DevOps Project Part2

Install SonarQube and create RDS with no public access:

# cd /opt

# wget <https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.7.6.zip>

# unzip sonarqube-6.7.6.zip

# cd /opt/sonarqube-6.0

Install mysql – yum install mysql -y

mysql -h < RDS\_DATABAE\_ENDPOINT > -u admin -p admin123

CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;

Create a local and a remote user

CREATE USER sonar@localhost IDENTIFIED BY 'sonar';

CREATE USER sonar@'%' IDENTIFIED BY 'sonar';

Grant database access permissions to users

GRANT ALL ON sonar.\* TO sonar@localhost;

GRANT ALL ON sonar.\* TO sonar@'%';

check users and databases

use mysql

show databases;

Install java

#**sudo yum install java-1.8.0**

Edit sonar.properties file to uncomment and provide required information for below properties.

**File Name: /opt/sonar/conf/sonar.properties**

**sonar.jdbc.username=admin 🡪 (db username & pw)**

**sonar.jdbc.password=admin123**

**sonar.jdbc.url=jdbc:mysql://<RDS\_DATABAE\_ENDPOINT>:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false**

**sonar.web.host=0.0.0.0**

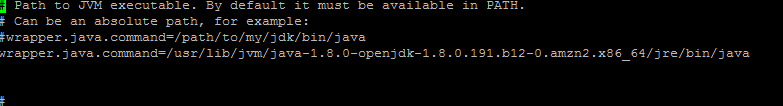
**sonar.web.context=/sonar**

Change owner from root to ec2-user, since sonarqube won’t start from root user

#**chown -R ec2-user:ec2-user /opt/sonarqube**

Do changes in wrapper.conf file

**#vi /opt/sonar/conf/wrapper.conf** 🡪 update the Java jre path (refer below screenshot



Start SonarQube service

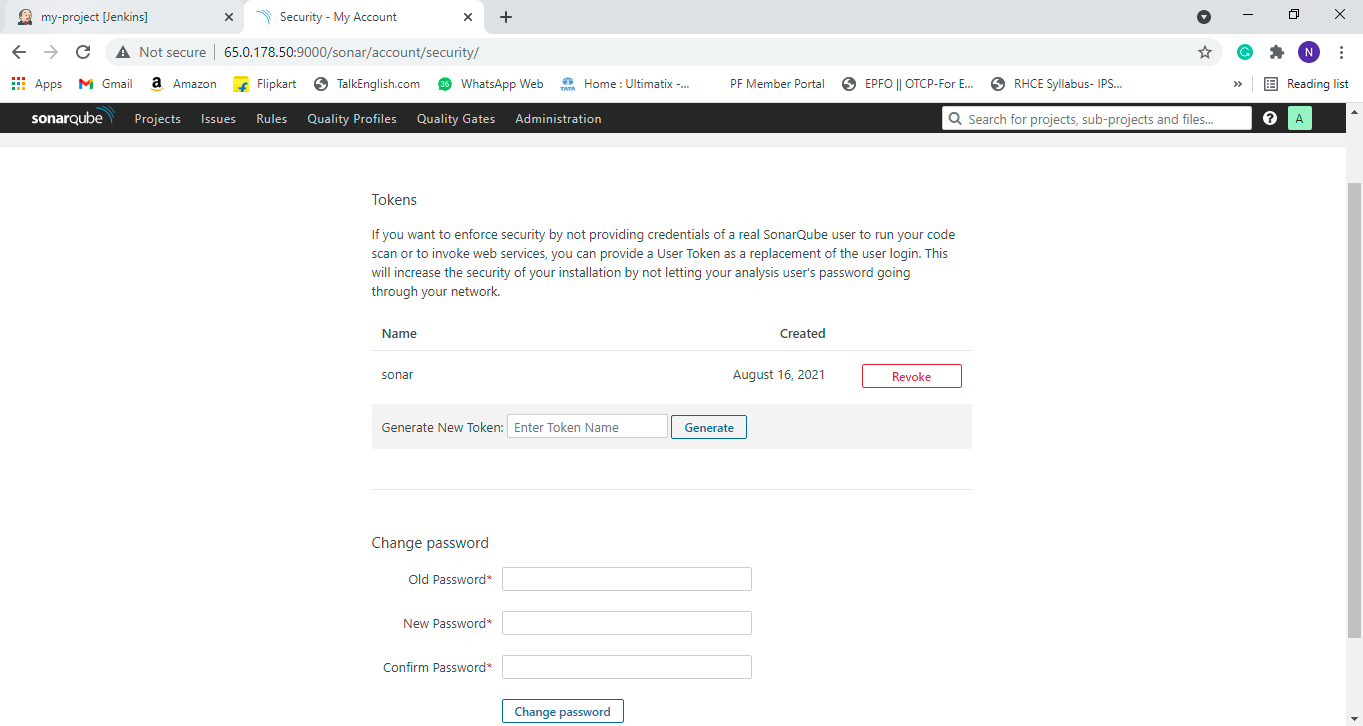
**# cd /opt/sonar/bin/linux-x86-64/**

**# ./sonar.sh start**

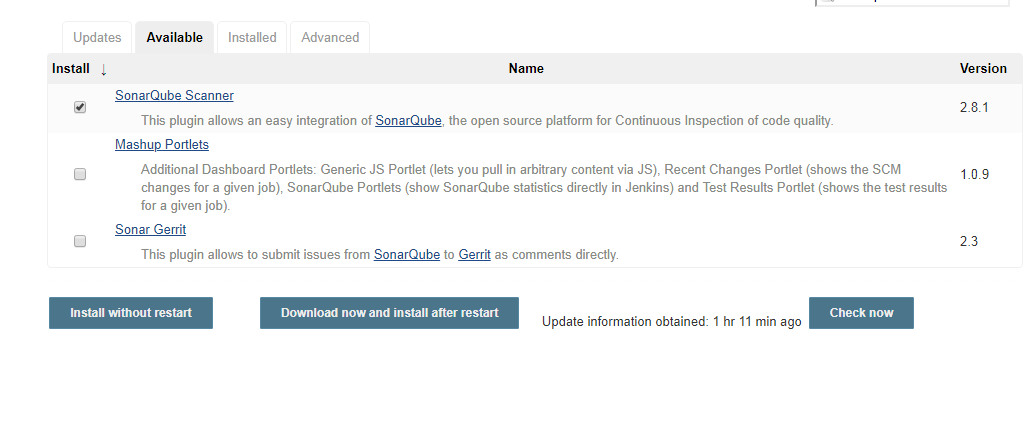
Login into sonarqube machine with **<IP>:9000/sonar** default username & PW is **– admin**

Copy & paste the token from sonarqube or else, after login go to **account/security** to generate a token

