Task 1

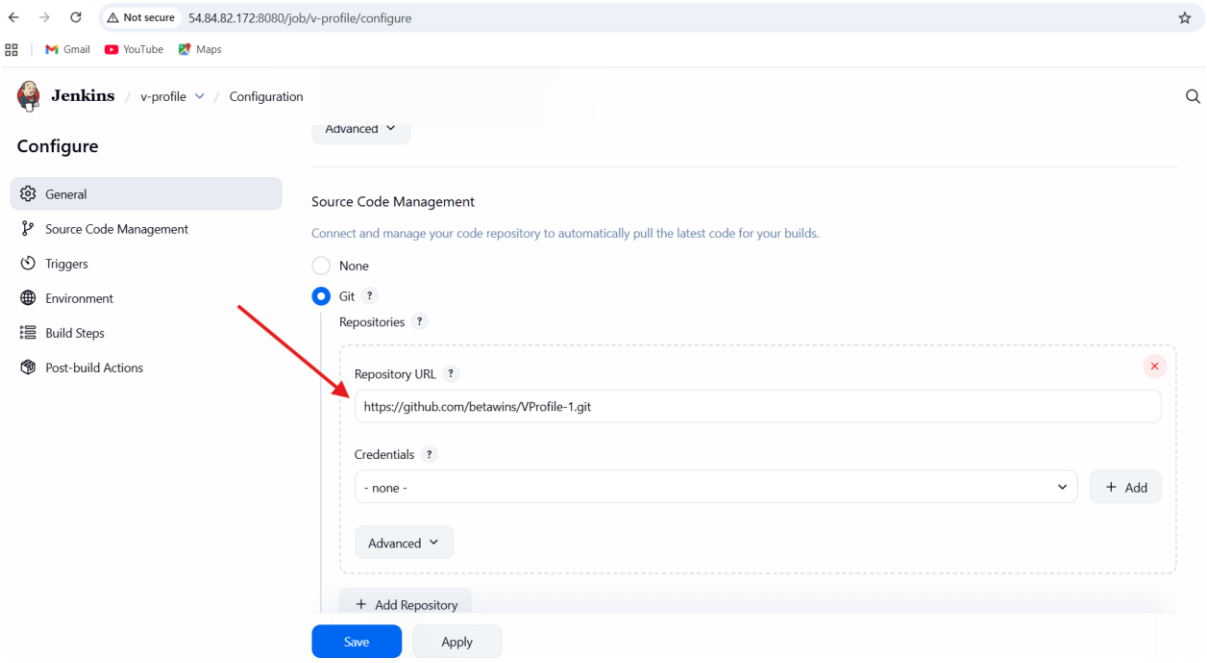
Create one Jenkins job using the below code and create three stages:

- **stage1**: Git clone to download the source code.
- **stage2**: Sonarqube Integration to check the quality of code.
- **stage3**: Slack Integration to send the alerts to slack.

URL: https://github.com/betawins/VProfile-1.git

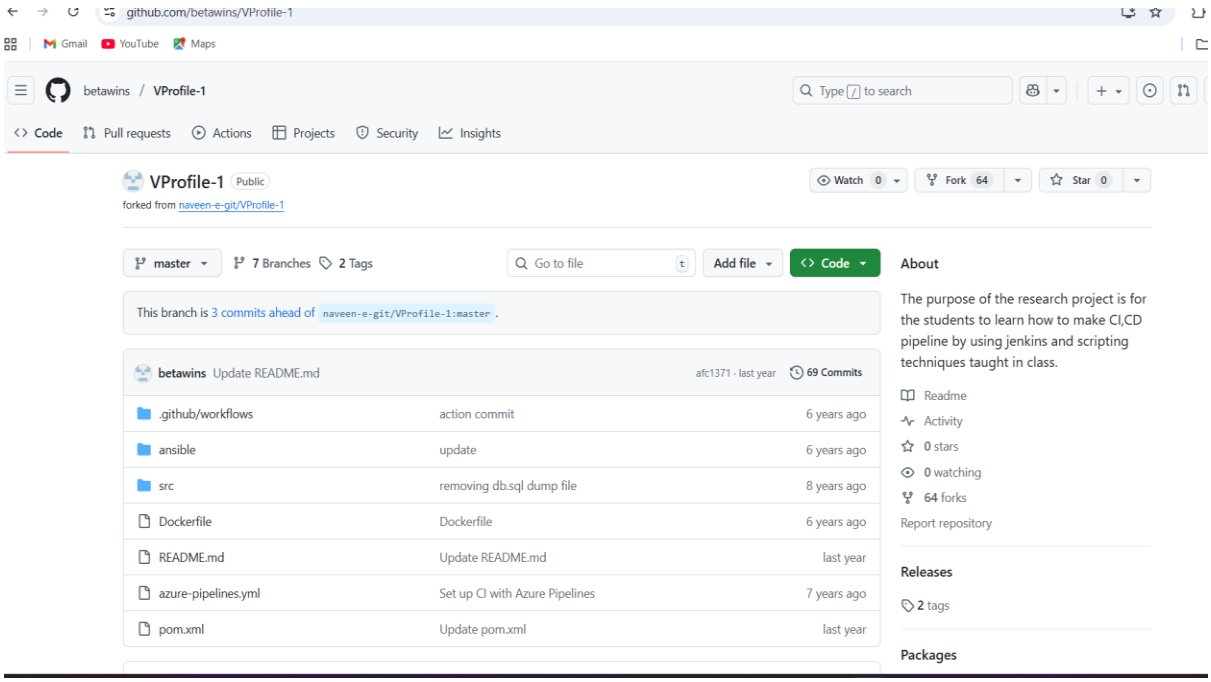
Stage-1:

Click on new item and create an item and in the git give the repository https://github.com/betawins/VProfile-1.git



