

**Build a CI/CD Pipeline using Jenkins, SonarQube, Docker and AWS**

TABLE OF CONTENTS

* [Creating EC2 Instances](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-creating-ec2-instances)
  + [Setting Security Group](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-setting-security-group)
* [Setting up Instances](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-setting-up-instances)
  + [Installing SonarQube on 2nd EC2 Instance](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-installing-sonarqube-on-2nd-ec2-instance)
* [Integrating SonarQube For Jenkins](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-integrating-sonarqube-for-jenkins)
* [Now open Jenkins](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-now-open-jenkins)
* [Create a Jenkins Job for Deploying our Web Application](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-create-a-jenkins-job-for-deploying-our-web-application)
  + [Configuring Pipeline for SonarQube](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-configuring-pipeline-for-sonarqube)
  + [Configuring Pipeline for Docker](https://ritik3311.hashnode.dev/build-a-cicd-pipeline-using-jenkins-sonarqube-docker-and-aws#heading-configuring-pipeline-for-docker)

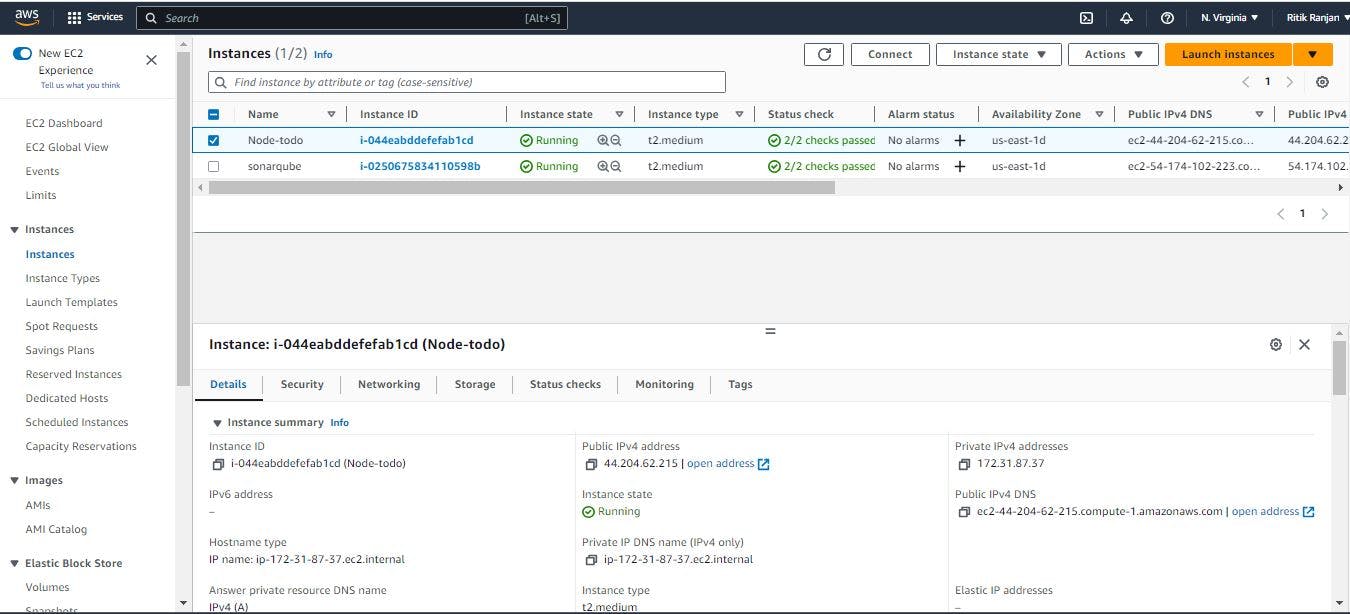
We'll use Jenkins, SonarQube, Docker, and AWS to build an automated CI/CD pipeline for your web application. This pipeline will build, test, analyze, and deploy your project to your AWS EC2 instance.

**Creating EC2 Instances**

Create two EC2 Instances

1st for Jenkins and Docker

2nd for SonarQube



**Setting Security Group**

Under the security details for instance, click on security groups and then click on "**Edit Inbound rules"**.

Generally, there is only one rule present for port 22 for SSH.  
We have to add some more rules as:

In 1st Instance

* Port 8080, for Jenkins
* Port 8000, for exposing our web application

***Set the source to "Anywhere-IPv4" i.e. 0.0.0.0/0***

Now, click on save rules to save your changes.