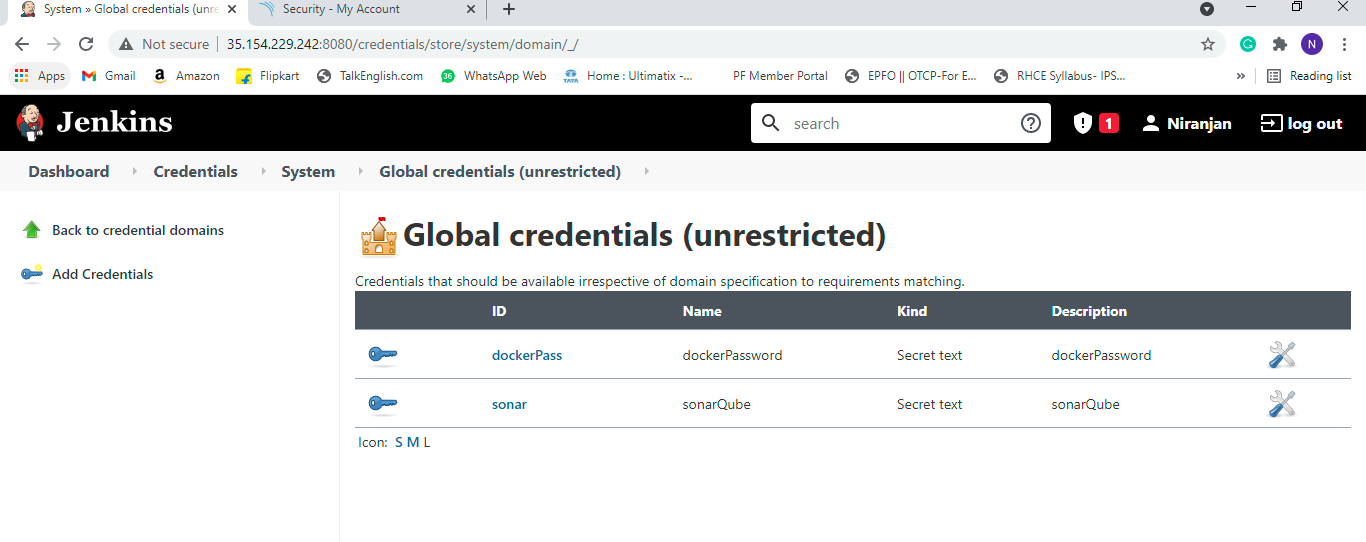
Administrator-🡪my account🡪security



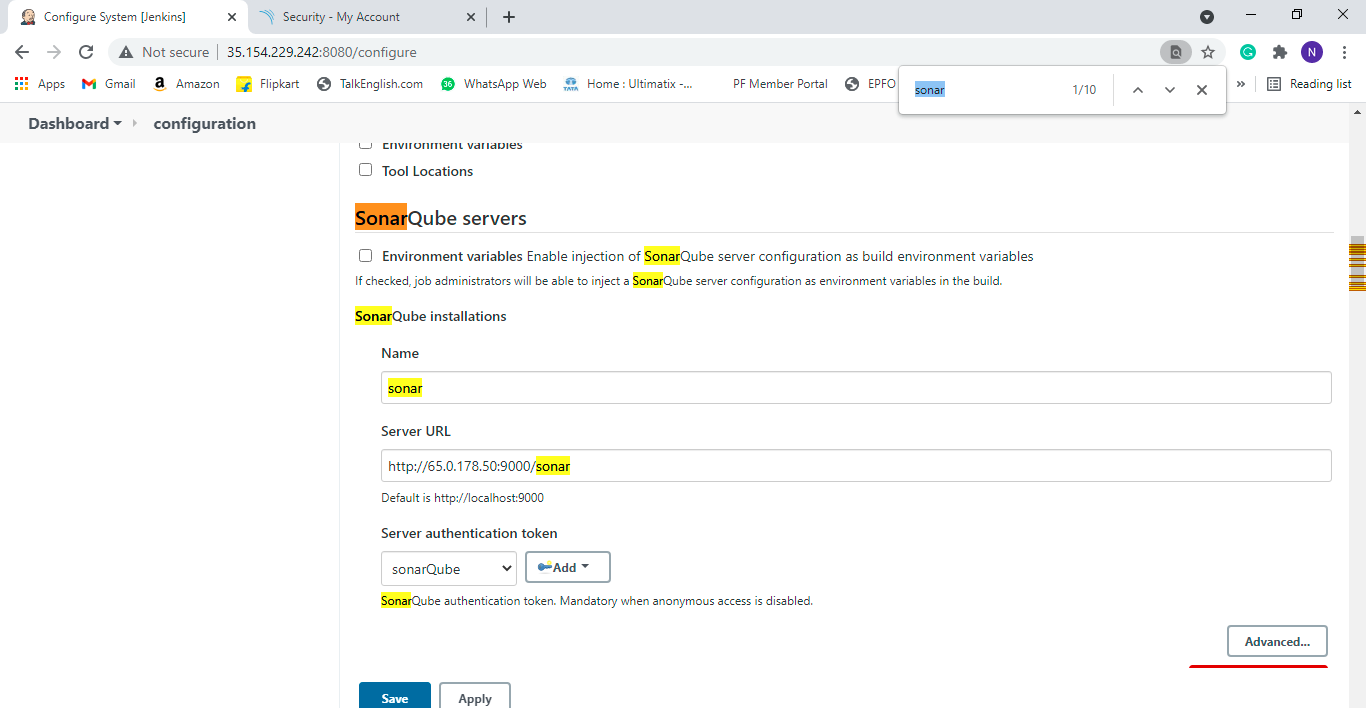
Install **SonarQube Scanner plugin** in Jenkins