Go to cd var/lib/Jenkins/workspace/v-profile

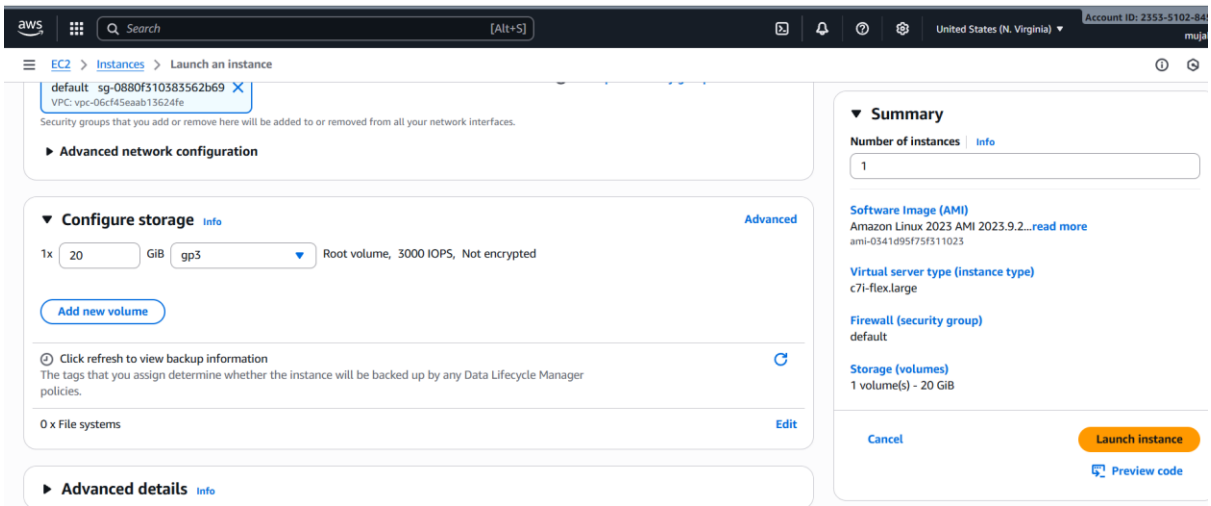
We are successfully clone the repository.




```
[root@ip-172-31-101-218 ~]# cd /var/lib/jenkins
[root@ip-172-31-101-218 jenkins]# ls
config.xml                     jenkins.install.InstallUtil.lastExecVersion     jobs             queue.xml                updates
hudson.model.UpdateCenter.xml  jenkins.install.UpgradeWizard.state             logs             secret.key               userContent
hudson.plugins.git.GitTool.xml jenkins.model.JenkinsLocationConfiguration.xml  nodeMonitors.xml secret.key.not-so-secret users
identity.key.enc               jenkins.telemetry.Correlator.xml                plugins          secrets                  workspace
[root@ip-172-31-101-218 jenkins]# cd /var/lib/jenkins/workspace
[root@ip-172-31-101-218 workspace]# ls
first_job  v-profile
[root@ip-172-31-101-218 workspace]# cd /var/lib/jenkins/workspace/v-profile
[root@ip-172-31-101-218 v-profile]# ls
Dockerfile  README.md  ansible  azure-pipelines.yml  pom.xml  src
[root@ip-172-31-101-218 v-profile]#
```

Stage2:

Launch an instance with the name of sonarqube select instance type as t2 large and and storage as 20 gb.



- yum install java-21 -y
- sudo dnf remove mysql80-community-release

```
[root@ip-172-31-45-79 ~]# sudo dnf remove mysql80-community-release
Dependencies resolved.
================================================================================
 Package                    Architecture        Version          Repository     Size
================================================================================
Removing:
 mysql80-community-release  noarch              el8-5            @System       8.6 k

Transaction Summary
================================================================================
Remove  1 Package

Freed space: 8.6 k
Is this ok [y/N]: y
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Erasing          : mysql80-community-release-el8-5.noarch                 1/1
```

wget https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm

```
[root@ip-172-31-45-79 ~]# wget https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm
--2025-10-22 12:49:16--  https://dev.mysql.com/get/mysql80-community-release-el9-1.noarch.rpm
Resolving dev.mysql.com (dev.mysql.com)... 23.207.138.29, 2600:1408:9000:683::2e31, 2600:1408:9000:698::2e31
Connecting to dev.mysql.com (dev.mysql.com)|23.207.138.29|:443... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://repo.mysql.com//mysql80-community-release-el9-1.noarch.rpm [following]
--2025-10-22 12:49:17--  https://repo.mysql.com//mysql80-community-release-el9-1.noarch.rpm
Resolving repo.mysql.com (repo.mysql.com)... 23.33.203.94, 2600:1408:c400:1692::1d68, 2600:1408:c400:168d::1d68
Connecting to repo.mysql.com (repo.mysql.com)|23.33.203.94|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10534 (10K) [application/x-redhat-package-manager]
Saving to: 'mysql80-community-release-el9-1.noarch.rpm'

mysql80-community-release-el9-1.noarch.rpm 100%[=============================================================================>]

2025-10-22 12:49:17 (287 MB/s) - 'mysql80-community-release-el9-1.noarch.rpm' saved [10534/10534]
```

sudo dnf install mysql80-community-release-el9-1.noarch.rpm

```
[root@ip-172-31-45-79 ~]# sudo dnf install mysql80-community-release-el9-1.noarch.rpm
Last metadata expiration check: 0:07:39 ago on Wed Oct 22 12:41:56 2025.
Dependencies resolved.
================================================================================
 Package                    Architecture        Version          Reposi
================================================================================
Installing:
 mysql80-community-release  noarch              el9-1            @comma

Transaction Summary
================================================================================
Install  1 Package

Total size: 10 k
Installed size: 5.7 k
Is this ok [y/N]: y
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

- sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023

- sudo dnf clean all
- sudo dnf makecache

```
[root@ip-172-31-45-79 ~]# sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2023
[root@ip-172-31-45-79 ~]# sudo dnf clean all
32 files removed
[root@ip-172-31-45-79 ~]# sudo dnf makecache
Amazon Linux 2023 repository
Amazon Linux 2023 Kernel Livepatch repository
MySQL 8.0 Community Server
MySQL Connectors Community
MySQL Tools Community
Metadata cache created.
```

sudo dnf install mysql-community-server mysql-community-client

```
[root@ip-172-31-45-79 ~]#  sudo dnf install mysql-community-server mysql-community-client
Last metadata expiration check: 0:00:20 ago on Wed Oct 22 12:50:54 2025.
Dependencies resolved.
================================================================================
 Package                          Architecture          Version                Rep
================================================================================
Installing:
 mysql-community-client           x86_64                8.0.44-1.el9           mys
 mysql-community-server           x86_64                8.0.44-1.el9           mys
Installing dependencies:
 mysql-community-client-plugins   x86_64                8.0.44-1.el9           mys
 mysql-community-common           x86_64                8.0.44-1.el9           mys
 mysql-community-icu-data-files   x86_64                8.0.44-1.el9           mys
 mysql-community-libs             x86_64                8.0.44-1.el9           mys

Transaction Summary
================================================================================
Install  6 Packages

Total download size: 59 M
Installed size: 337 M
Is this ok [y/N]: y
Downloading Packages:
```

- systemctl start mysqld
- systemctl status mysqld