In 2nd Instance

* Port 9000, for SonarQube

***Set the source to "Anywhere-IPv4" i.e. 0.0.0.0/0***

Now, click on save rules to save your changes

**Setting up Instances**

Now install Jenkins and Docker on 1st instance and

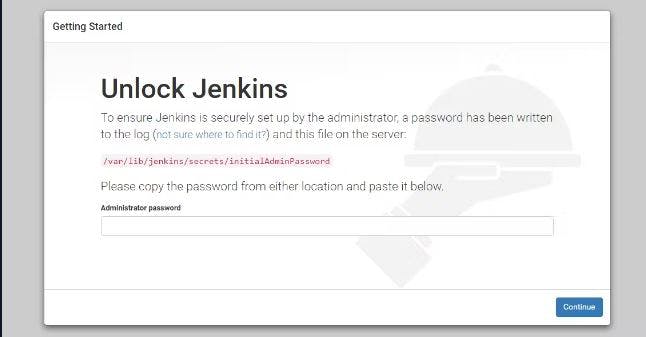
Install SonarQube on 2nd instance

To install Jenkins, Docker and SonarQube on instances, firstly you have to ssh to your instance in your terminal by selecting that instance and then clicking on connect.

Now install Jenkins <https://www.jenkins.io/doc/book/installing/linux/>

Now install Docker "sudo apt-get install [docker.io](http://docker.io/)"

Now, copy the public IPv4 address of Jenkins, paste it into your browser, and put the port number 8080 after it in the format i.e <ip\_address>:8080



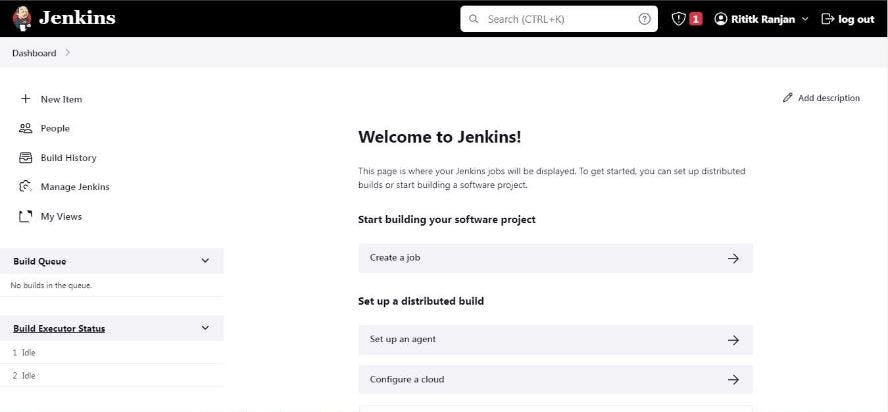
***To unlock jenkins paste the secret key to Administrator password, you can get it with the following command:***

COPY

COPY

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

After unlocking Jenkins click on "Install suggested plugins" and then provide the basic details on the "Create First Admin User" page and get to the Jenkins dashboard after completing all the required steps.



**Installing SonarQube on 2nd EC2 Instance**

To install SonarQube to your instance run the following command in your terminal:

COPY

COPY

sudo apt update

sudo apt install openjdk-17-jre

wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-10.0.0.68432.zip

To unzip the zip file, run the following command:

COPY

COPY

sudo apt install unzip

unzip sonarqube-10.0.0.68432.zip

Now, head over to the Linux directory to execute the installation file:

COPY

COPY

cd /home/ubuntu/sonarqube-10.0.0.68432/bin/linux-x86-64

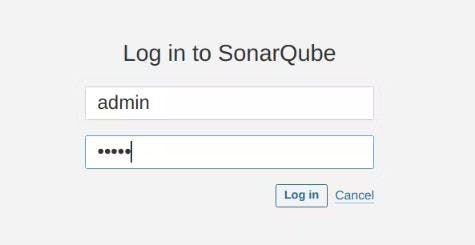
./sonar.sh console

#this will start the SonarQube on port 9000

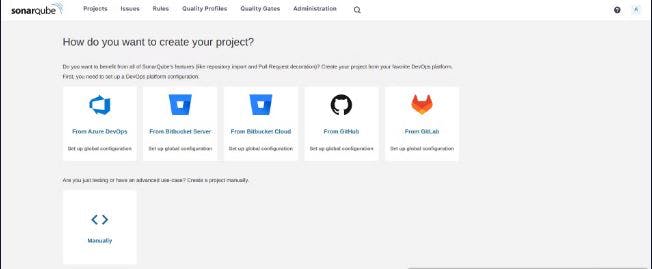
**Integrating SonarQube For Jenkins**

Copy the Public IP of SonarQube and paste it on the browser and put port 9000 after it in such format <public IP>:9000  
For ex:- 54.174.102.233 :9000