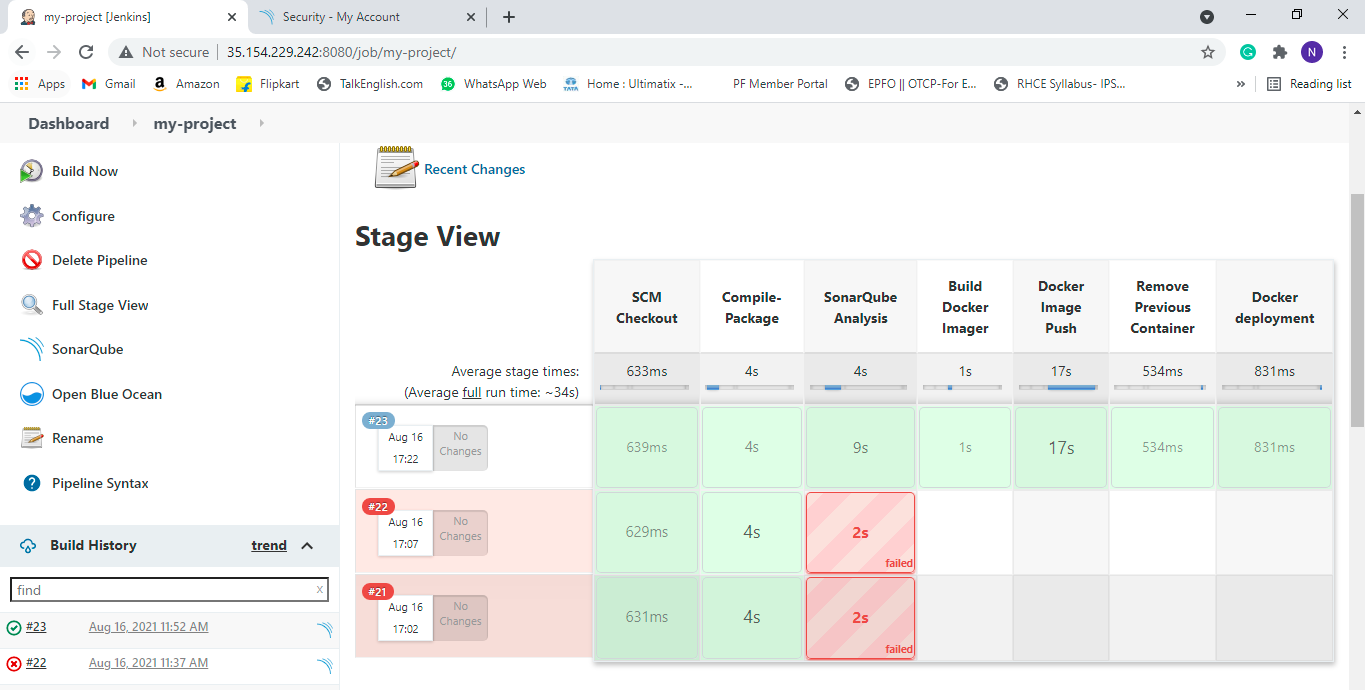
Add sonarQube token on **Manage credentials** in Jenkins