```
[root@ip-172-31-45-79 ~]# systemctl start mysqld
[root@ip-172-31-45-79 ~]# systemctl status mysqld
● mysqld.service - MySQL Server
     Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; preset: disabled)
     Active: active (running) since Wed 2025-10-22 12:52:35 UTC; 7min ago
       Docs: man:mysqld(8)
             http://dev.mysql.com/doc/refman/en/using-systemd.html
    Process: 28935 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited, status=0/SUCCESS)
   Main PID: 29006 (mysqld)
     Status: "Server is operational"
      Tasks: 37 (limit: 4536)
     Memory: 472.3M
        CPU: 4.660s
     CGroup: /system.slice/mysqld.service
             └─29006 /usr/sbin/mysqld

Oct 22 12:52:29 ip-172-31-45-79.ec2.internal systemd[1]: Starting mysqld.service - MySQL Server...
Oct 22 12:52:35 ip-172-31-45-79.ec2.internal systemd[1]: Started mysqld.service - MySQL Server.
[root@ip-172-31-45-79 ~]#
```

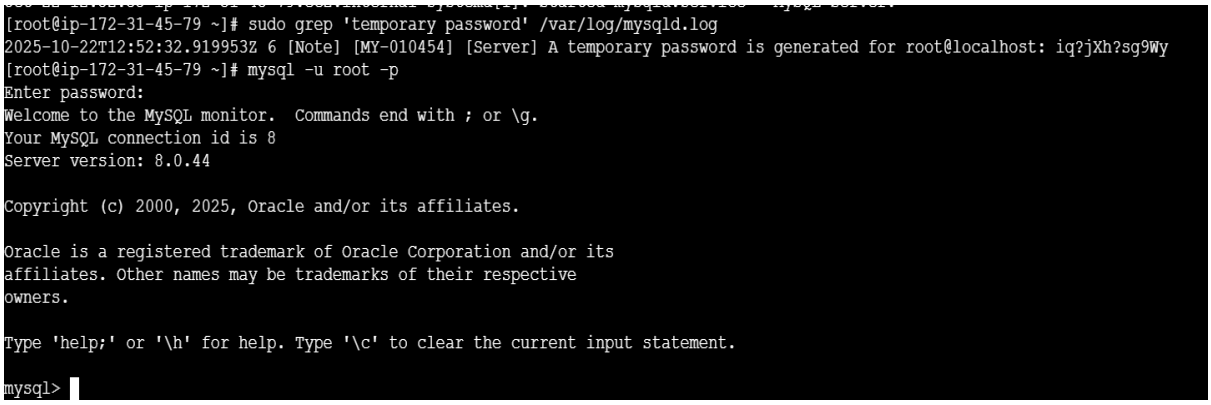
create temporary password to enter into mysql

- sudo grep ‘temporary password’ /var/log/mysqld.log

and login to the sql my using the command

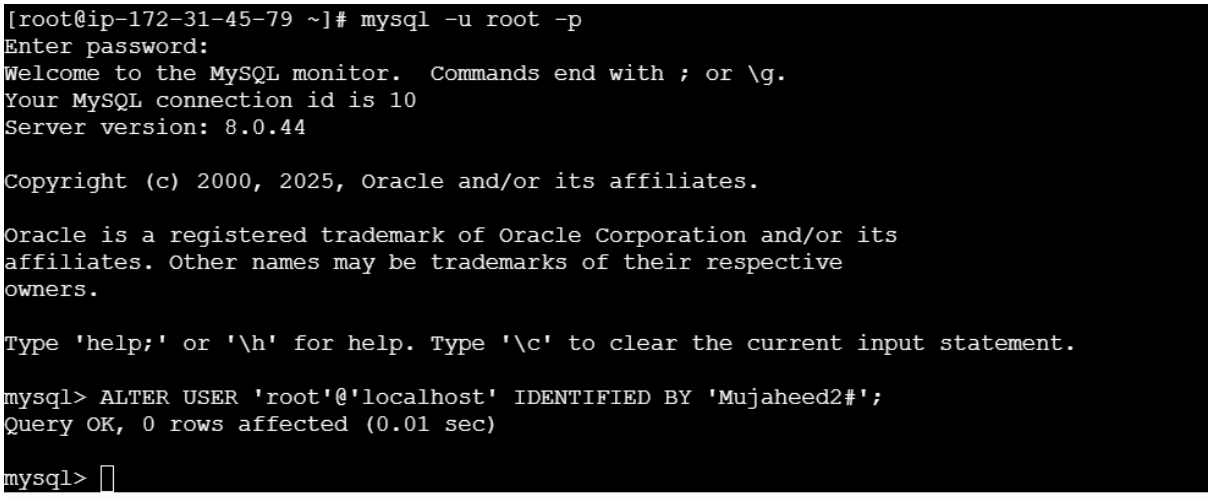
- mysql -u root -p

and paste the password.



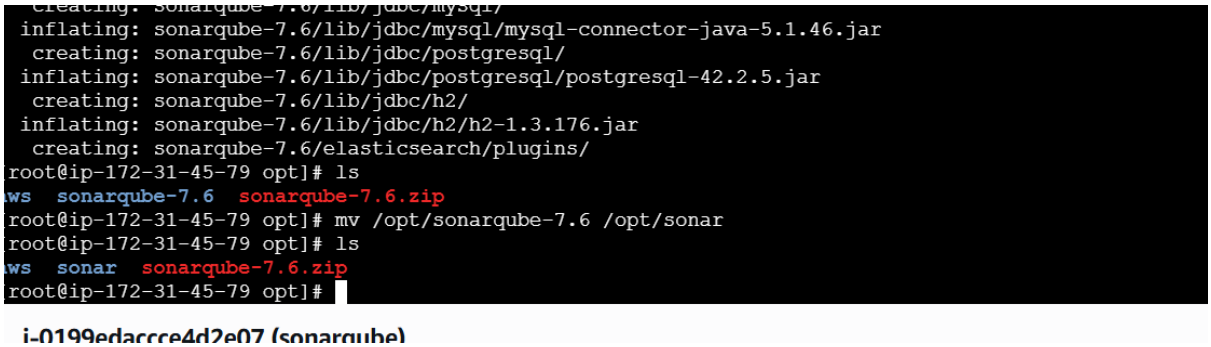
You change the password for mysqld by using the command

- ALTER USER ‘root’@’localhost’ IDENTIFIED BY ‘newpassword’;
- flush privileges;



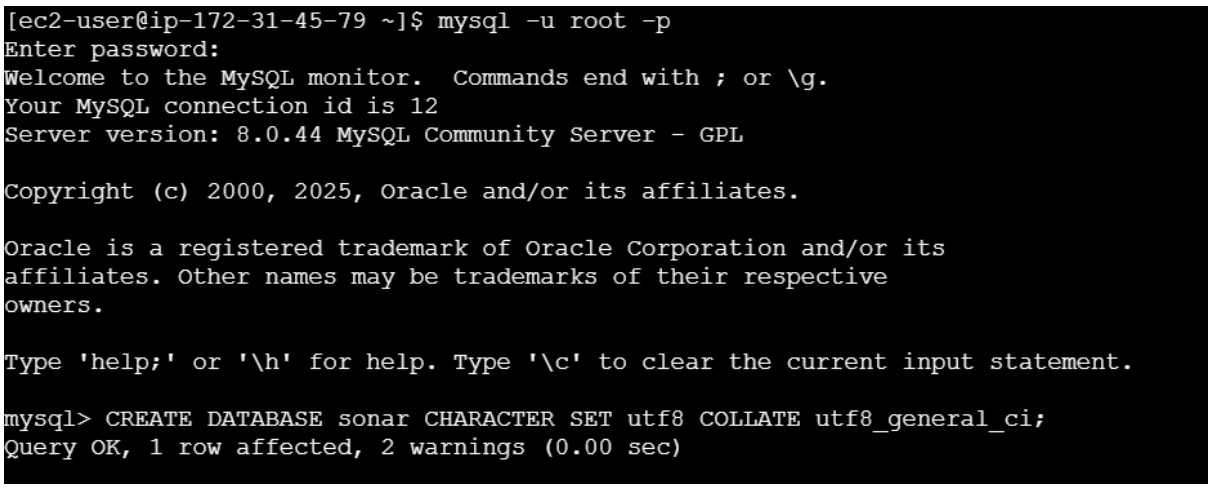
Download and unzip sonarqube.

- cd /opt
- wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.6.zip
- unzip sonarqube-7.6.zip
- mv /opt/sonarqube-7.6 /opt/sonar



Switch to mysql data and create a database

- CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;



Add 2 users in that data base by using

- CREATE USER sonar@localhost IDENTIFIED BY 'Mujaheed2#';

- CREATE USER sonar@’%’ IDENTIFIED BY ‘Mujaheed2#’;