Now, login with username - admin & password - admin

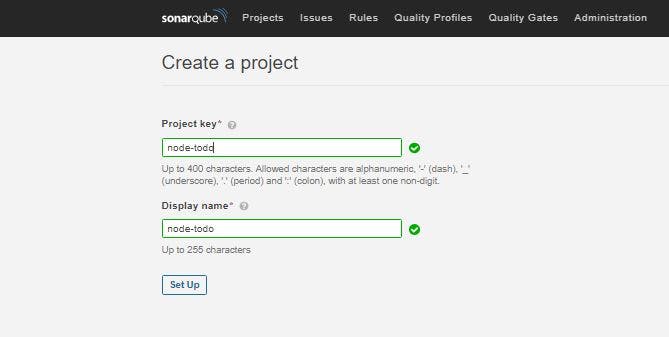


After login and changing your password, you'll land on the SonarQube dashboard.

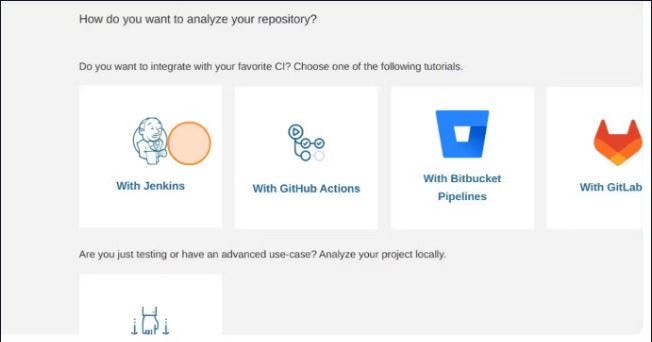


Click "Manually" icon.

Give a name to your Project and provide the name of the branch you want to analyze.

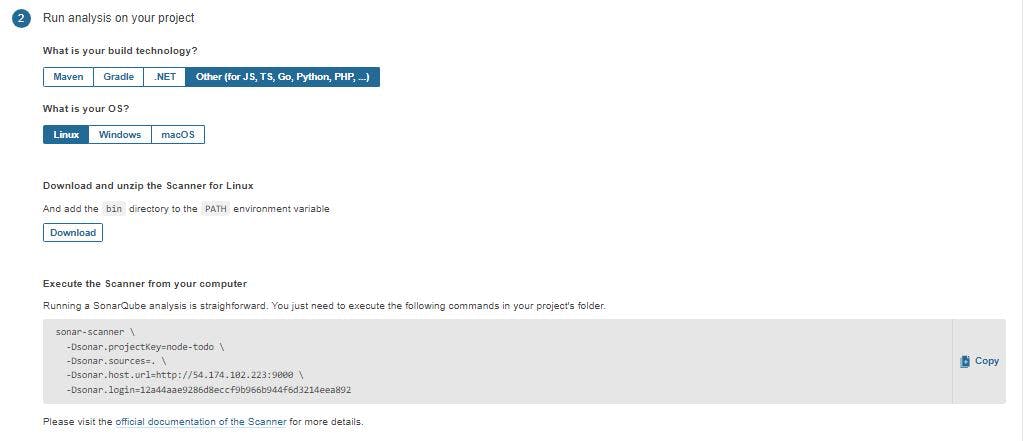


Now, we have many options to select for Continuous Integration tools. For now, we will move forward with Jenkins.



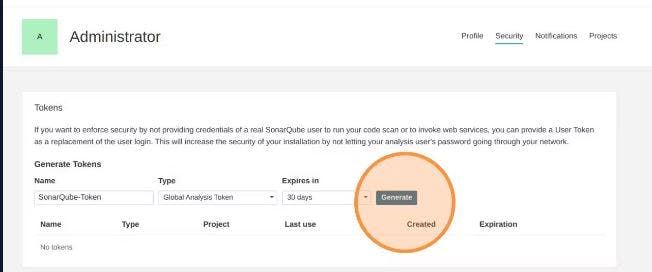
On the next page, we'll choose GitHub for the DevOps platform.

Now, it will ask for configuration. Click continue up to the 3rd step and select other as we are going to deploy our application. Copy the code that it is providing and save it somewhere safe as we will going to use it later.



