Copy the Downloads plugins And paste in SonarQube/extensions/plugins

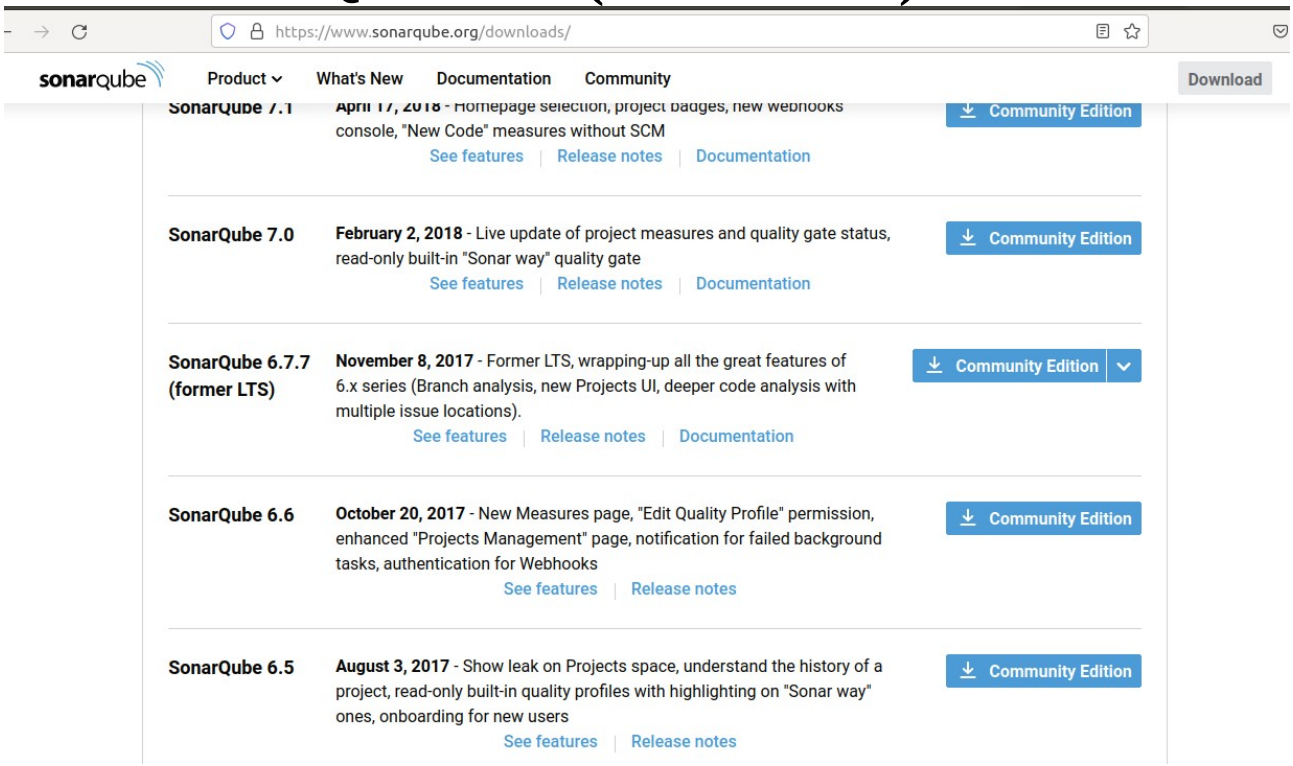
TASK-CI/CD

(SONARQUBE, NEXUS, PUSH IMAGE TO DOCKER-HUB)

Prequisties: SonarQube of version 6.7.7(LTS), Nexus, SonarQube Scanner, Git Hub ,Docker Hub Jenkins

Download Zip file (SonarQube) using below link : https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/

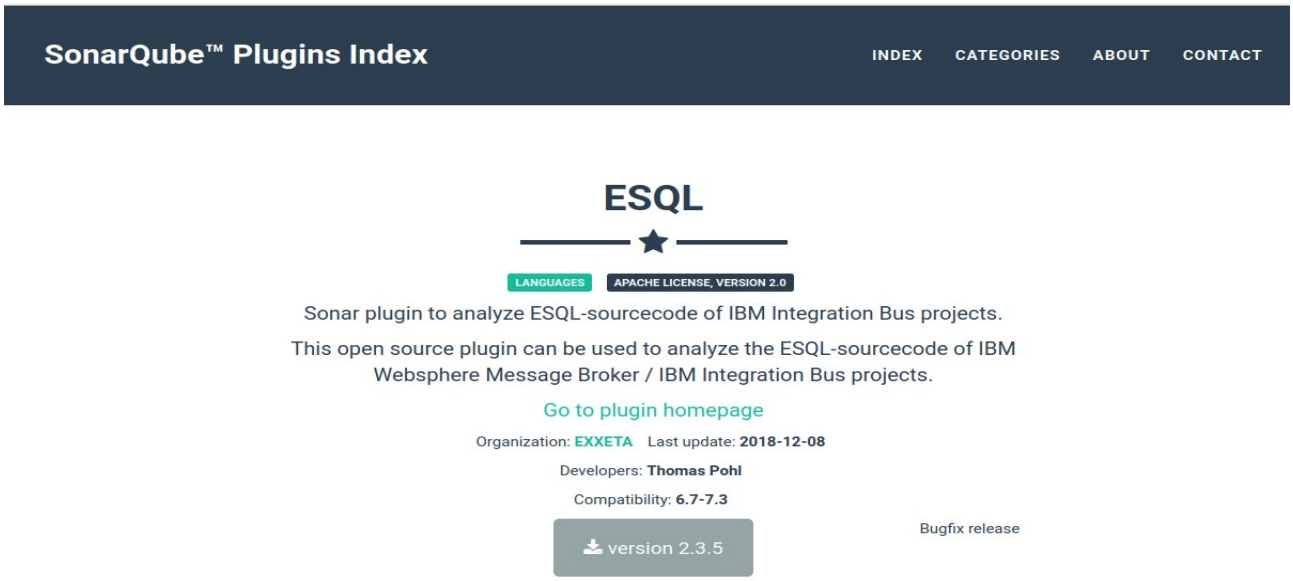
Select SonarQube 6.7.7(former LTS)



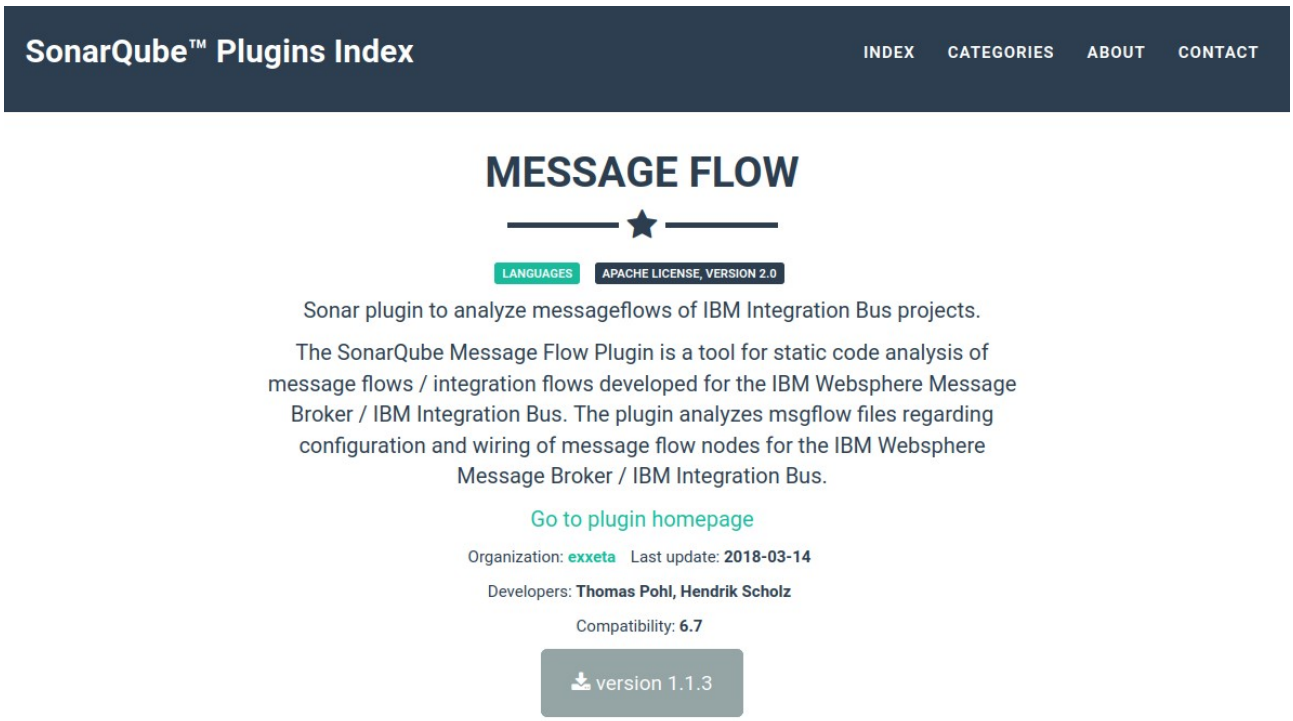
Installing SonarQube:

To Access .Bar files
we need additional Plugins of ESQL and MSGFLOW plugin

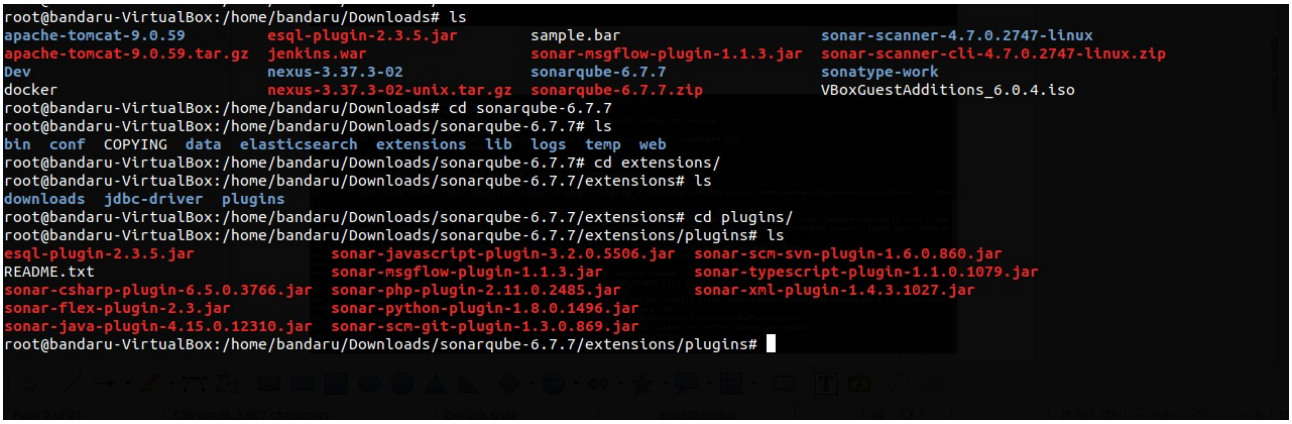
To Download these Plugins
For ESQL go to https://www.sonarplugins.com/esql