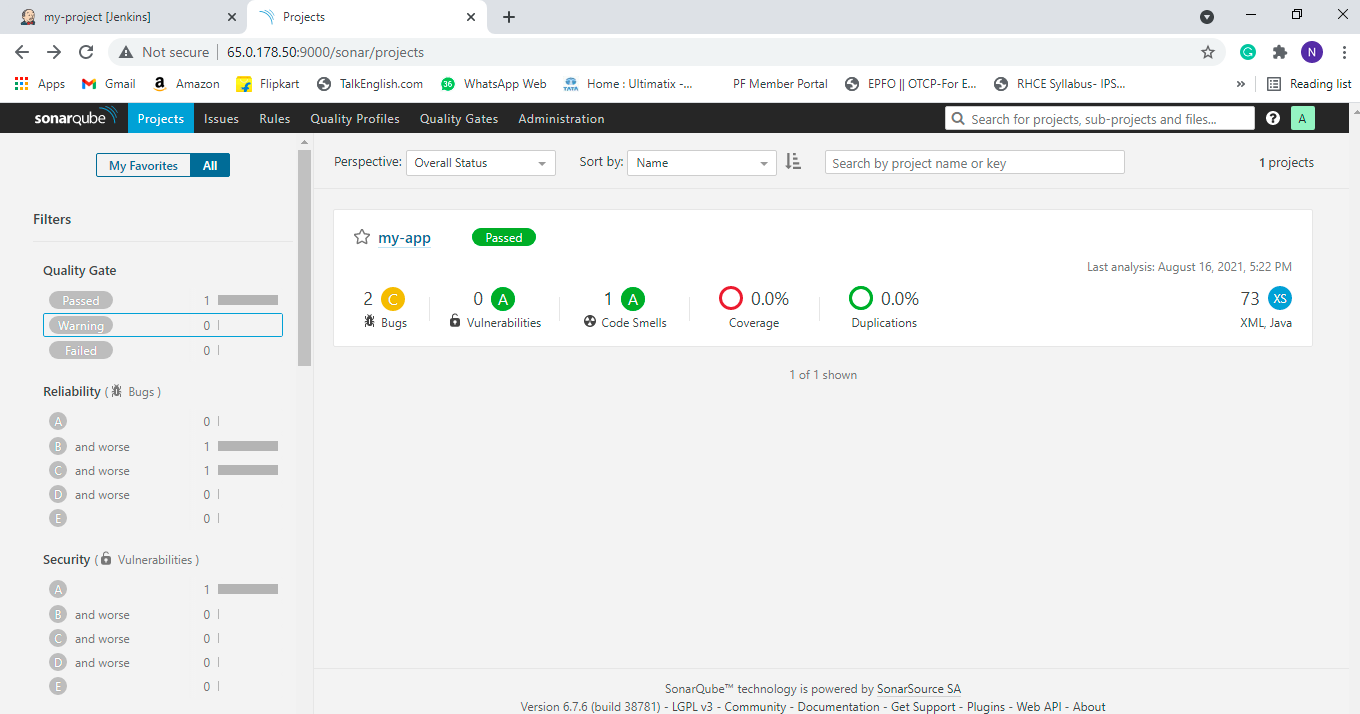
Add SonarQube console IP on **Configure system** in Jenkins



Copy the **SonarQube Analysis** stage from GitHub and paste it in pipeline and click on build now