Click on Finish this tutorial to save it, and after saving it move to **My Accounts>Security>Generate Tokens**

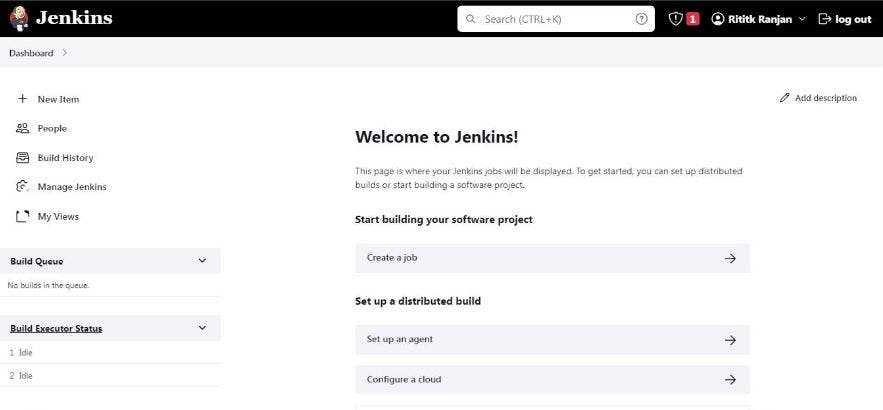
Now, create a Token for integrating with Jenkins.



Copy the token and save it somewhere safe, as we are going to use it later.

**Now open Jenkins**

***Now, head over to Jenkins Dashboard, by pasting the Public IP of Jenkins and put port 8080 after it in such fromat <Public IP>:8080***



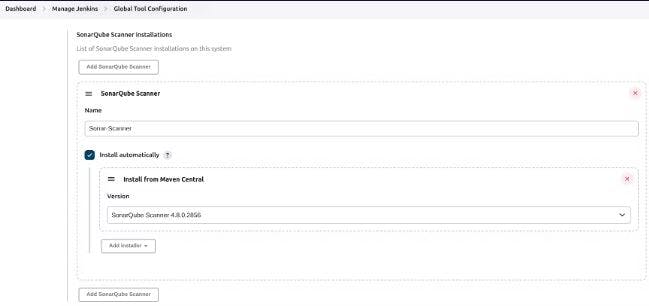
We have to install some plugins on Jenkins for setting up SonarQube.

On Jenkins Dashboard, click on **Manage Jenkins>>Manage Plugins>>Available Plugins**

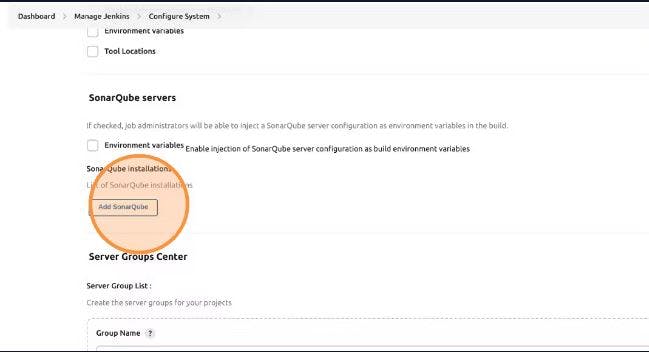
Now, in Available Plugins, search for **SonarQube Scanner** **.**  
After selecting them click on "**Install without restart".**

On Jenkins Dashboard click on **Manage Jenkins>>Global Tool Configuration** and search for **SonarQube Scanner** under it.

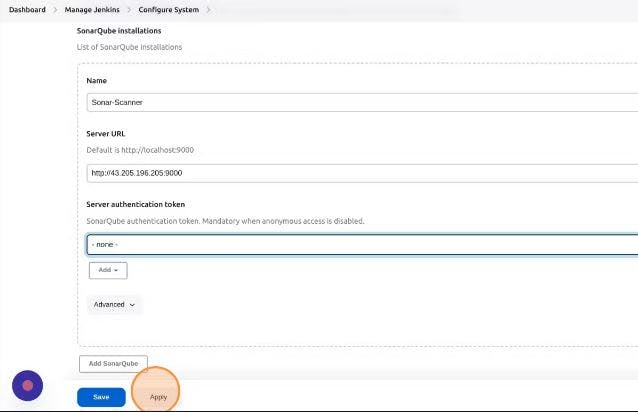
After clicking on "**Add SonarQube Scanner"**, give a suitable name to it and check on **Install automatically** for its installation, and then **save it**.