For MSGFLOW
go to https://www.sonarplugins.com/msgflow

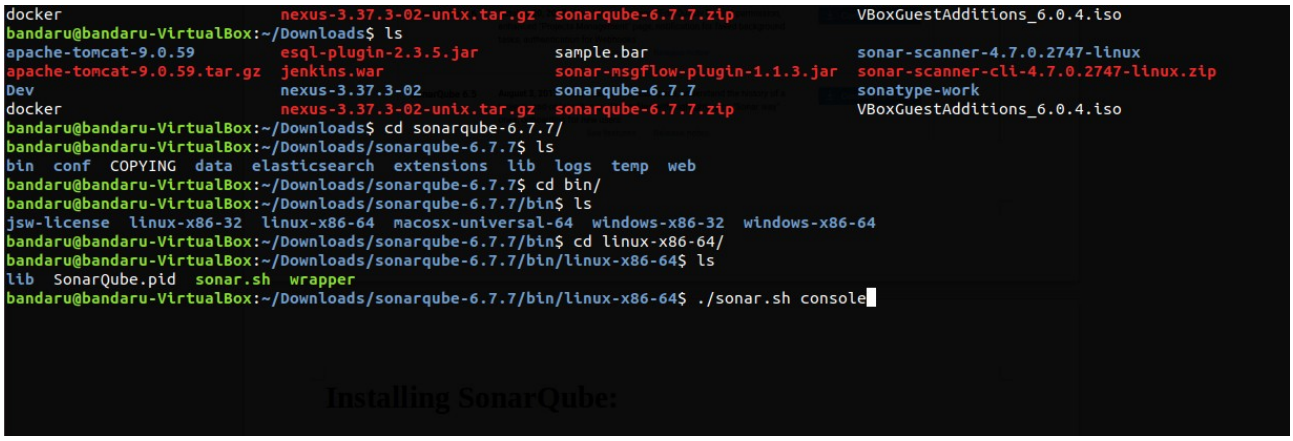


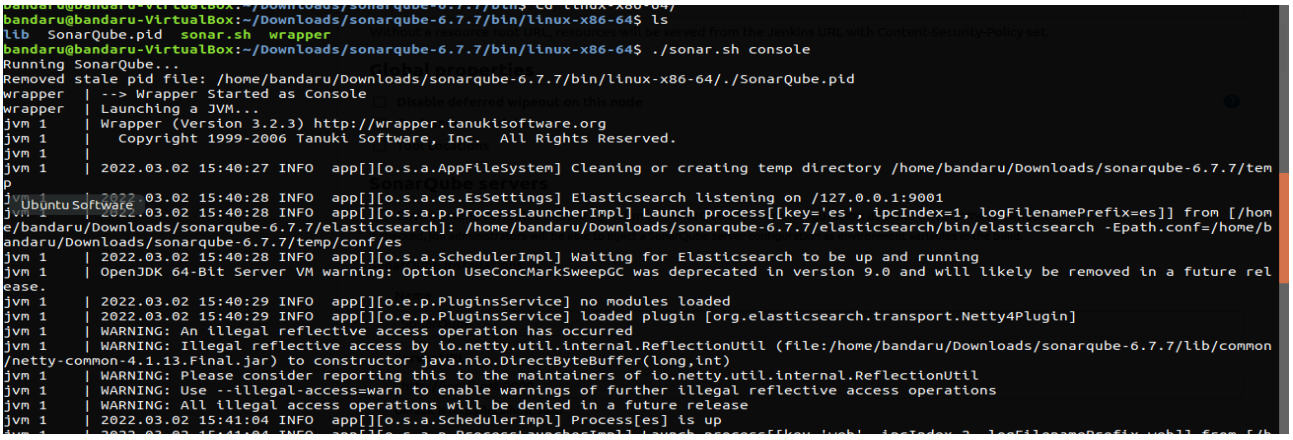
Copy the Downloaded plugins And paste in SonarQube/extensions/plugins



next Installing SonarQube:

SonarQube can be installed only in USER only not as a ROOT user



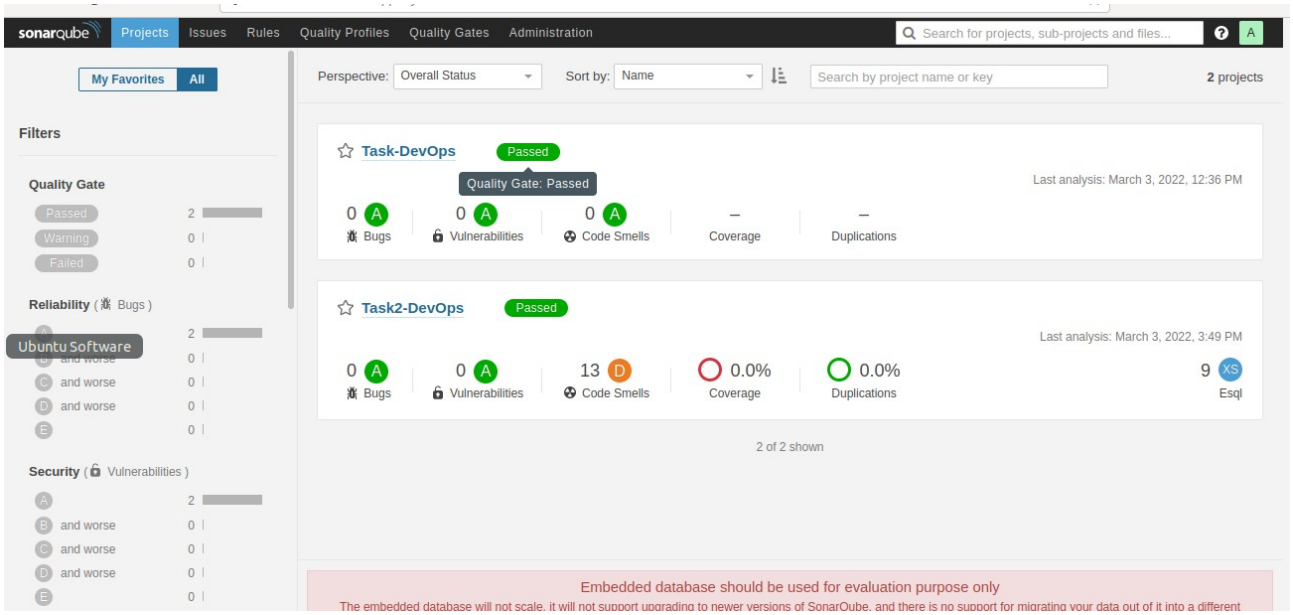


SonarQube by Default run at port no. 9000

To see SonarQube Interface
Migrate to Browser and Type http://localhost:9000

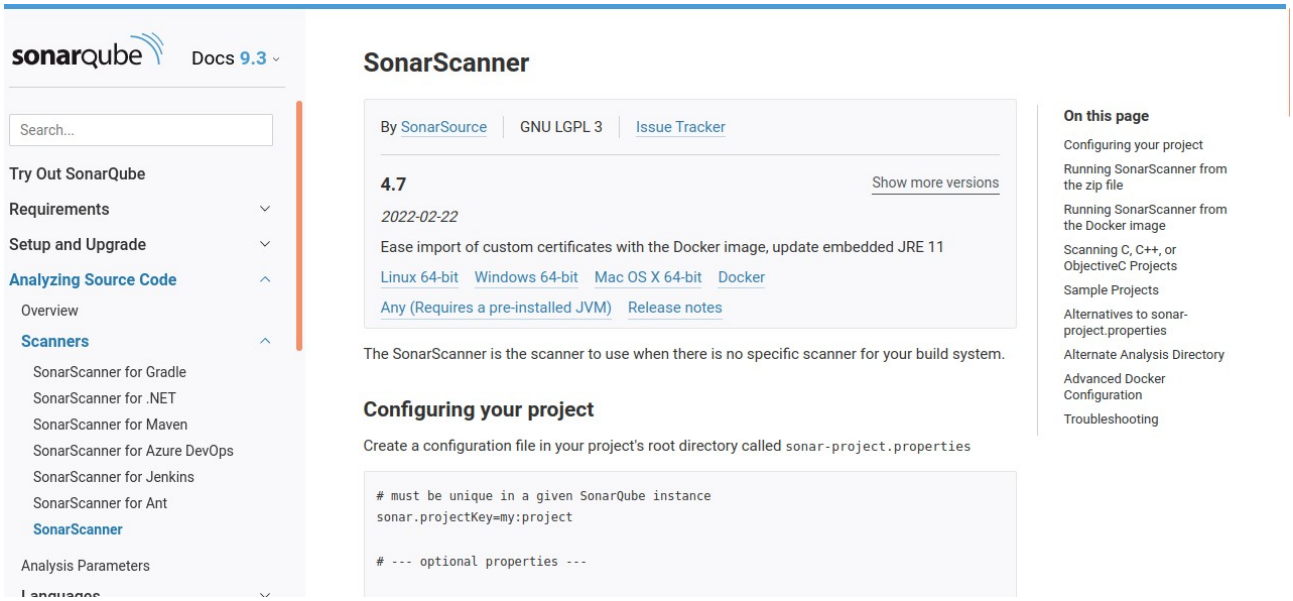
Login Credentials by default are

Username: admin
Password: admin



Download Zip file (SonarScanner) using below link:
https://www.sonarqube.org/downloads/

Select Linux-64-bit



Installing SonarQube Scanner:

```
bandaru@bandaru-VirtualBox:~/Downloads$ ls
apache-tomcat-9.0.59          esql-plugin-2.3.5.jar          sample.bar                      sonar-scanner-4.7.0.2747-linux
apache-tomcat-9.0.59.tar.gz   jenkins.war                    sonar-msgflow-plugin-1.1.3.jar  sonar-scanner-cli-4.7.0.2747-linux.zip
Dev                           nexus-3.37.3-02                sonarqube-6.7.7                 sonatype-work
docker                        nexus-3.37.3-02-unix.tar.gz    sonarqube-6.7.7.zip             VBoxGuestAdditions_6.0.4.iso
bandaru@bandaru-VirtualBox:~/Downloads$ unzip sonar-scanner-cli-4.7.0.2747-linux.zip
```

```
bandaru@bandaru-VirtualBox:~/Downloads$ cd sonar-scanner-4.7.0.2747-linux/
bandaru@bandaru-VirtualBox:~/Downloads/sonar-scanner-4.7.0.2747-linux$ ls
bin  conf  jre  lib
bandaru@bandaru-VirtualBox:~/Downloads/sonar-scanner-4.7.0.2747-linux$ cd conf/
bandaru@bandaru-VirtualBox:~/Downloads/sonar-scanner-4.7.0.2747-linux/conf$ ls
sonar-scanner.properties
bandaru@bandaru-VirtualBox:~/Downloads/sonar-scanner-4.7.0.2747-linux/conf$ vim
sonar-scanner.properties       .sonar-scanner.properties.swp
bandaru@bandaru-VirtualBox:~/Downloads/sonar-scanner-4.7.0.2747-linux/conf$ vim sonar-scanner.properties
```

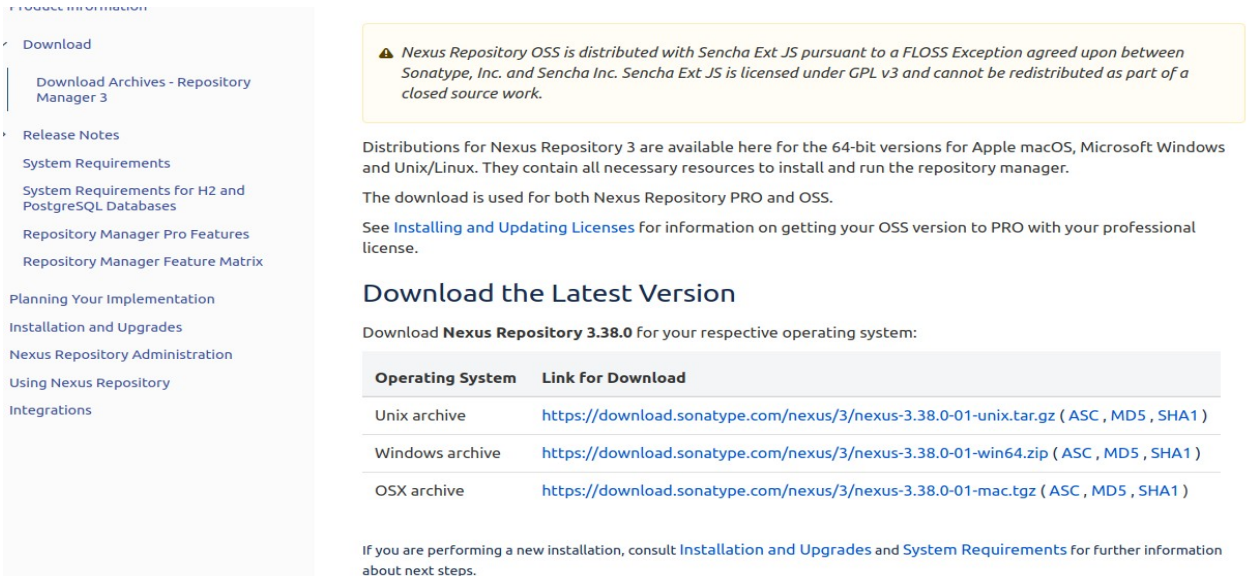
Uncomment the line sonar.host.url=http://localhost:9000

```
#Configure here general information about the environment, such as SonarQube server connection details for example
#No information about specific project should appear here

#----- Default SonarQube server
sonar.host.url=http://localhost:9000

#----- Default source code encoding
#sonar.sourceEncoding=UTF-8

~
~
~
```