```
mysql> CREATE USER sonar@localhost IDENTIFIED BY 'sonar';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> CREATE USER sonar@localhost IDENTIFIED BY 'sonar';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> CREATE USER sonar@localhost IDENTIFIED BY 'Mujaheed2#';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER sonar@'%' IDENTIFIED BY 'sonar';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> CREATE USER sonar@'%' IDENTIFIED BY 'sonar';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> CREATE USER sonar@'%' IDENTIFIED BY 'Mujaheed2#';
Query OK, 0 rows affected (0.01 sec)

mysql>
```

- GRANT ALL ON sonar.* TO sonar@localhost;
- GRANT ALL ON sonar.* TO sonar@'%';

```
mysql> GRANT ALL ON sonar.* TO sonar@localhost;
Query OK, 0 rows affected (0.01 sec)

mysql> GRANT ALL ON sonar.* TO sonar@'%';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

- use mysql
- show databases;
- SELECT User FROM mysql.user;
- FLUSH PRIVILEGES;
- QUIT

```
mysql> use mysql
Database changed
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sonar              |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> SELECT User FROM mysql.user;
+------------------+
| User             |
+------------------+
| sonar            |
| mysql.infoschema |
| mysql.session    |
| mysql.sys        |
| root             |
| sonar            |
+------------------+
6 rows in set (0.00 sec)
```

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> QUIT
Bye
[ec2-user@ip-172-31-45-79 ~]$
```

- Go to cd /opt/sonar/conf/sonar.properties file and change the properties to

- sonar.jdbc.username=sonar
- sonar.jdbc.password=sonar
- 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

and save it.

```
[ec2-user@ip-172-31-45-79 opt]$ ls
aws  sonar  sonarqube-7.6.zip
[ec2-user@ip-172-31-45-79 opt]$ cd sonar
[ec2-user@ip-172-31-45-79 sonar]$ ls
COPYING  bin  conf  data  elasticsearch  extensions  lib  logs  temp  web
[ec2-user@ip-172-31-45-79 sonar]$ cd conf
[ec2-user@ip-172-31-45-79 conf]$ ls
sonar.properties  wrapper.conf
[ec2-user@ip-172-31-45-79 conf]$ vi sonar.properties
```

```
# Property values can:
# - reference an environment variable, for example sonar.jdbc.url= ${env:SONAR_JDBC_URL}
# - be encrypted. See https://redirect.sonarsource.com/doc/settings-encryption.html

#--------------------------------------------------------------------------------------------------
# DATABASE
#
# IMPORTANT:
# - The embedded H2 database is used by default. It is recommended for tests but not for
#   production use. Supported databases are MySQL, Oracle, PostgreSQL and Microsoft SQLServer.
# - Changes to database connection URL (sonar.jdbc.url) can affect SonarSource licensed products.

# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC user.
# The schema must be created first.
sonar.jdbc.username=sonar
sonar.jdbc.password=Mujaheed2#