Now, on **Manage Jenkins>>Configure system**, search for **SonarQube servers** and click on "**Add SonarQube"**

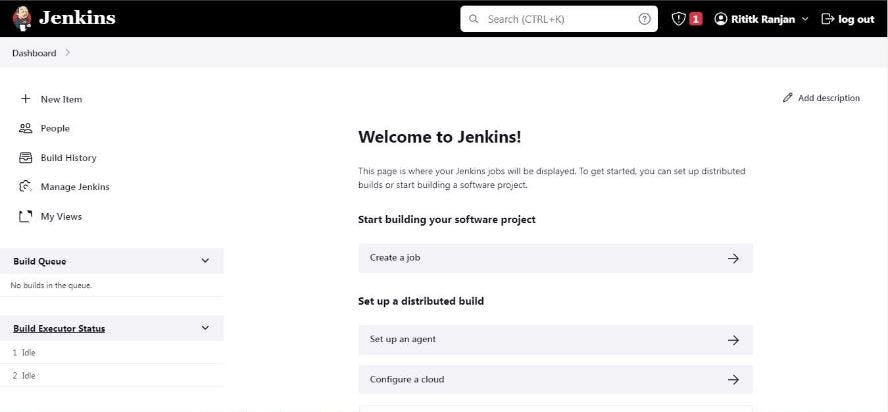


Name it accordingly, copy the Public IP of SonarQube and paste it, and put port 9000 after it in such format http://<Public IP>:9000 . Apply & save it.