Download Tar file (Nexus) using below link:

https://help.sonatype.com/repomanager3/product-information/download

Select Unix Archive



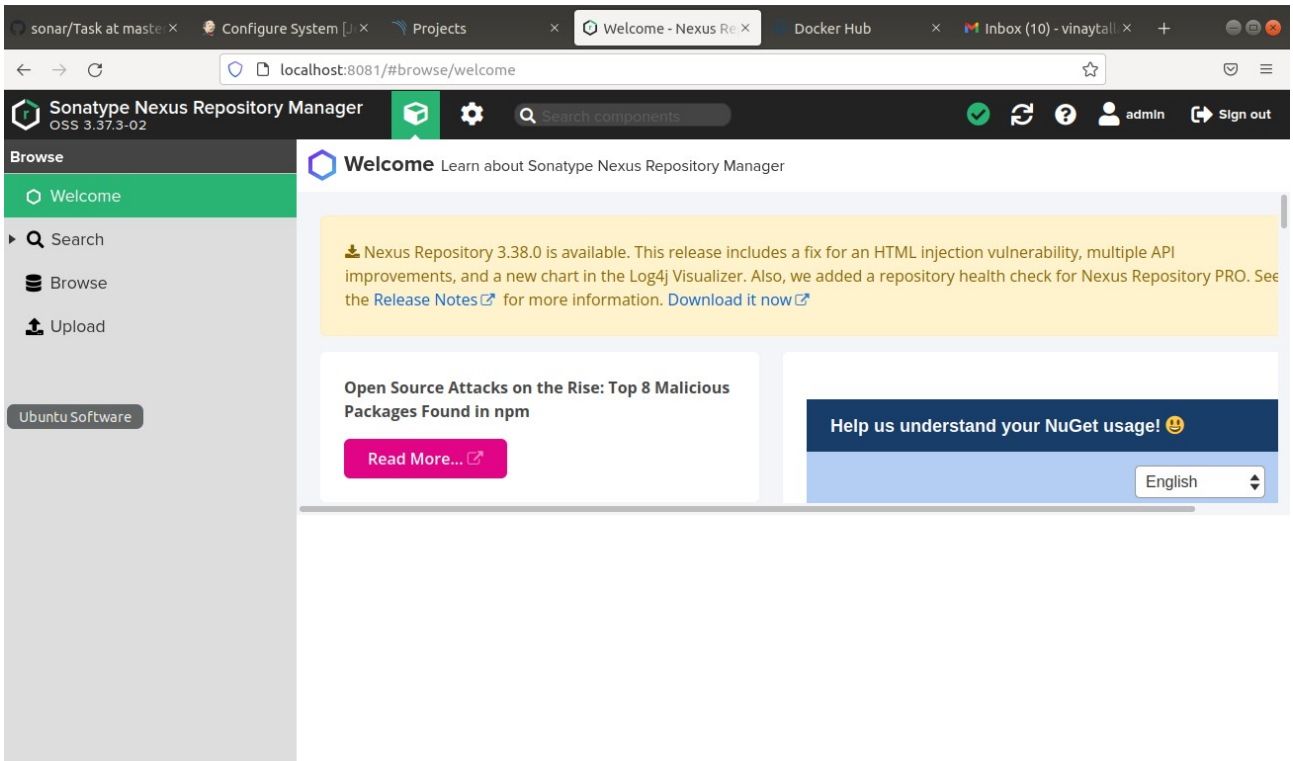
Installing Nexus :

```
root@bandaru-VirtualBox:/home/bandaru/Downloads# ls
apache-tomcat-9.0.59          esql-plugin-2.3.5.jar        sample.bar                    sonar-scanner-4.7.0.2747-linux
apache-tomcat-9.0.59.tar.gz   jenkins.war                  sonar-msgflow-plugin-1.1.3.jar sonar-scanner-cli-4.7.0.2747-linux.zip
Dev                           nexus-3.37.3-02              sonarqube-6.7.7               sonatype-work
docker                        nexus-3.37.3-02-unix.tar.gz  sonarqube-6.7.7.zip           VBoxGuestAdditions_6.0.4.iso
root@bandaru-VirtualBox:/home/bandaru/Downloads# tar -xvzf nexus-3.37.3-02-unix.tar.gz
```

```
root@bandaru-VirtualBox:/home/bandaru/Downloads# ls
apache-tomcat-9.0.59          esql-plugin-2.3.5.jar        sample.bar                    sonar-scanner-4.7.0.2747-linux
apache-tomcat-9.0.59.tar.gz   jenkins.war                  sonar-msgflow-plugin-1.1.3.jar sonar-scanner-cli-4.7.0.2747-linux.zip
Dev                           nexus-3.37.3-02              sonarqube-6.7.7               sonatype-work
docker                        nexus-3.37.3-02-unix.tar.gz  sonarqube-6.7.7.zip           VBoxGuestAdditions_6.0.4.iso
root@bandaru-VirtualBox:/home/bandaru/Downloads# cd nexus-3.37.3-02
root@bandaru-VirtualBox:/home/bandaru/Downloads/nexus-3.37.3-02# ls
bin  deploy  etc  lib  NOTICE.txt  OSS-LICENSE.txt  PRO-LICENSE.txt  public  replicator  system
root@bandaru-VirtualBox:/home/bandaru/Downloads/nexus-3.37.3-02# cd bin/
root@bandaru-VirtualBox:/home/bandaru/Downloads/nexus-3.37.3-02/bin# ls
contrib  nexus  nexus.rc  nexus.vmoptions
root@bandaru-VirtualBox:/home/bandaru/Downloads/nexus-3.37.3-02/bin# ./nexus run
and System.err PrintStreams with SLF4JPrintStreams
2022-03-02 15:41:19,139+0530 INFO  [FelixStartLevel] *SYSTEM uk.org.lidalia.sysoutslf4j.context.SysOutOverSLF4J - Redirected System.out and Sys
tem.err to SLF4J for this context
2022-03-02 15:41:19,172+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder - Properties:
2022-03-02 15:41:19,173+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   application-host='0.0.0.0'
2022-03-02 15:41:19,173+0530 INFO  [FelixStartLevel] *SYSTEM org.sonatype.nexus.bootstrap.ConfigurationBuilder -   application-port='8081'
```

By default Nexus run at port no. 8081
To see Nexus Interface migrate to browser and Type: http://localhost:8081

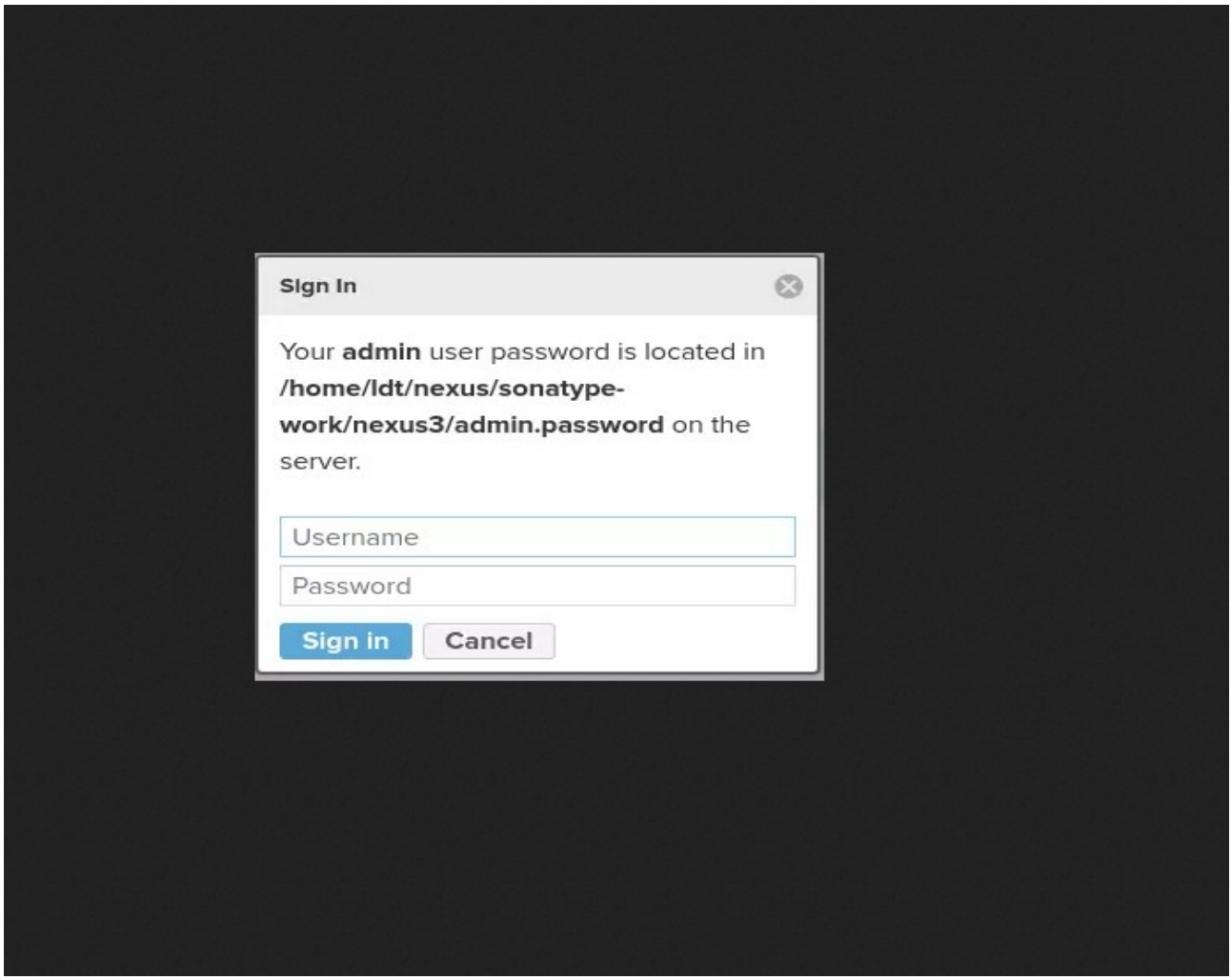
Nexus Interface



To login:

Username: admin

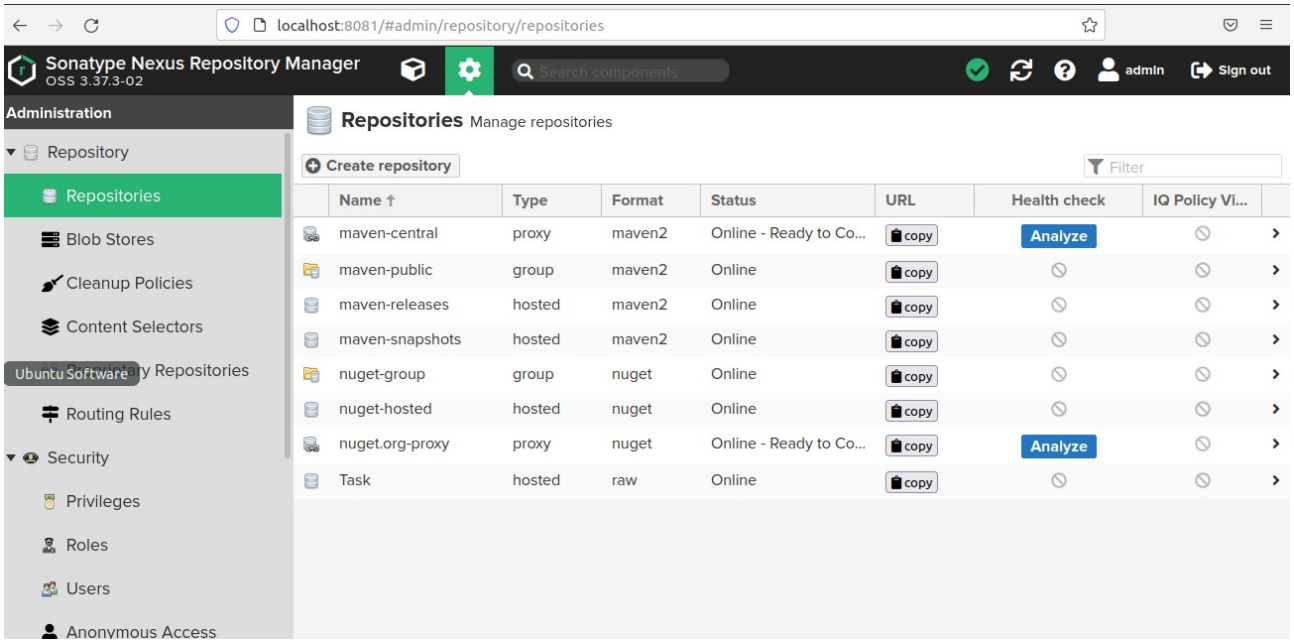
password is obtained by from the path
copy the path from the login interface
and obtain the password using CAT the path
in Terminal