**Once build successful, verify SonarQube console Output:**



**Refer below urls for Installing & Configuring Nexus**

<https://qiita.com/leechungkyu/items/86cad0396cf95b3b6973>

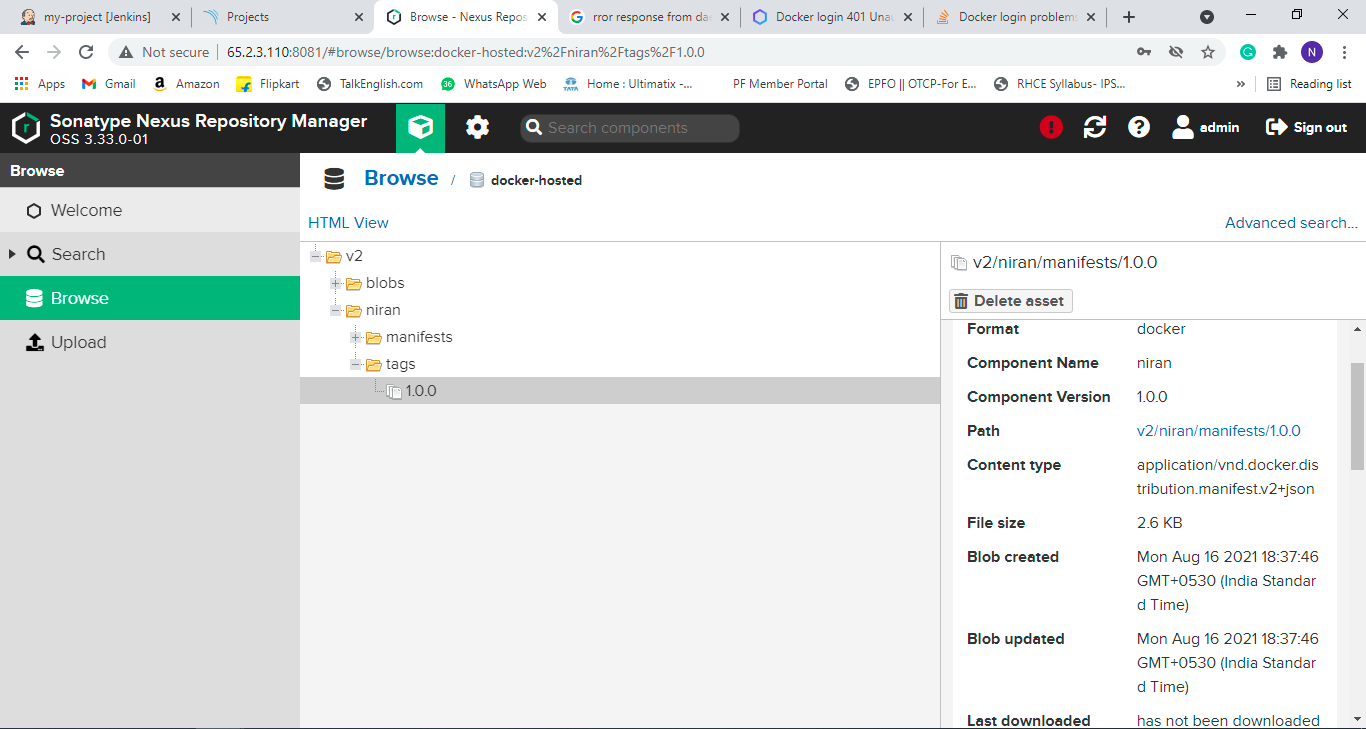
<https://devopscube.com/how-to-install-latest-sonatype-nexus-3-on-linux/>

Once given steps completed, copy & paste the nexus stage which has been written in Jenkins file

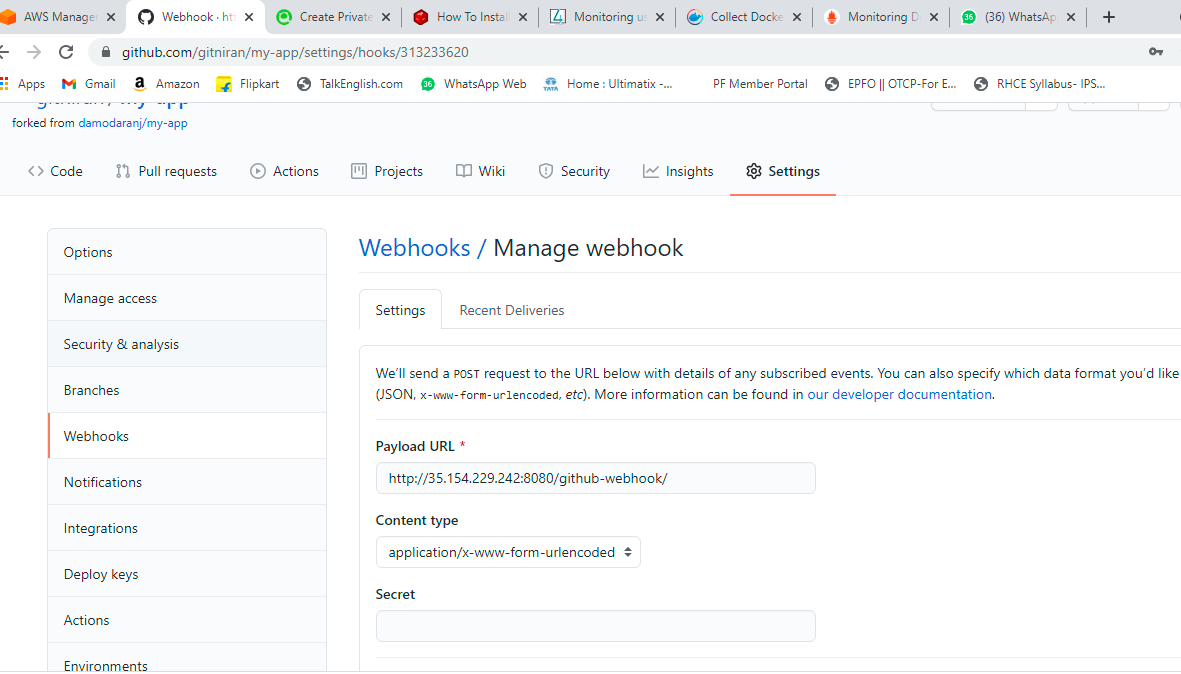
**Login into Nexus Console using “<IP>:8081”**