#----- Embedded Database (default)
# H2 embedded database server listening port, defaults to 9092
#sonar.embeddedDatabase.port=9092
```

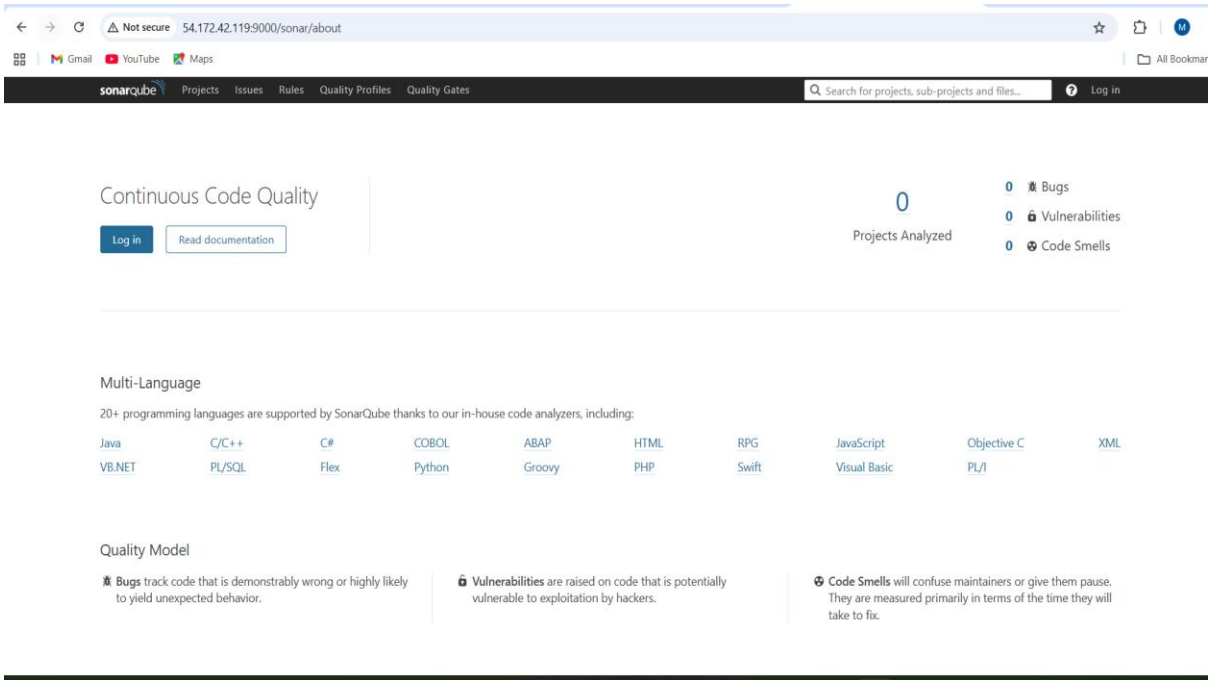
```
#----- MySQL >=5.6 && <8.0
# Support of MySQL is dropped in Data Center Editions and deprecated in all other editions
# Only InnoDB storage engine is supported (not myISAM).
# Only the bundled driver is supported. It can not be changed.
sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false


#----- Oracle 11g/12c
# The Oracle JDBC driver must be copied into the directory extensions/jdbc-driver/oracle/.
```

```
# address will be used for listening on the specified ports.
# By default, ports will be used on all IP addresses associated with the server.
sonar.web.host=0.0.0.0

# Web context. When set, it must start with forward slash (for example /sonarqube).
# The default value is root context (empty value).
sonar.web.context=/sonar
# TCP port for incoming HTTP connections. Default value is 9000.
#sonar.web.port=9000
```

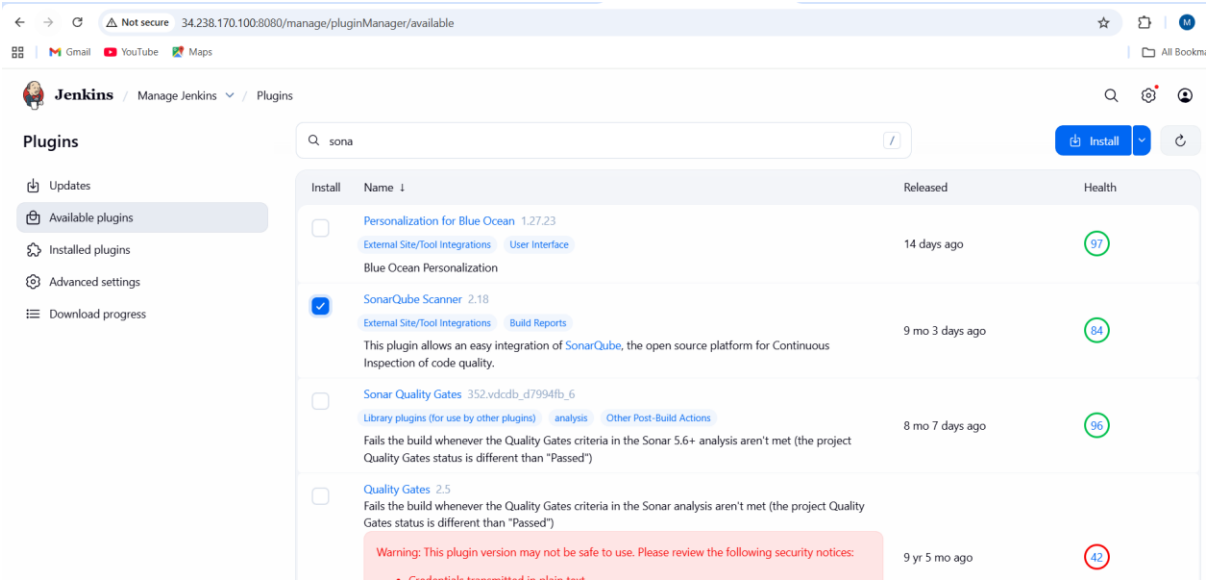
- go to ec2-user
- cd opt/sonar/bin/linux-x86-64

- here do ./sonar.sh start.