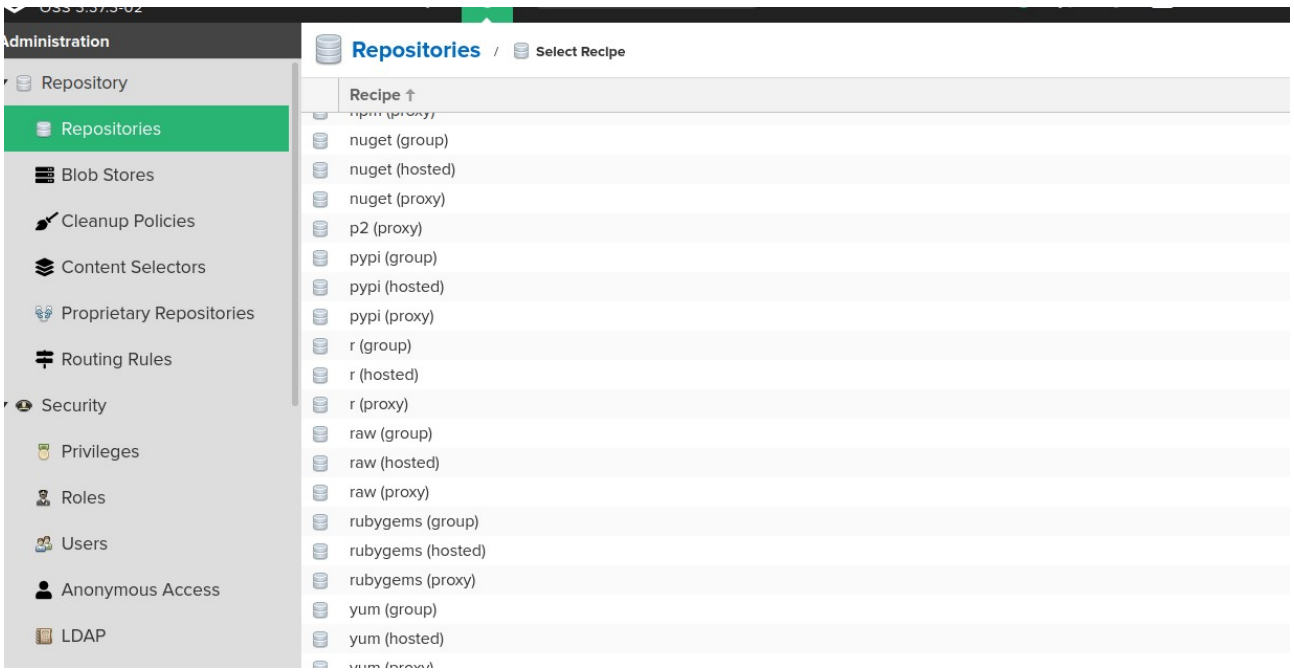
After signing in

Go to settings

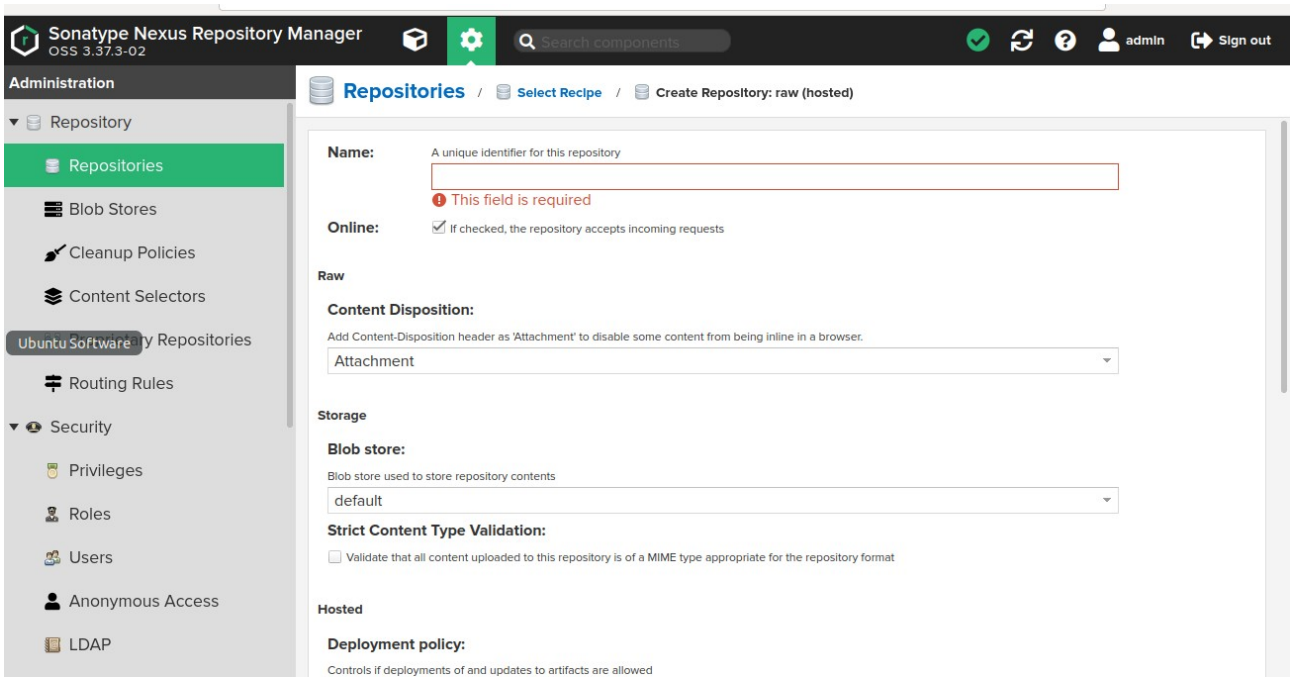
select Create Repository



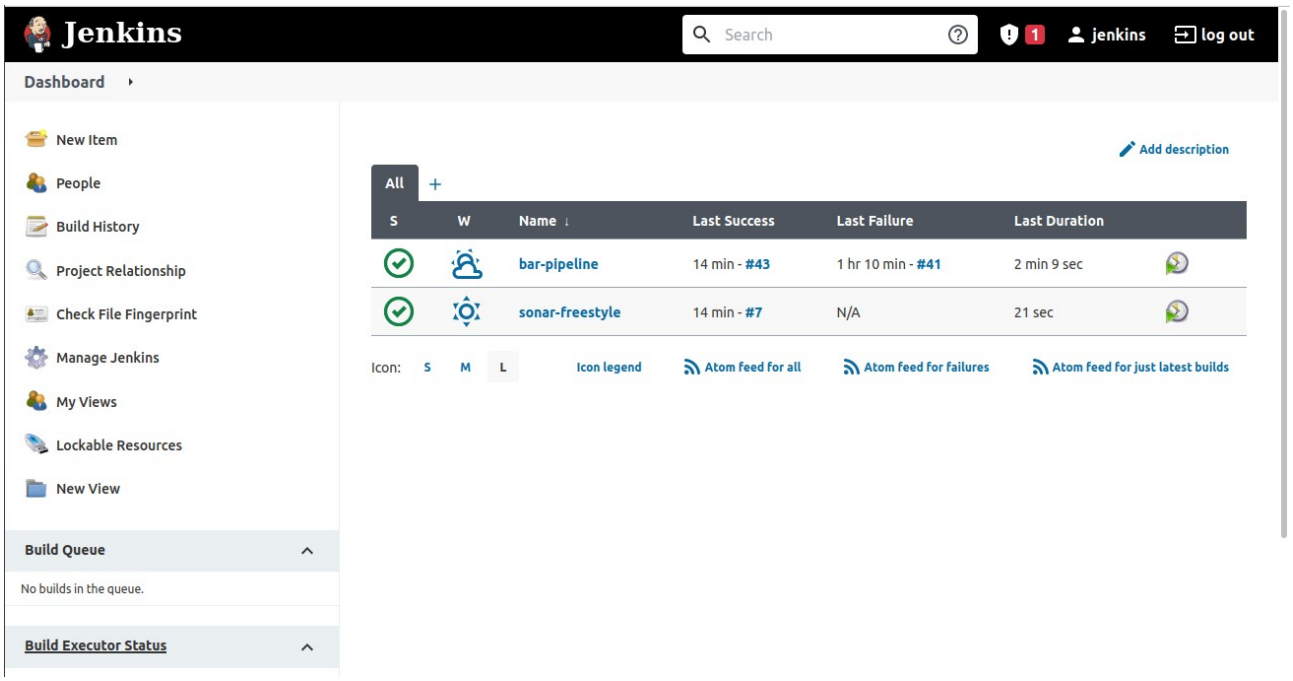
Select Raw(hosted)



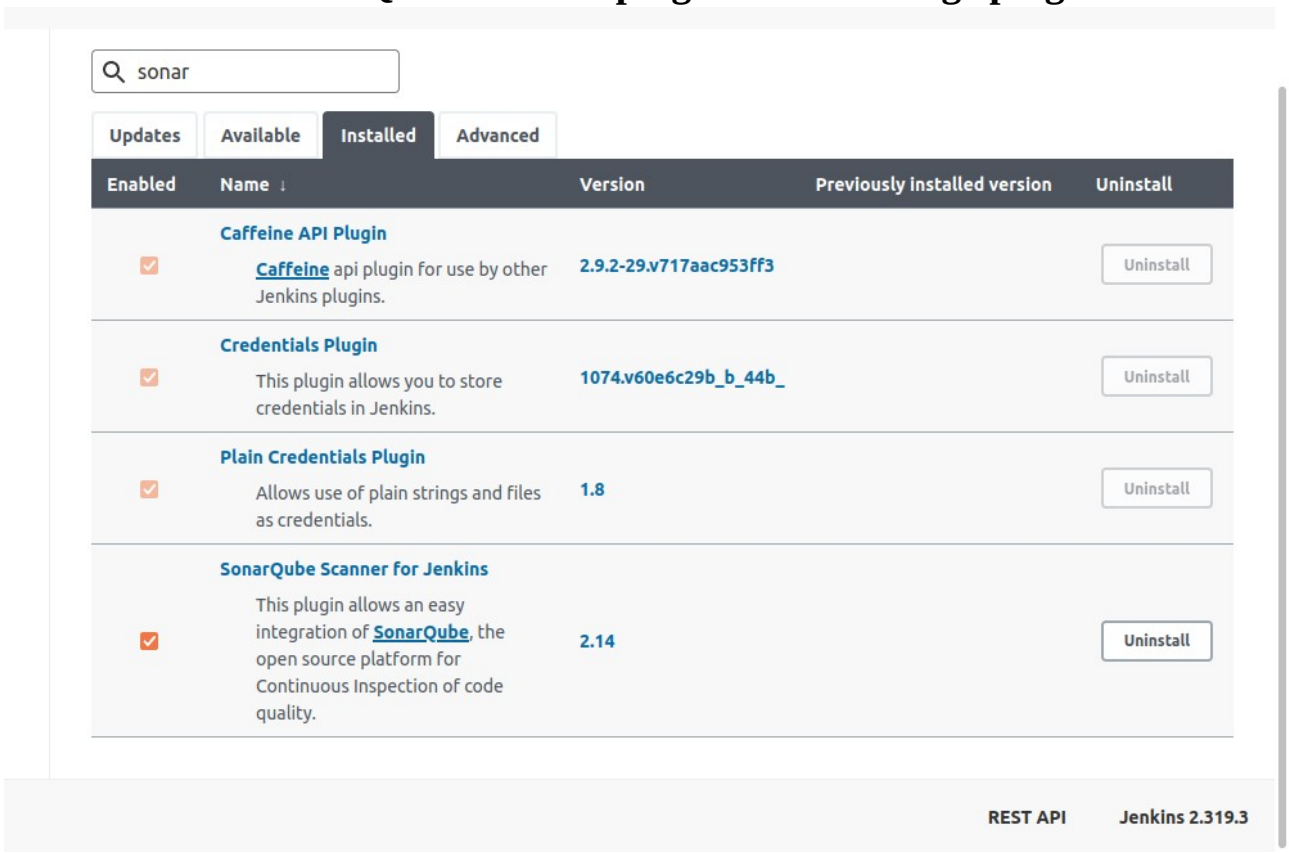
Provide name of Repository and Save the Repository