**Login into Nexus repository using “<IP>:8083”**

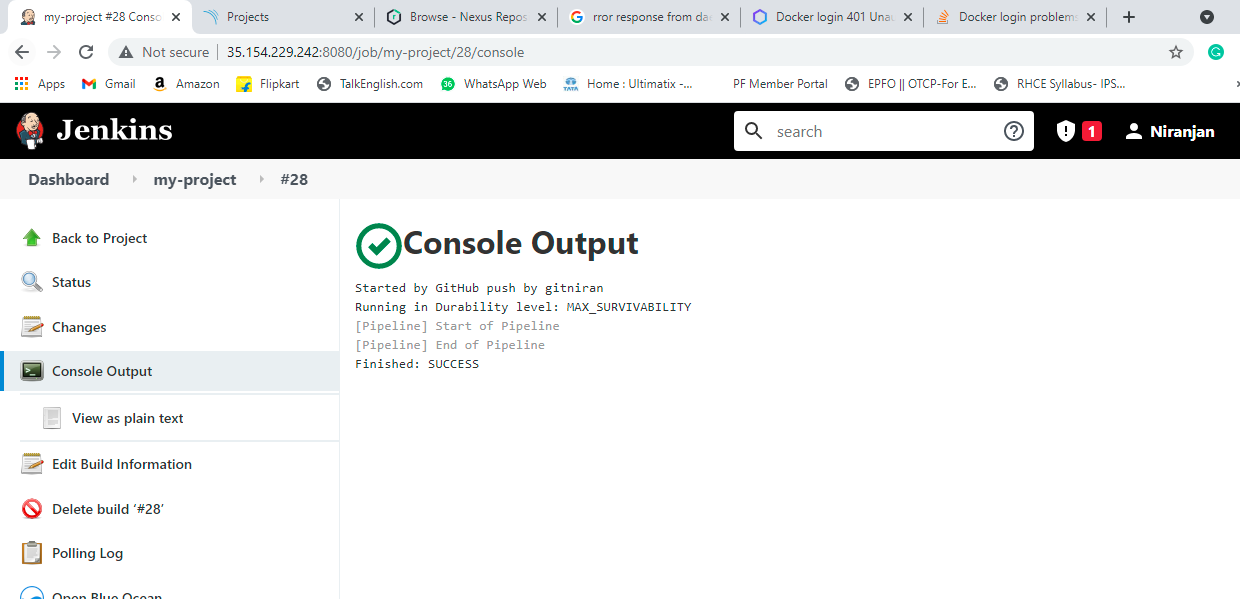
**Nexus Repository Output:**



**Login into GitHub 🡪 go to project settings 🡪 Add webhook 🡪 provide Jenkins <IP>:8080/github-webhook/**



**After enabling GitHub project & GitHub hook trigger for GITScm polling on working project, once developer committed the change in GitHub, job will be triggered automatically.**



**For manual Prometheus installation refer below url** [**https://devops4solutions.com/monitoring-using-prometheus-and-grafana-on-aws-ec2/**](https://devops4solutions.com/monitoring-using-prometheus-and-grafana-on-aws-ec2/)