```
[ec2-user@ip-172-31-45-79 ~]$ cd /opt
[ec2-user@ip-172-31-45-79 opt]$ ls
aws  containerd  sonar  sonarqube-7.6.zip
[ec2-user@ip-172-31-45-79 opt]$ cd sonar
[ec2-user@ip-172-31-45-79 sonar]$ ls
COPYING  bin  conf  data  elasticsearch  extensions  lib  logs  temp  web
[ec2-user@ip-172-31-45-79 sonar]$ cd bin
[ec2-user@ip-172-31-45-79 bin]$ ls
jsw-license  linux-x86-32  linux-x86-64  macosx-universal-64  windows-x86-32  windows-x86-64
[ec2-user@ip-172-31-45-79 bin]$ cd linux-x86-64
[ec2-user@ip-172-31-45-79 linux-x86-64]$ ls
lib  sonar.sh  wrapper
[ec2-user@ip-172-31-45-79 linux-x86-64]$ ./sonar.sh start
```

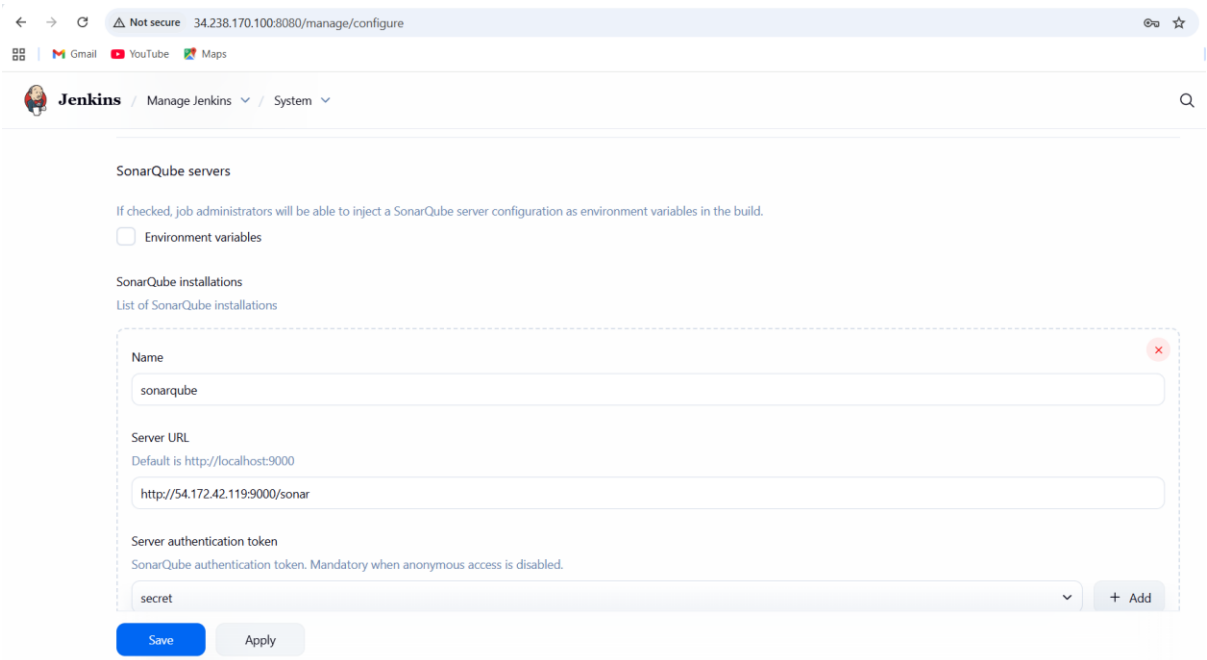


Go to plugins click on available plugins and search sonarqubescanner install that.



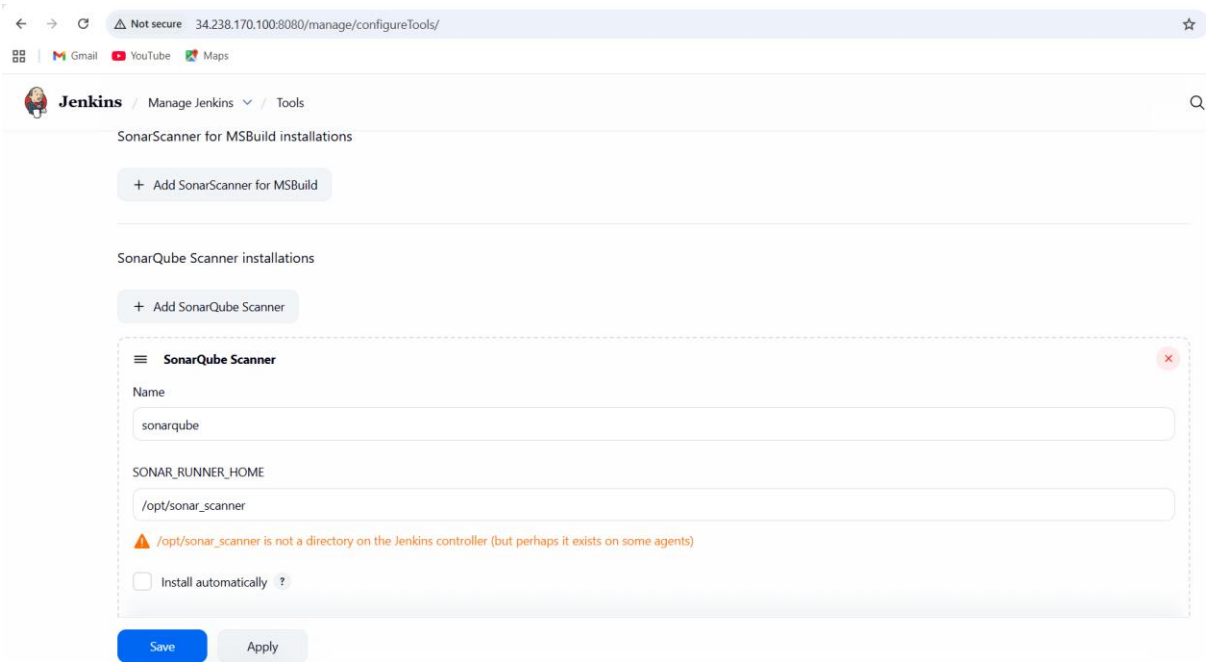
Go to manage Jenkins and go to system in sonarqube installations

- give name
- give your ip address like http://<ip>:9000/sonar
- in the credentials add credentials generate a token in sonarqube add that secret here in credentials like select secret text giving secret and click on save .

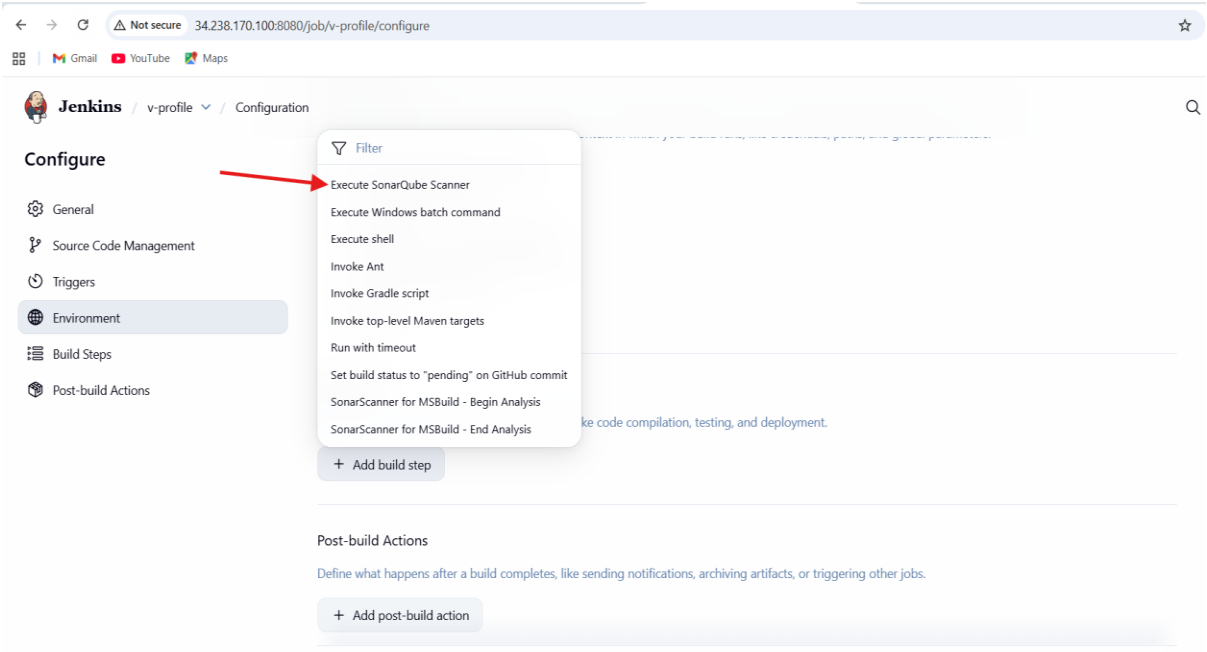