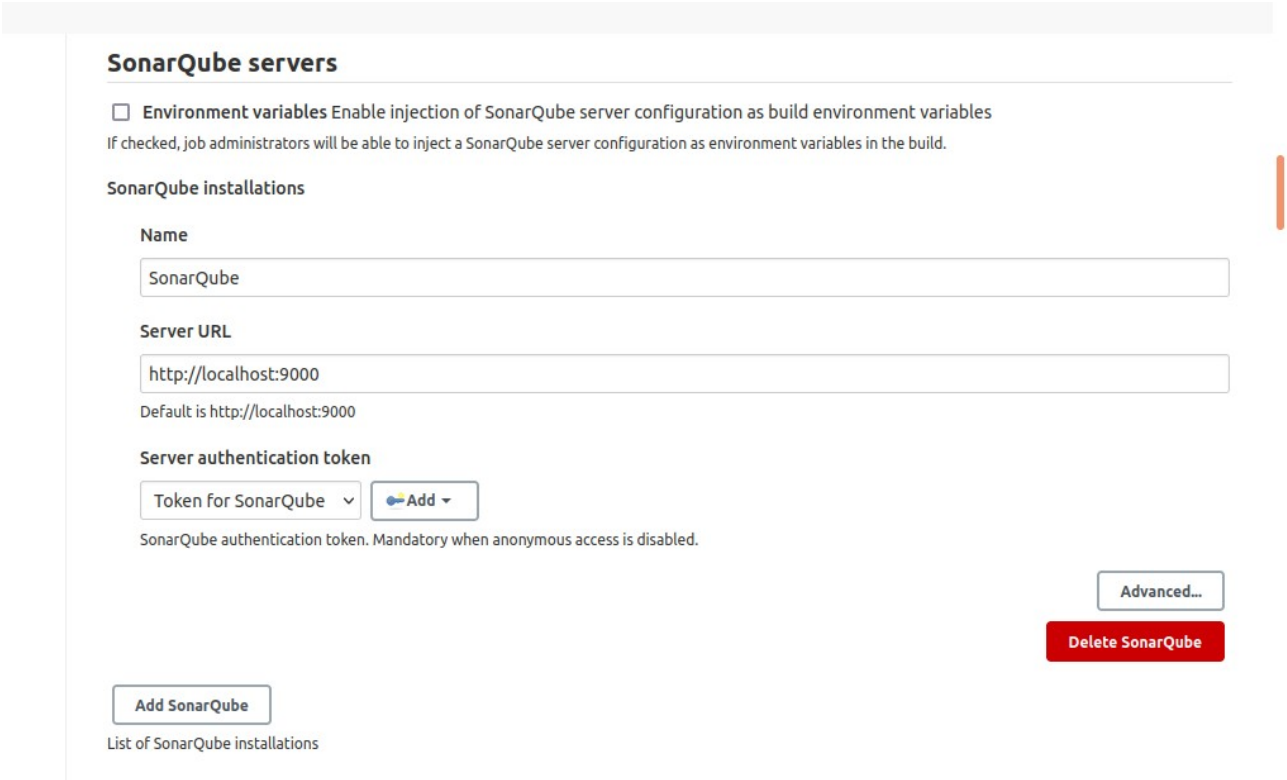
1.Create Two Jobs of Type Pipeline and FreeStyle



Download the SonarQube Scanner plugin under manage plugins



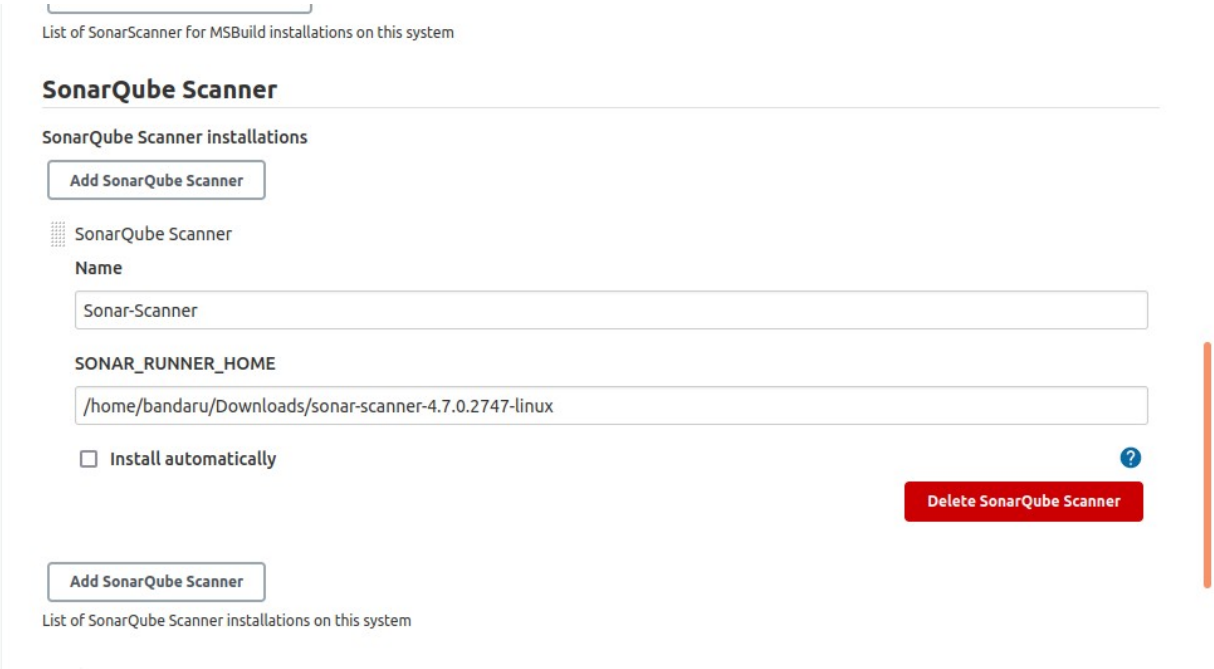
2. Setup SonarQube servers in Configure System(Jenkins)



3. Set up SonarQube Scanner Configuration in Global Tool Configuration(Jenkins):

uncheck the install automatically option

Add sonar scanner home directory or installation path



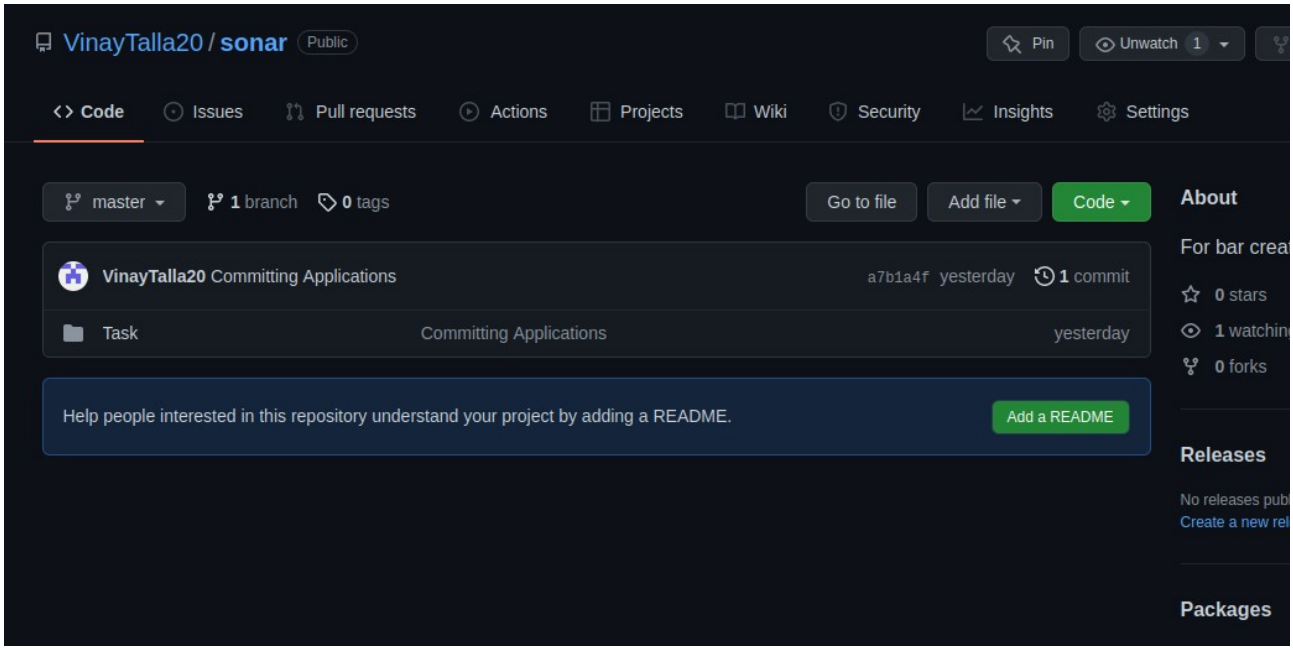
Next Push Application To GitHub:

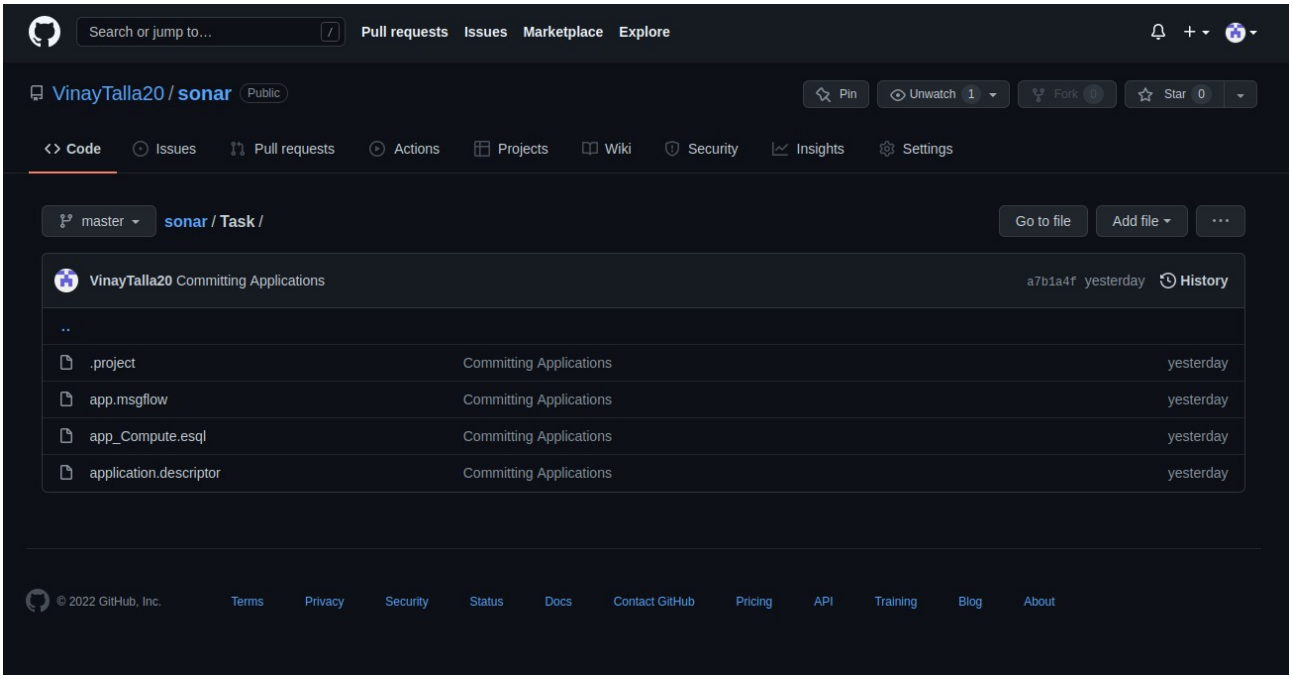
copy all applications to a separate Directory and
make sure Directory contains only Applications which you have copied to the Directory.

