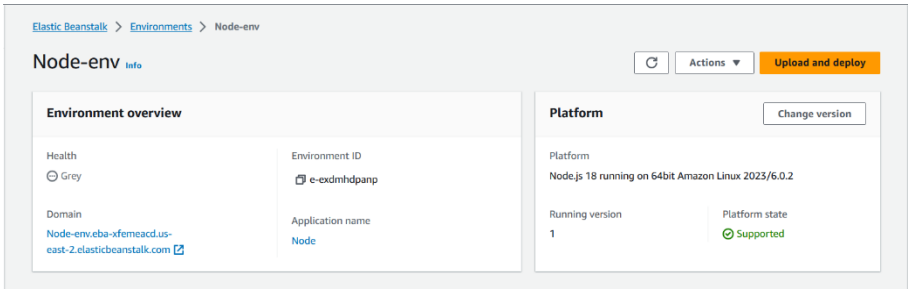
- Next we need to select pipeline script from scm and choose your repo link and Jenkinsfile.

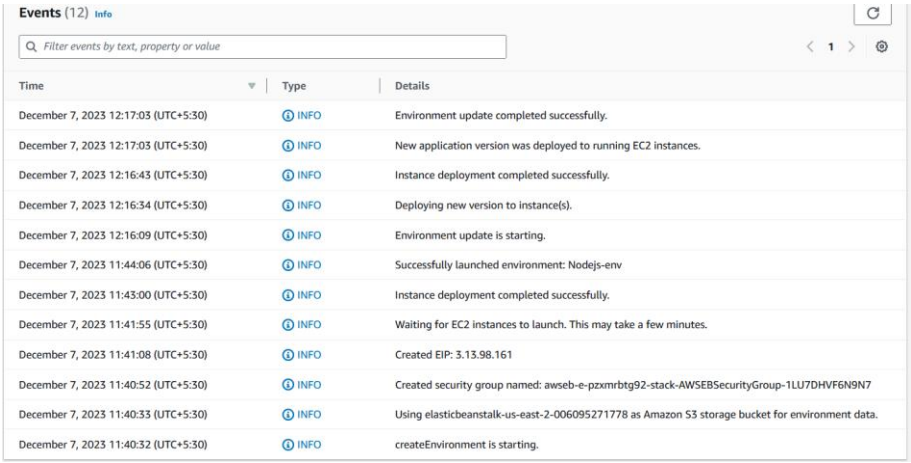




```
    "EnvironmentId": "e-pzxmrbtg92",
    "ApplicationName": "Nodejs",
    "VersionLabel": "8",
    "SolutionStackName": "64bit Amazon Linux 2023 v6.0.3 running Node.js 18",
    "PlatformArn": "arn:aws:elasticbeanstalk:us-east-2::platform/Node.js 18 running on 64bit Amazon Linux 2023/6.0.3",
    "EndpointURL": "3.13.98.161",
    "CNAME": "Nodejs-env.eba-bftymmwy.us-east-2.elasticbeanstalk.com",
    "DateCreated": "2023-12-07T06:10:32.974000+00:00",
    "DateUpdated": "2023-12-07T06:46:10.040000+00:00",
    "Status": "Updating",
    "AbortableOperationInProgress": true,
    "Health": "Grey",
    "Tier": {
        "Name": "WebServer",
        "Type": "Standard",
        "Version": "1.0"
    },
    "EnvironmentArn": "arn:aws:elasticbeanstalk:us-east-2:006095271778:environment/Nodejs/Nodejs-env"
}
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // withCredentials
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```


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





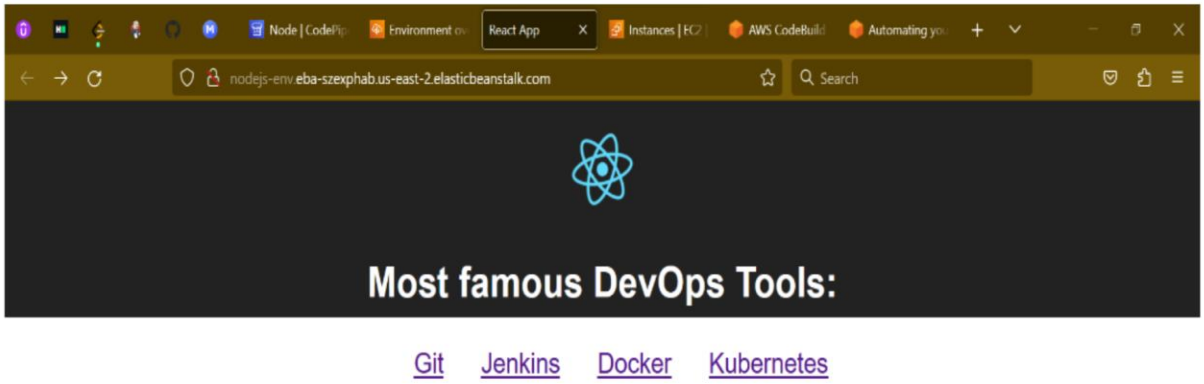
Events (12) Info

Filter events by text, property or value

1

Time	Type	Details
December 7, 2023 12:17:03 (UTC+5:30)	INFO	Environment update completed successfully.
December 7, 2023 12:17:03 (UTC+5:30)	INFO	New application version was deployed to running EC2 instances.
December 7, 2023 12:16:43 (UTC+5:30)	INFO	Instance deployment completed successfully.
December 7, 2023 12:16:34 (UTC+5:30)	INFO	Deploying new version to instance(s).
December 7, 2023 12:16:09 (UTC+5:30)	INFO	Environment update is starting.
December 7, 2023 11:44:06 (UTC+5:30)	INFO	Successfully launched environment: Nodejs-env
December 7, 2023 11:43:00 (UTC+5:30)	INFO	Instance deployment completed successfully.
December 7, 2023 11:41:55 (UTC+5:30)	INFO	Waiting for EC2 instances to launch. This may take a few minutes.
December 7, 2023 11:41:08 (UTC+5:30)	INFO	Created EIP: 3.13.98.161
December 7, 2023 11:40:52 (UTC+5:30)	INFO	Created security group named: awseb-e-pzxmrbtg92-stack-AWSEBSecurityGroup-1LU7DHVF6N9N7
December 7, 2023 11:40:33 (UTC+5:30)	INFO	Using elasticbeanstalk-us-east-2-006095271778 as Amazon S3 storage bucket for environment data.
December 7, 2023 11:40:32 (UTC+5:30)	INFO	createEnvironment is starting.

- Check the Beanstalk environment to reach our website.



Reference:

github url: https://github.com/Vennilavan12/AWS-EB.git