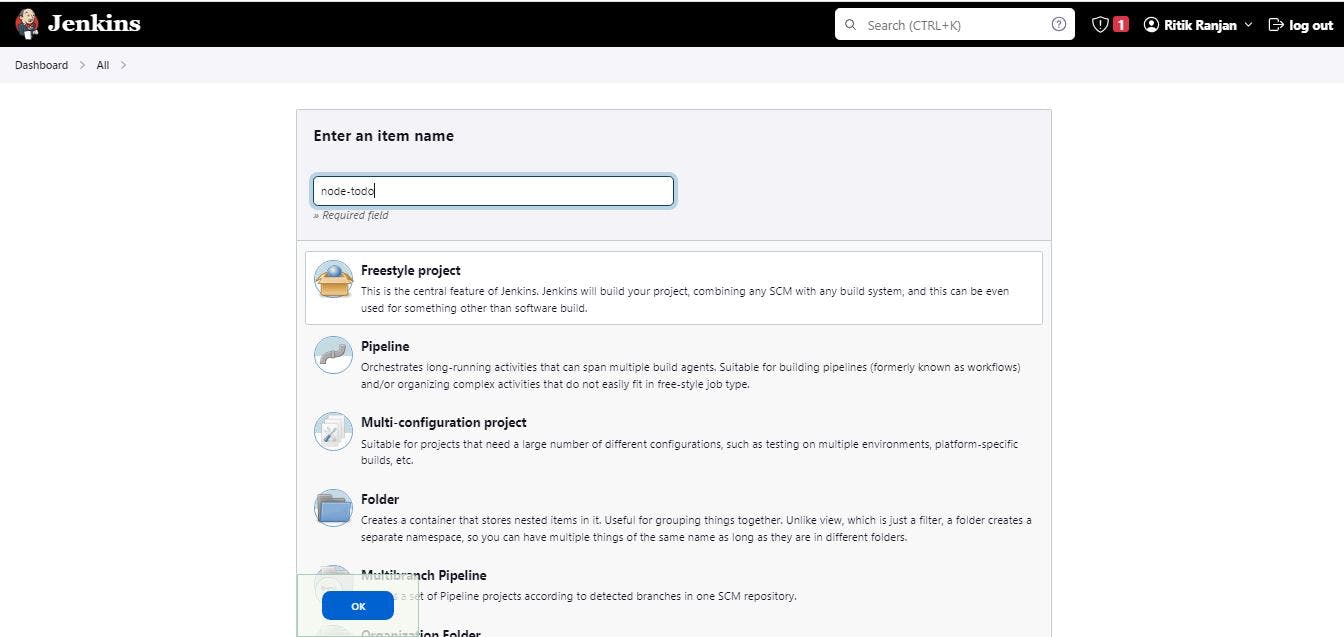


**Create a Jenkins Job for Deploying our Web Application**

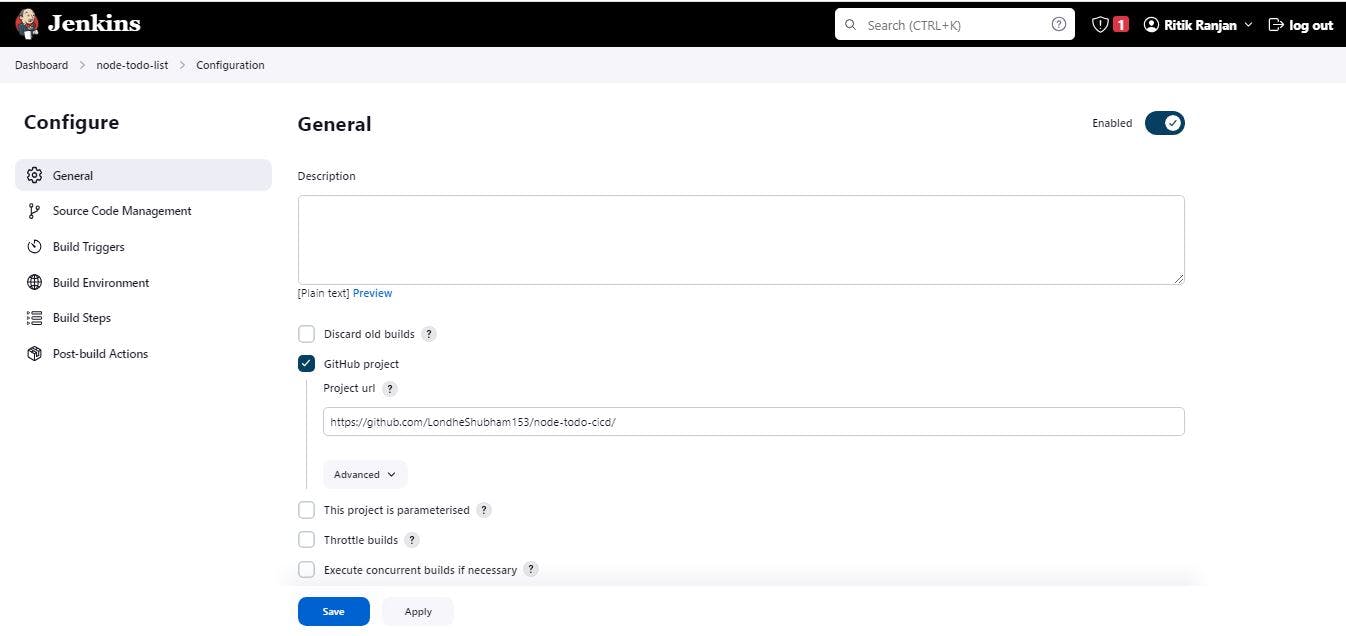
On the Jenkins Dashboard, click on **"Create a Job".**

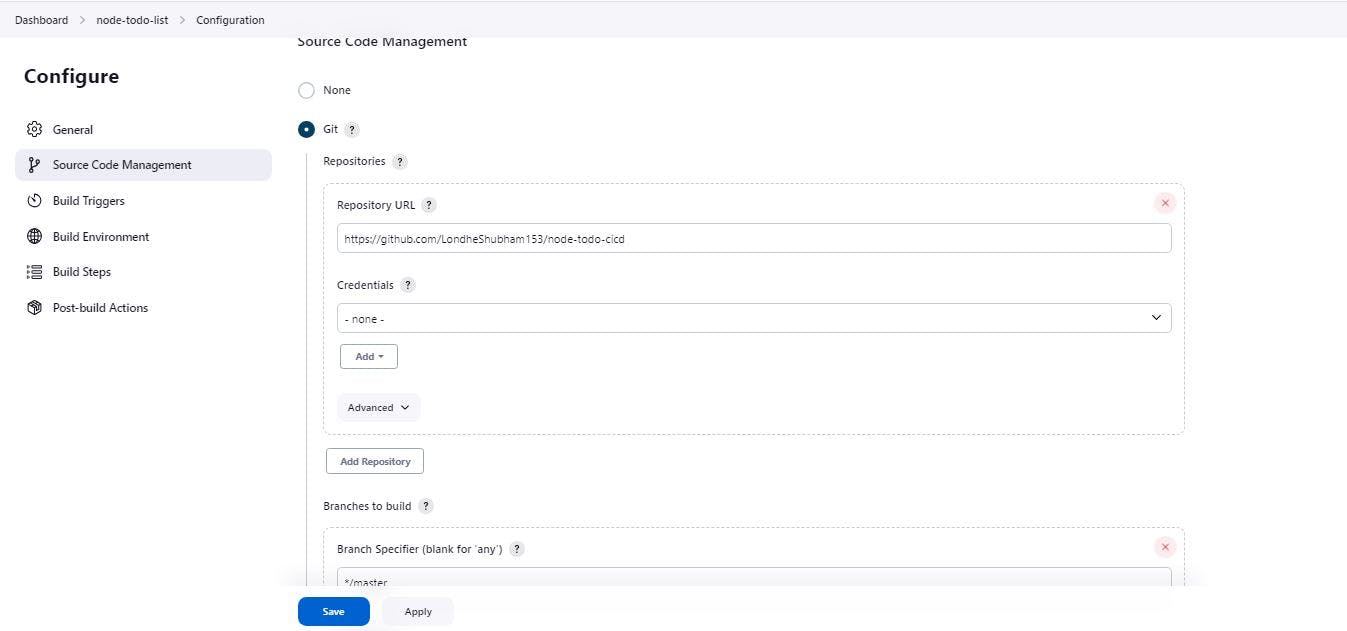


Give it a name, select the type of pipeline you want to create, and then click **"OK"**.