push the entire Directory containing Applications

suggestion to push entire Directory:
command : git add directoryname/

providing " / " copies all the files inside the Directory to staging Area

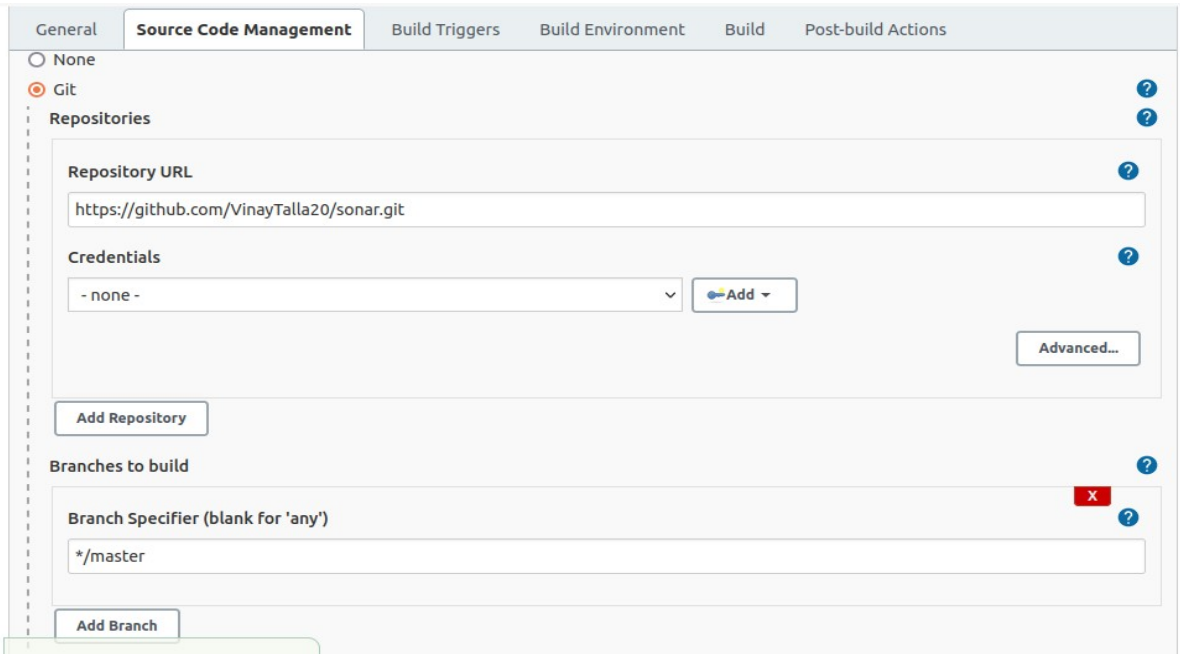

VinayTalla20 / sonar
Public
Pin
Unwatch 1
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
master
1 branch
0 tags
Go to file
Add file
Code
About
For bar creat
VinayTalla20 Committing Applications
a7b1a4f yesterday
1 commit
0 stars
1 watchin
Task
Committing Applications
yesterday
0 forks
Help people interested in this repository understand your project by adding a README.
Add a README
Releases
No releases pub
Create a new rel
Packages


Search or jump to…
Pull requests
Issues
Marketplace
Explore
VinayTalla20 / sonar
Public
Pin
Unwatch 1
Fork 0
Star 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
master
sonar / Task /
Go to file
Add file
VinayTalla20 Committing Applications
a7b1a4f yesterday
History
..
.project
Committing Applications
yesterday
app.msgflow
Committing Applications
yesterday
app_Compute.esql
Committing Applications
yesterday
application.descriptor
Committing Applications
yesterday
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

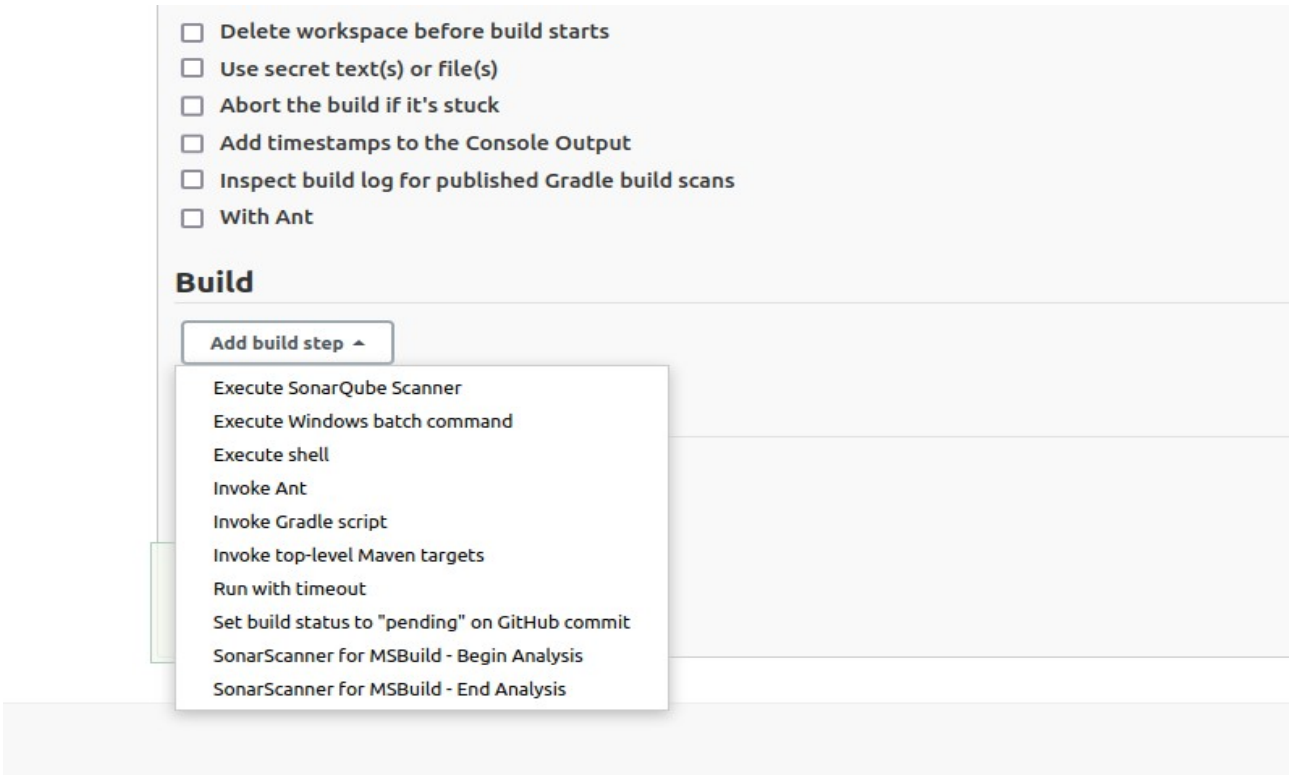
Next Deploying Application in SonarQube for QA

In Free-Style Project next Under Source Code Management

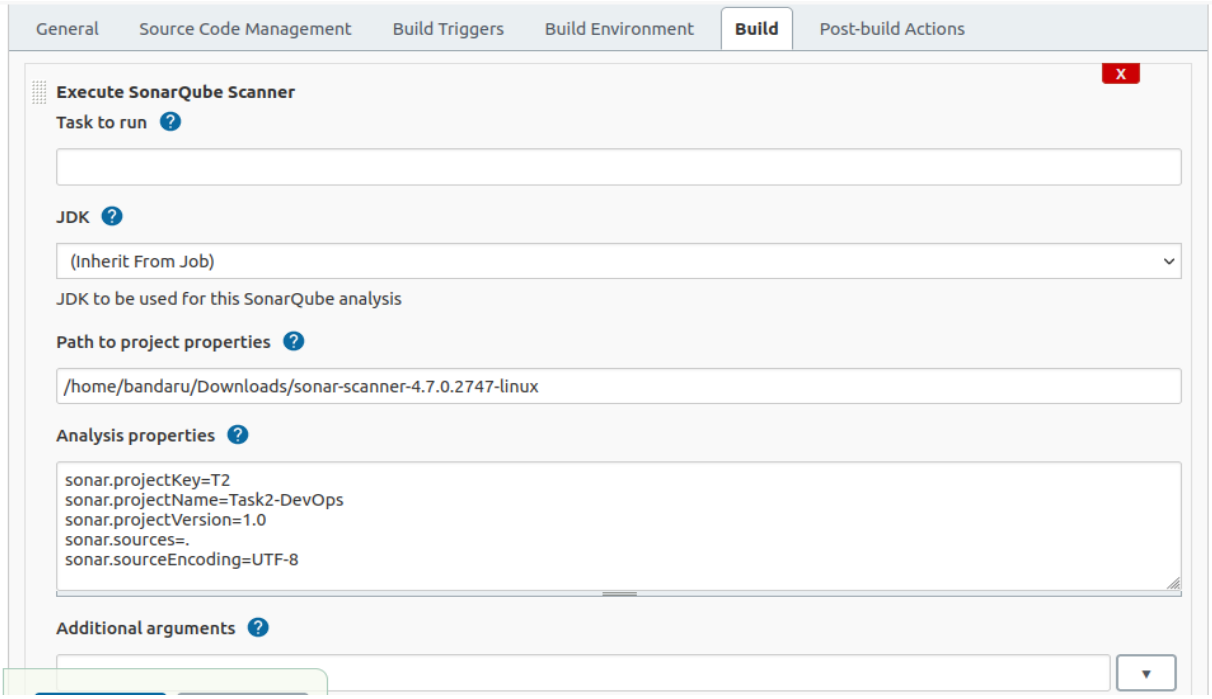
provide Github url and branch name containing the Applications



select "Execute SonarQube Scanner" under Add Build Step



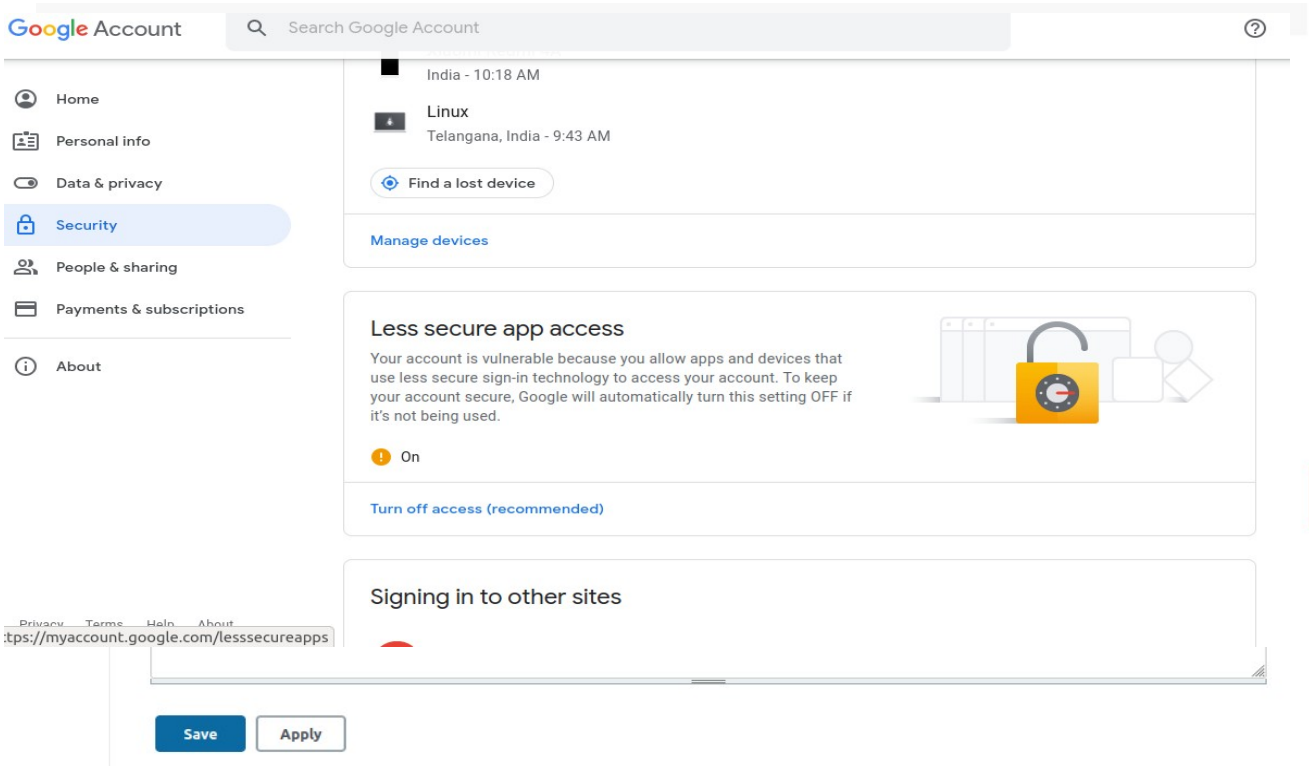
Provide the Home Directory of SonarQube Scanner and Analysis properties as same as below