**Prometheus Installation on nexus machine as docker container**

**#docker run -itd --name prom -v /etc/prometheus:/etc/prometheus -p 9090:9090 prom/Prometheus**

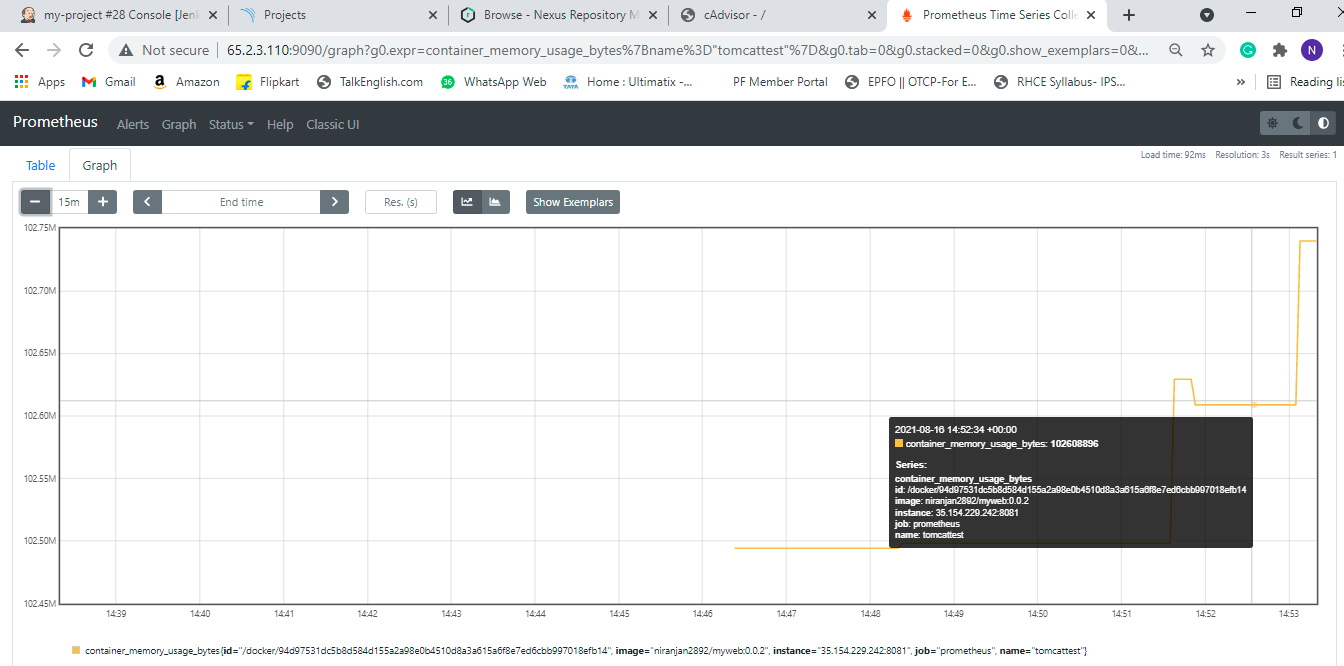
[**https://docs.docker.com/config/daemon/prometheus/**](https://docs.docker.com/config/daemon/prometheus/)

**#vi /etc/Prometheus/Prometheus.yml 🡪 copy & paste the yaml content from above url**

**#docker start containerID**

**Login into Prometheus console using Nexus machine “<IP>:9090”**

**Prometheus Output:**



Go to Client machine and install Cadvisor

**#docker run -d --name cadvisor -p 8081:8080 -v /:/rootfs:ro -v /var/run:/var/run:rw -v /sys:/sys:ro -v /var/lib/docker/:/var/lib/docker:ro google/cadvisor**

Go to Nexus machine where Prometheus container installed; change the port no as 8081

**#vi /etc/Prometheus/Prometheus.yml**

**#docker restart containerID**

Refer official site

[**https://prometheus.io/docs/guides/cadvisor/**](https://prometheus.io/docs/guides/cadvisor/)

**Login into Client machine as <IP>:8081, in our case it is Jenkins server**

**Cadvisor Output:**