Under the configuration setting, select **"Git"** for **Source Code Management**. Copy and paste the git link from the GitHub repository.



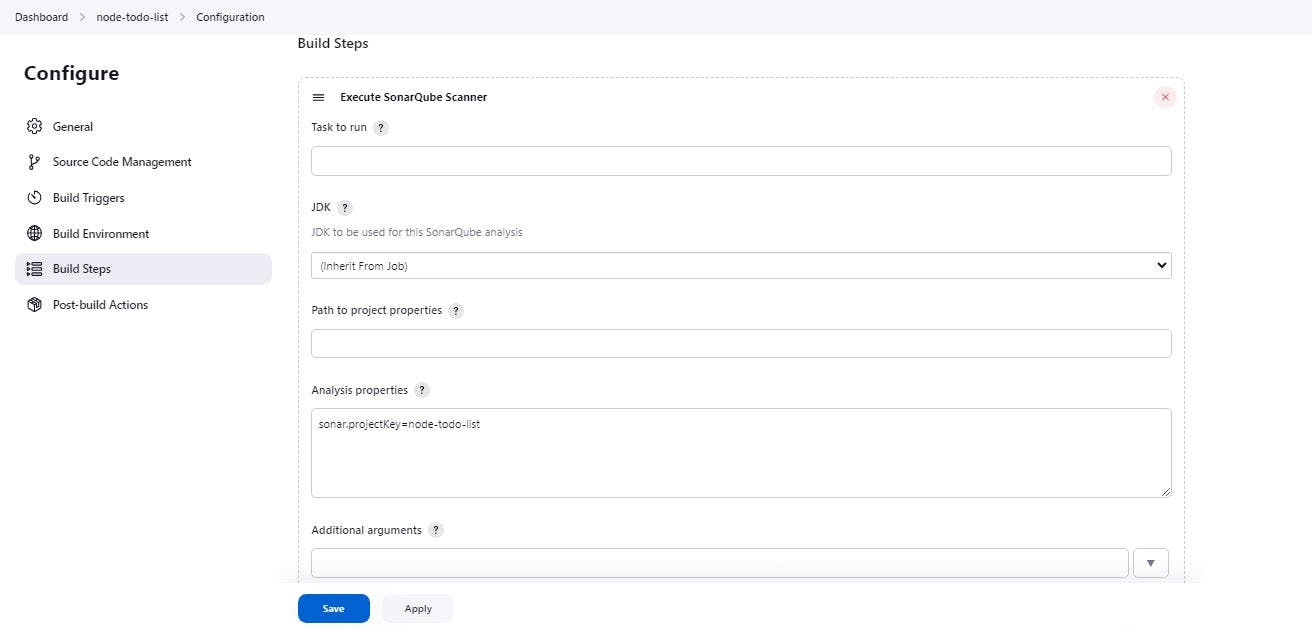


Specify the branch you want to build and deploy.

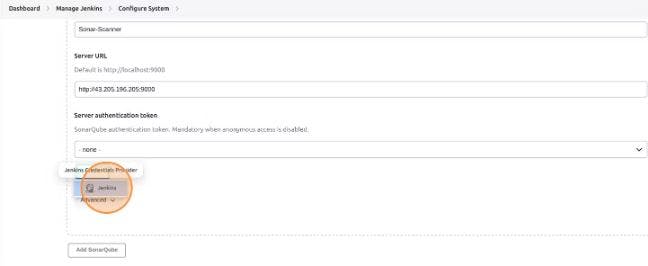
**Configuring Pipeline for SonarQube**

Search for **Build Steps** and click on **"Add build step"** and select **"SonarQube Scanner".**

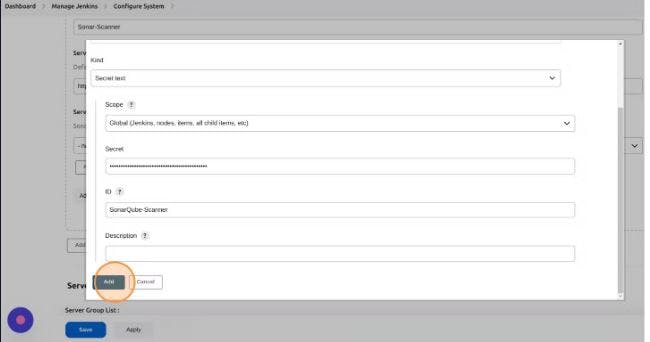
Now, paste the code in the **Analysis Properties** field, which you copied from SonarQube which we kept saved.