E-mail configuration

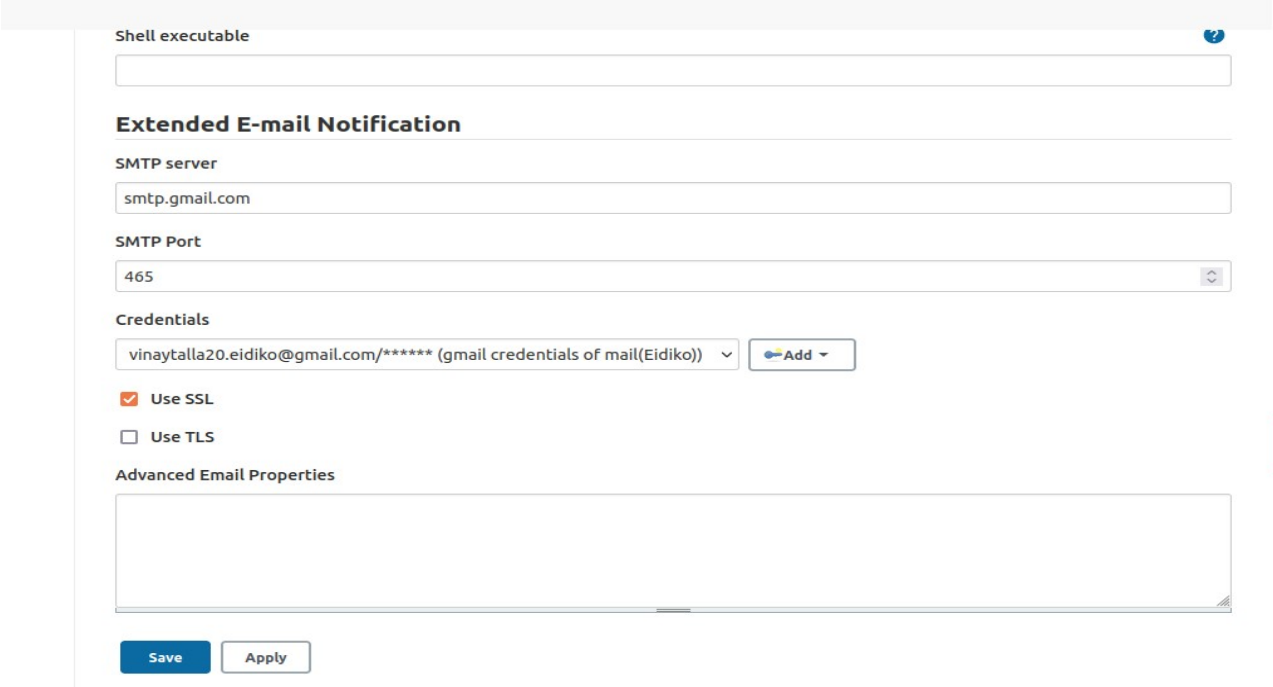
To go Configure System in Manager Jenkins Section
Provide Credentials of Gmail Account

Turn on less Secure App Access in Security settings in your Gmail

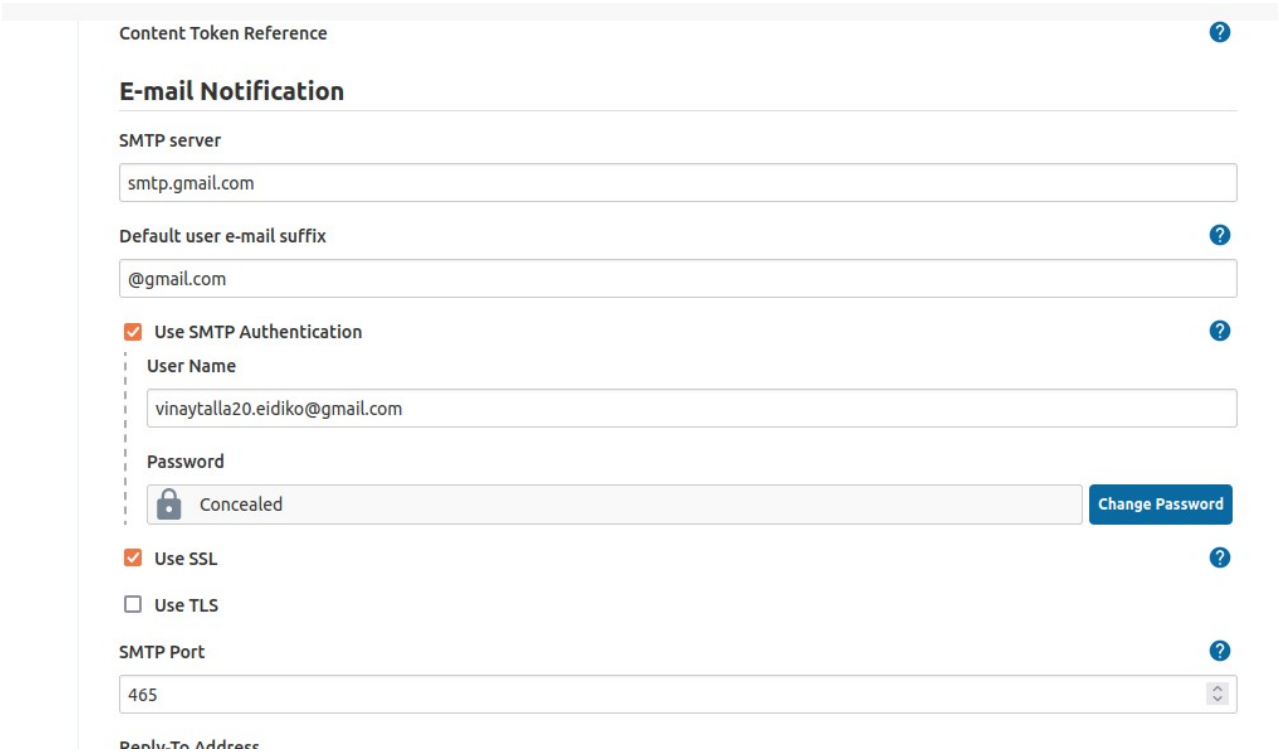


E-mail configuration in Configure System

Select Added Credentials in Credentials Section
and Configure below Details in “Extended E-mail Notification”

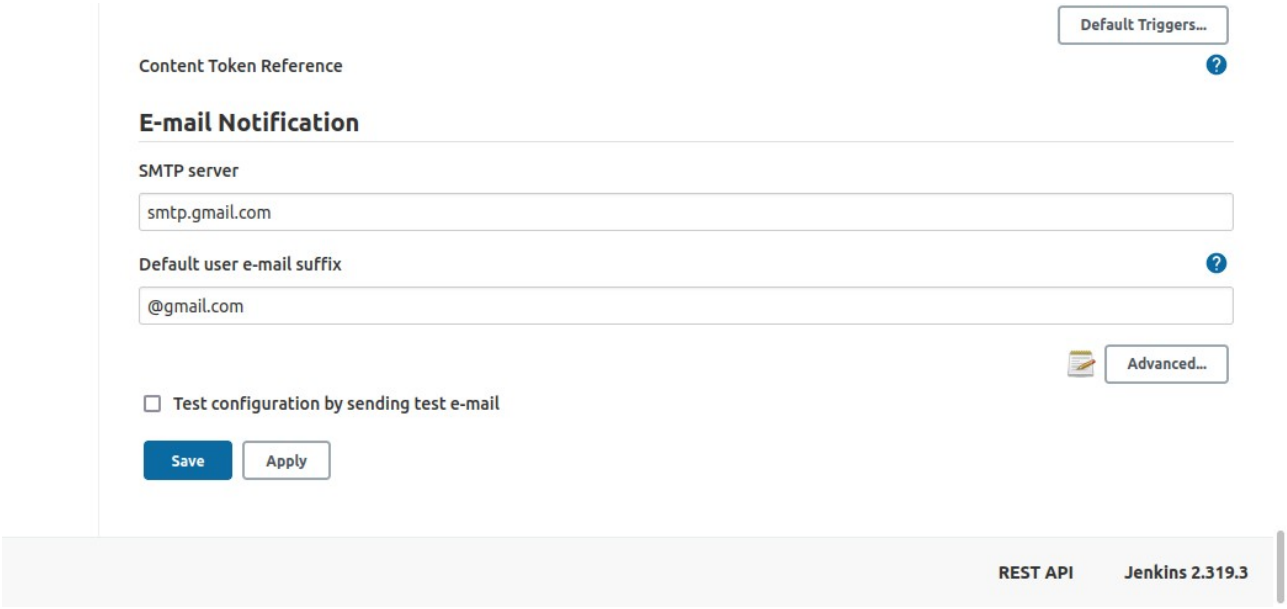


Next E-mail notification
select Advanced and provide below information

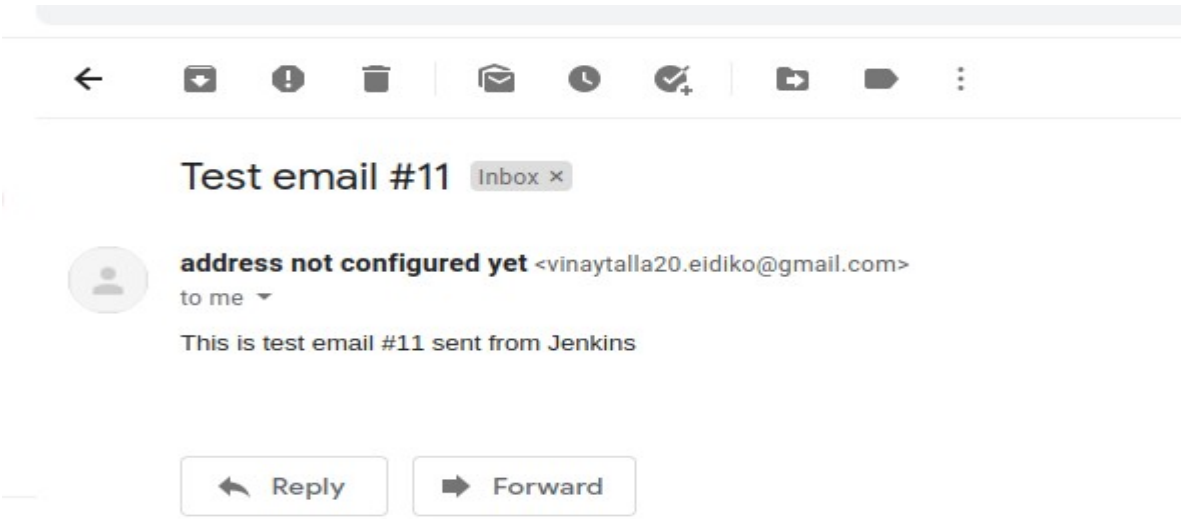


we can Test the Configuration we made to our Gmail Account
Go to E-Mail Notification
provide the below details and also the configure the Advanced option by Providing your gmail credentails

Next go to "Test Configuration by Sending test e-mail"
provide gmail credentials as requied and select Test Configuration