Go to manage Jenkins and go to tools in that add sonar qube scanner details

- give any name
- in path where you installed the sonar qube scannerin Jenkins server like /opt/sonar_scanner save it.



Go to your job and click on configure and click on execute sonarqube scanner



In analysis properties give this code and save it.

sonar.projectKey=Sabear

sonar.projectName=Sabear

sonar.projectVersion=1.0

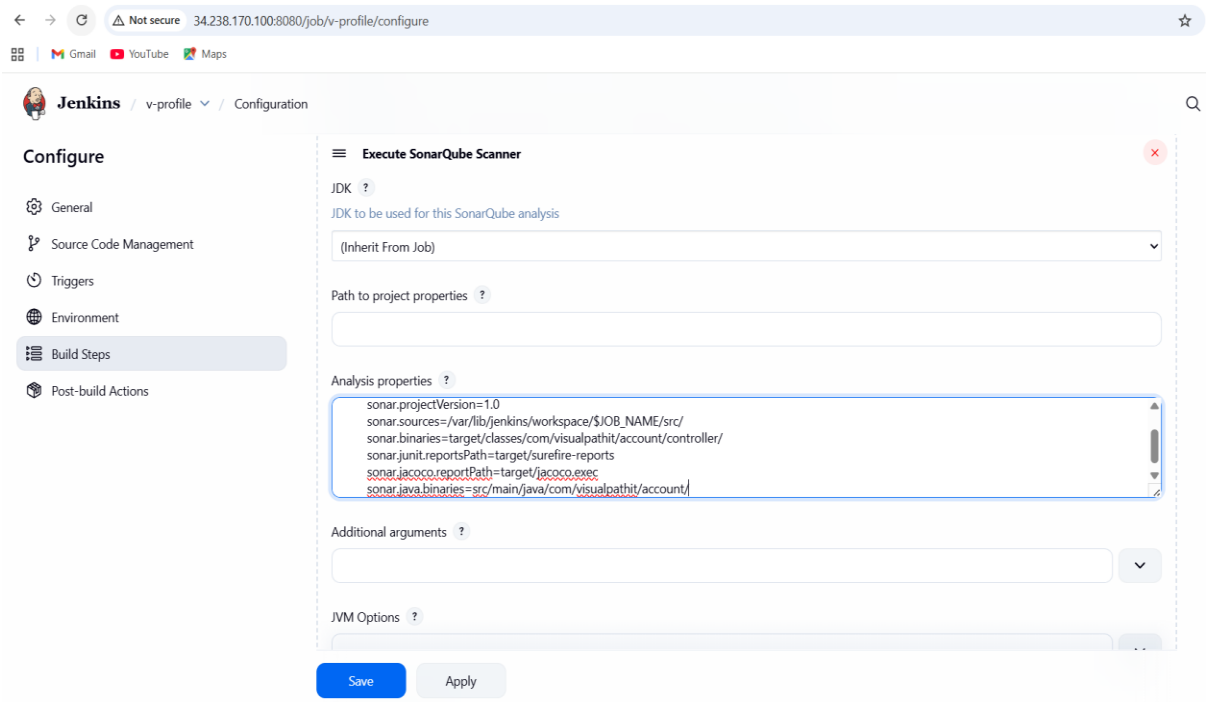
sonar.sources=/var/lib/jenkins/workspace/$JOB_NAME/src/

sonar.binaries=target/classes/com/visualpathit/account/controller/

sonar.junit.reportsPath=target/surefire-reports

sonar.jacoco.reportPath=target/jacoco.exec

sonar.java.binaries=src/main/java/com/visualpathit/account/

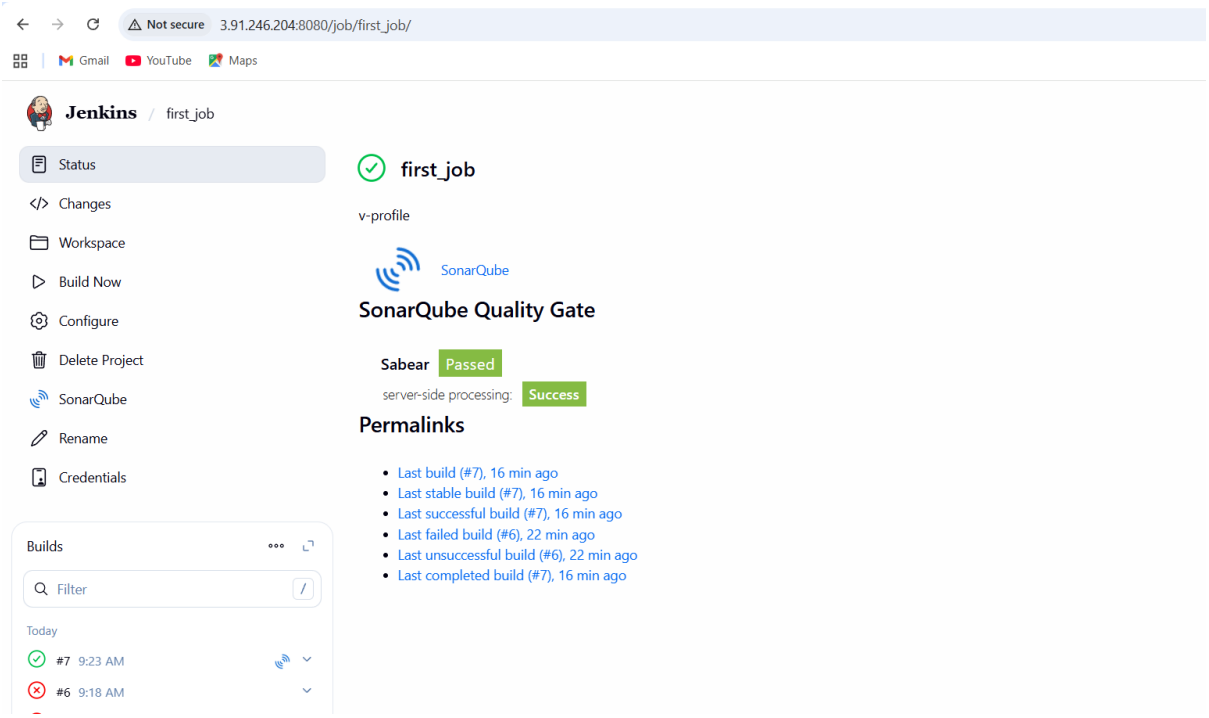


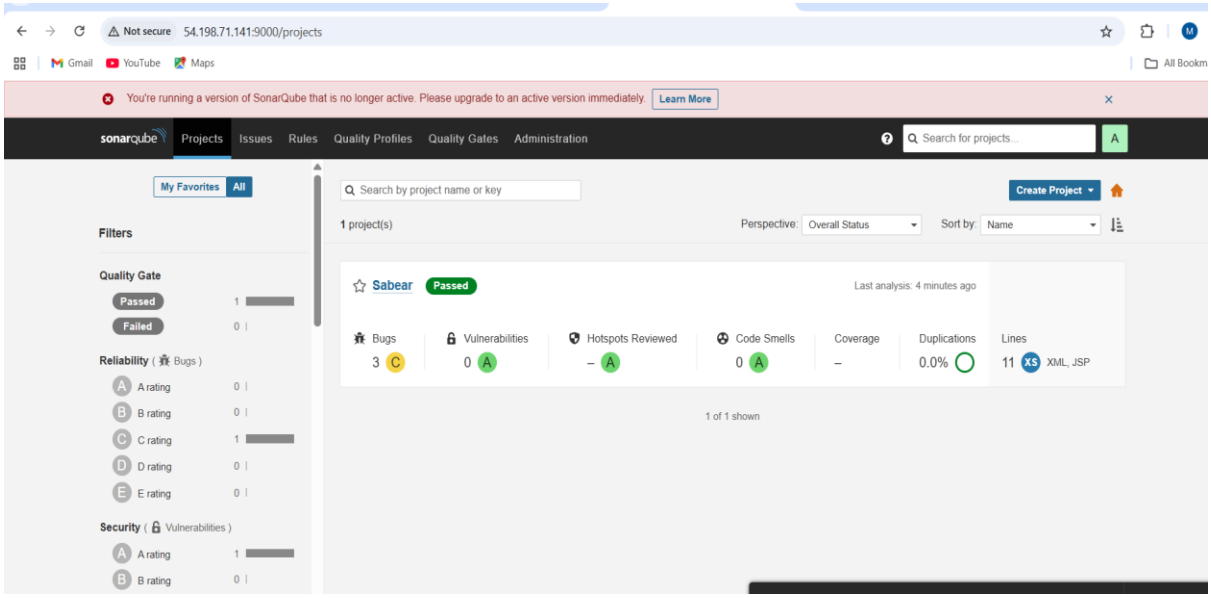
Sonar scanner installation in Jenkins server

- wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip
- unzip sonar-scanner-cli-4.6.2.2472-linux.zip
- mv sonar-scanner-4.6.2.2472-linux /opt/sonar_scanner
- In the Jenkins server you need to change the permissions of sonar by using sudo chown -R ec2-user:ec2-user sonar