Apply and save it, and head over to J**enkins Dashboard>>Manage Plugins>>Configure System** and add server authentication token for SonarQube by clicking on **"Add"**, and selecting **Jenkins**.



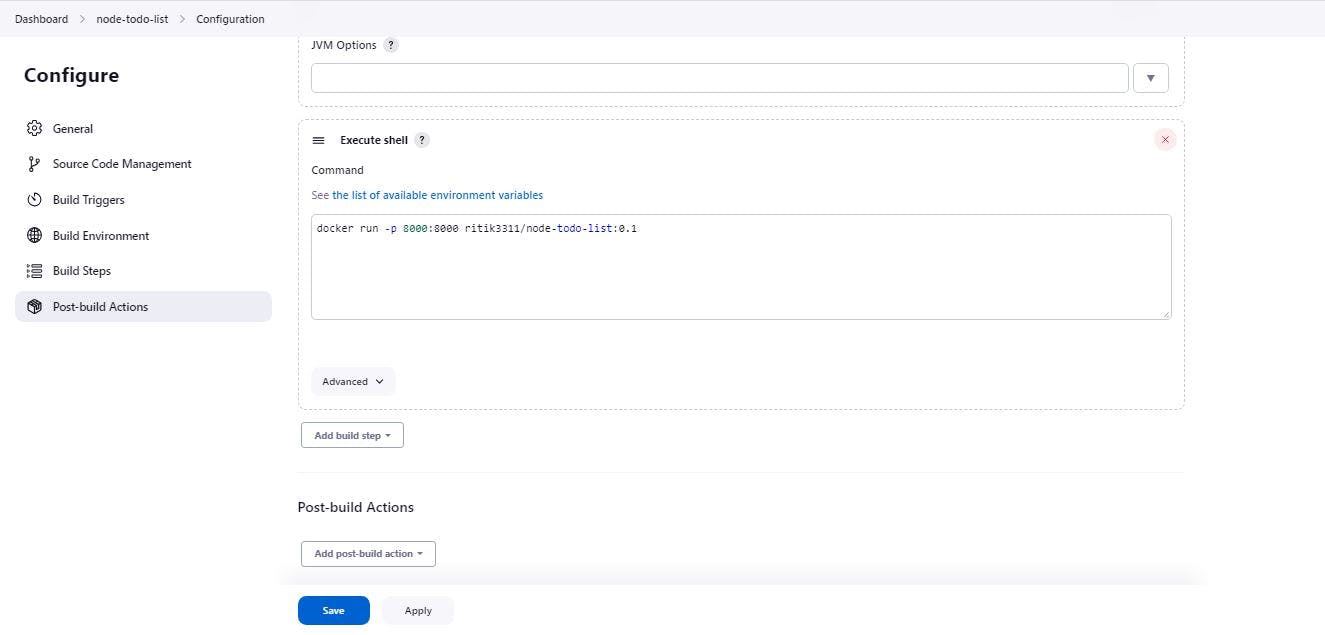
Change Kind to **secret text** and paste the secret token key which we copied earlier and saved it. Give it an ID name accordingly.



Now, for the **server authentication token**, from the scroll bar select the ID name, the one which you just created.

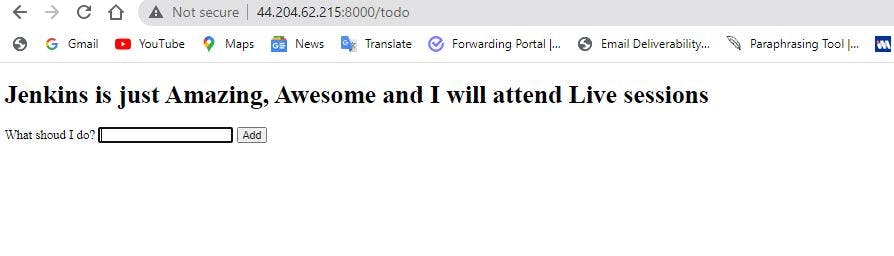
**Configuring Pipeline for Docker**

U**nder the CI-CD Pipeline** on Jenkins Dashboard click on "**Configure**", search for **Build Steps,** and select **Execute shell**.



Apply and save it.

Now, under the CI-CD Pipeline, click **"Build Now".**  
If everything works fine, then you'll see success. Here is our live website working on Port 8000 on Docker.



Now check SonarQube