Check your email for this output Test email

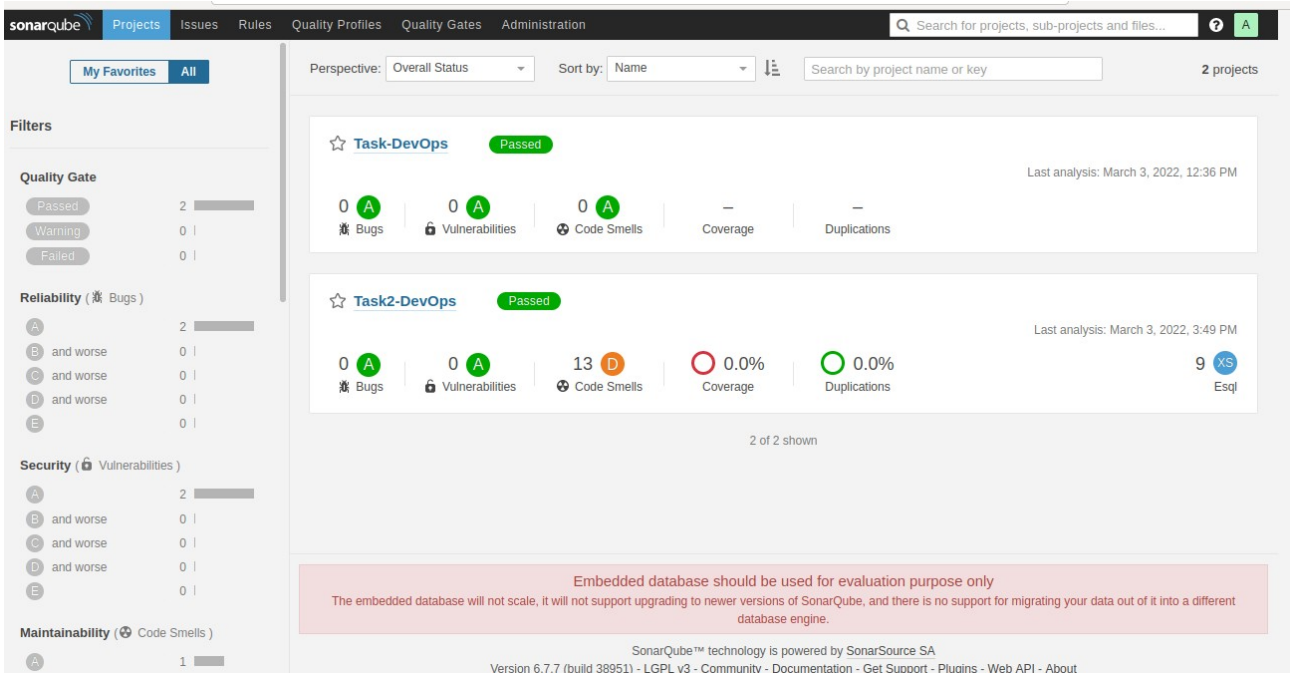


After Completion all the Configuration settings
Now Build The Free-Style Project
This Output in Available in Console Output

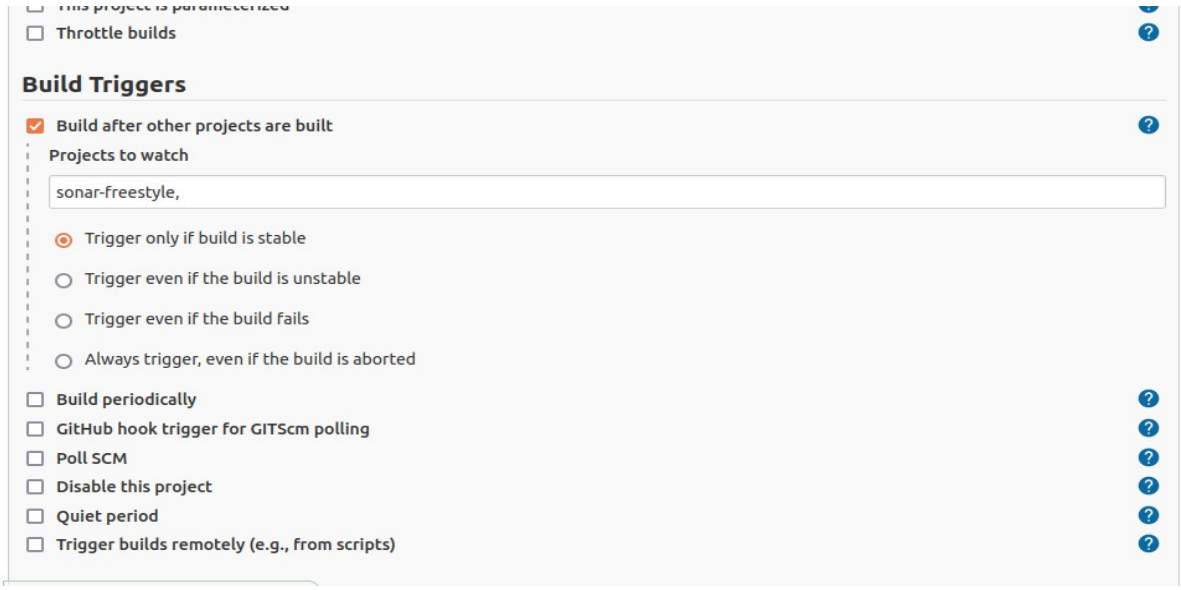
```
INFO: Calculating CPD for 6 files
INFO: CPD calculation finished
INFO: Analysis report generated in 84ms, dir size=34 KB
INFO: Analysis reports compressed in 33ms, zip size=11 KB
INFO: Analysis report uploaded in 535ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://localhost:9000/dashboard/index/T2
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted
analysis report
INFO: More about the report processing at http://localhost:9000/api/ce/task?id=AX9PSmvulCtP7S3Nfy7Q
INFO: Task total time: 8.850 s
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
INFO: Total time: 14.579s
INFO: Final Memory: 13M/50M
INFO: ------------------------------------------------------------------------
Triggering a new build of bar-pipeline #43
Email was triggered for: Always
Sending email for trigger: Always
Sending email to: vinaytalla20.eidiko@gmail.com
Finished: SUCCESS
```

REST API Jenkins 2.319.3

Go To SonarQube interface and you can see the Output under Projects Section



Now Go To Next Project that is
Pipeline project
under Build Triggers
Tick the "Build after other projects are Built"
and Provide the free-Style project name



Under Pipeline

Now its All about Script

Provide below Script

Modify the Script as per Requirements and Credentials and Docker Images Names and Docker Hub usernames, Nexus Repositories Names, Nexus Login Credentials

And also Modify Enviroment Variables as per your Installations paths and Storage Directories

Pipeline

Definition

Pipeline script

Script

```
pipeline {
    agent any
    environment {
        CREDS = credentials('docker')
        BARCREATION = '/opt/ace-11.0.0.11/tools/mqsicreatebar'
        bar = '/home/bandaru/Videos/'

    }


    stages {
        stage ('bar-creation') {
            steps {
                sh 'echo Building Bar'
                git branch: 'master', url: 'https://github.com/VinayTalla20/sonar.git'
              sh 'rm -rf $bar/*.bar'
                sh '$BARCREATION -data $WORKSPACE -b $bar/$JOB_NAME-$BUILD_NUMBER.bar -a Task'
            }
```

Use Groovy Sandbox

Pipeline Syntax

Pipeline script

Script

```
    stages {
        stage ('bar-creation') {
            steps {
                sh 'echo Building Bar'
                git branch: 'master', url: 'https://github.com/VinayTalla20/sonar.git'
              sh 'rm -rf $bar/*.bar'
                sh '$BARCREATION -data $WORKSPACE -b $bar/$JOB_NAME-$BUILD_NUMBER.bar -a Task'
            }
        }

        stage ('Uploading to Nexus'){
            steps {
                sh 'curl -v -u admin:sarasu10 --upload-file $bar/$JOB_NAME-$BUILD_NUMBER.bar http://localhost:8081/repository/Task/'
            }
        }

      stage ('dcoker Build ') {
```

Use Groovy Sandbox

Pipeline

Definition

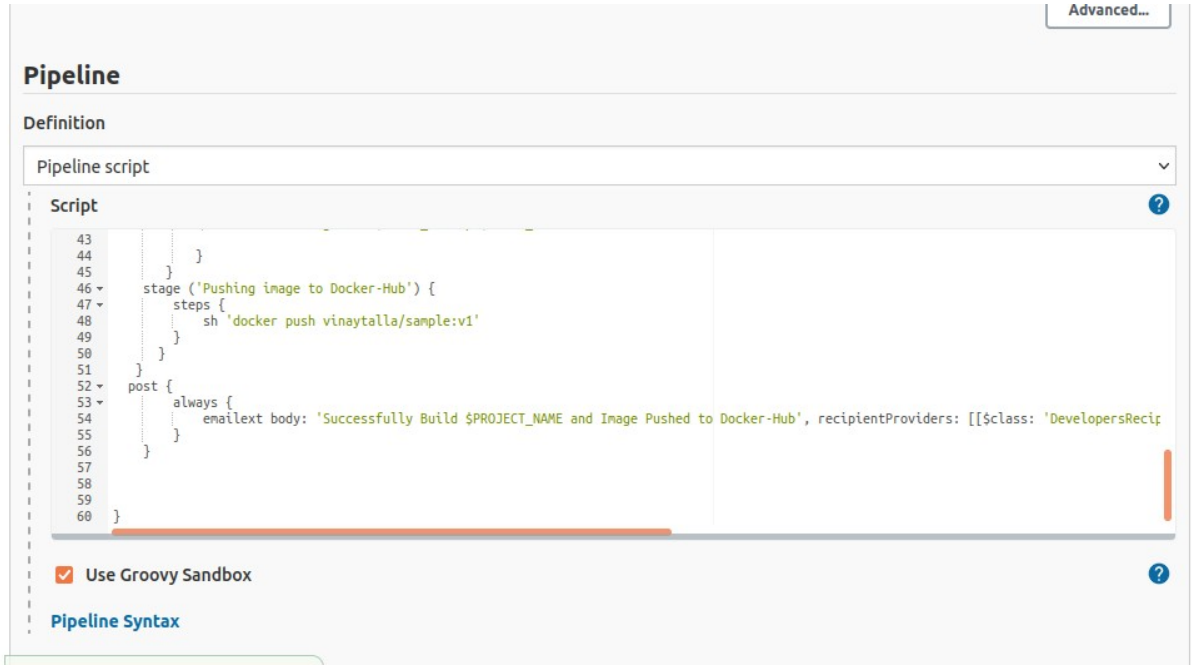
Pipeline script

Script

```
                sh 'cd /home/bandaru/Videos/ &&  docker build -t ace:v1 -f dockerfile . '
            }
        }


        stage ('Tagging Image ') {
            steps {
                sh 'docker tag ace:v1 vinaytalla/sample:v1'

            }
        }
        stage ('Login To Docker Hub') {
            steps {
                sh 'docker login -u $CREDS_USR -p $CREDS_PSW'

            }
        }
    stage ('Pushing image to Docker-Hub') {
```

Use Groovy Sandbox

Pipeline Syntax




```
pipeline {
    agent any
    environment {
        CREDS = credentials('docker')
        BARCREATION = '/opt/ace-11.0.0.11/tools/mqsicreatebar'
         bar = '/home/bandaru/Videos/'

    }


    stages {
        stage ('bar-creation') {
            steps {
                sh 'echo Building Bar'
                git branch: 'master', url: 'https://github.com/VinayTalla20/sonar.git'
              sh 'rm -rf $bar/*.bar'
                sh '$BARCREATION -data $WORKSPACE -b $bar/$JOB_NAME-$BUILD_NUMBER.bar -a Task'
            }
        }

        stage ('Uploading to Nexus'){
            steps {
                sh 'curl -v -u admin:sarasu10 --upload-file $bar/$JOB_NAME-$BUILD_NUMBER.bar http://localhost:8081/repository/Task/'
            }
        }

      stage ('dcoker Build ') {
          steps {
              sh 'cd /home/bandaru/Videos/ &&  docker build -t ace:v1 -f dockerfile . '
          }
      }


        stage ('Tagging Image ') {
            steps {
                sh 'docker tag ace:v1 vinaytalla/sample:v1'

            }
        }
```

```
                sh 'curl -v -u admin:sarasu10 --upload-file $bar/$JOB_NAME-$BUILD_NUMBER.bar http://localhost:8081/repository/Task/'
            }
        }

      stage ('dcoker Build ') {
          steps {
              sh 'cd /home/bandaru/Videos/ &&  docker build -t ace:v1 -f dockerfile . '
          }
      }


        stage ('Tagging Image ') {
            steps {
                sh 'docker tag ace:v1 vinaytalla/sample:v1'

            }
        }
       stage ('Login To Docker Hub') {
            steps {
                sh 'docker login -u $CREDS_USR -p $CREDS_PSW'

            }
        }
    stage ('Pushing image to Docker-Hub') {
        steps {
            sh 'docker push vinaytalla/sample:v1'
        }
      }
   }
  post {
        always {
            emailext body: 'Successfully Build $PROJECT_NAME and Image Pushed to Docker-Hub', recipientProviders: [[$class:
'DevelopersRecipientProvider'], [$class: 'RequesterRecipientProvider']], subject: 'Build Success for $JOB_NAME'
        }
    }



}
```

Plain Text Tab Width: 8 Ln 60, Col 2 INS