```
[root@ip-172-31-32-179 opt]# wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-
p
--2025-10-28 14:25:15--  https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-
Resolving binaries.sonarsource.com (binaries.sonarsource.com)... 99.84.188.45, 99.84.188.21, 99.84.188.
Connecting to binaries.sonarsource.com (binaries.sonarsource.com)|99.84.188.45|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 43099390 (41M) [application/zip]
Saving to: 'sonar-scanner-cli-4.6.2.2472-linux.zip'

sonar-scanner-cli-4.6.2.2472-linu 100%[=============================================================>]

2025-10-28 14:25:15 (77.0 MB/s) - 'sonar-scanner-cli-4.6.2.2472-linux.zip' saved [43099390/43099390]

[root@ip-172-31-32-179 opt]# unzip sonar-scanner-cli-4.6.2.2472-linux.zip
Archive:  sonar-scanner-cli-4.6.2.2472-linux.zip
   creating: sonar-scanner-4.6.2.2472-linux/
   creating: sonar-scanner-4.6.2.2472-linux/jre/
   creating: sonar-scanner-4.6.2.2472-linux/jre/conf/
   creating: sonar-scanner-4.6.2.2472-linux/jre/conf/management/
   creating: sonar-scanner-4.6.2.2472-linux/jre/conf/security/
   creating: sonar-scanner-4.6.2.2472-linux/jre/conf/security/policy/
   creating: sonar-scanner-4.6.2.2472-linux/jre/conf/security/policy/limited/
   creating: sonar-scanner-4.6.2.2472-linux/jre/conf/security/policy/unlimited/
   creating: sonar-scanner-4.6.2.2472-linux/jre/lib/
```

Go to your job and click on build now.

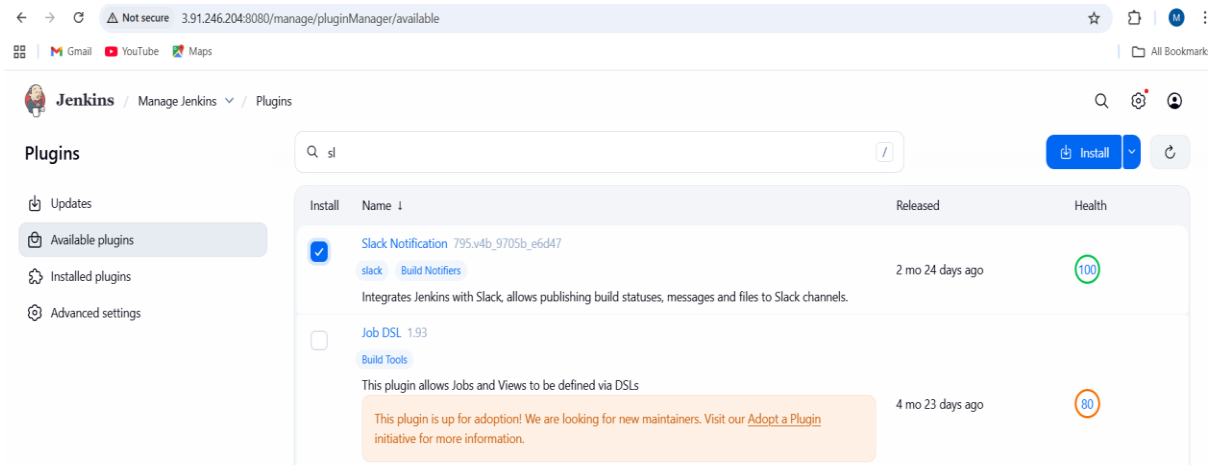




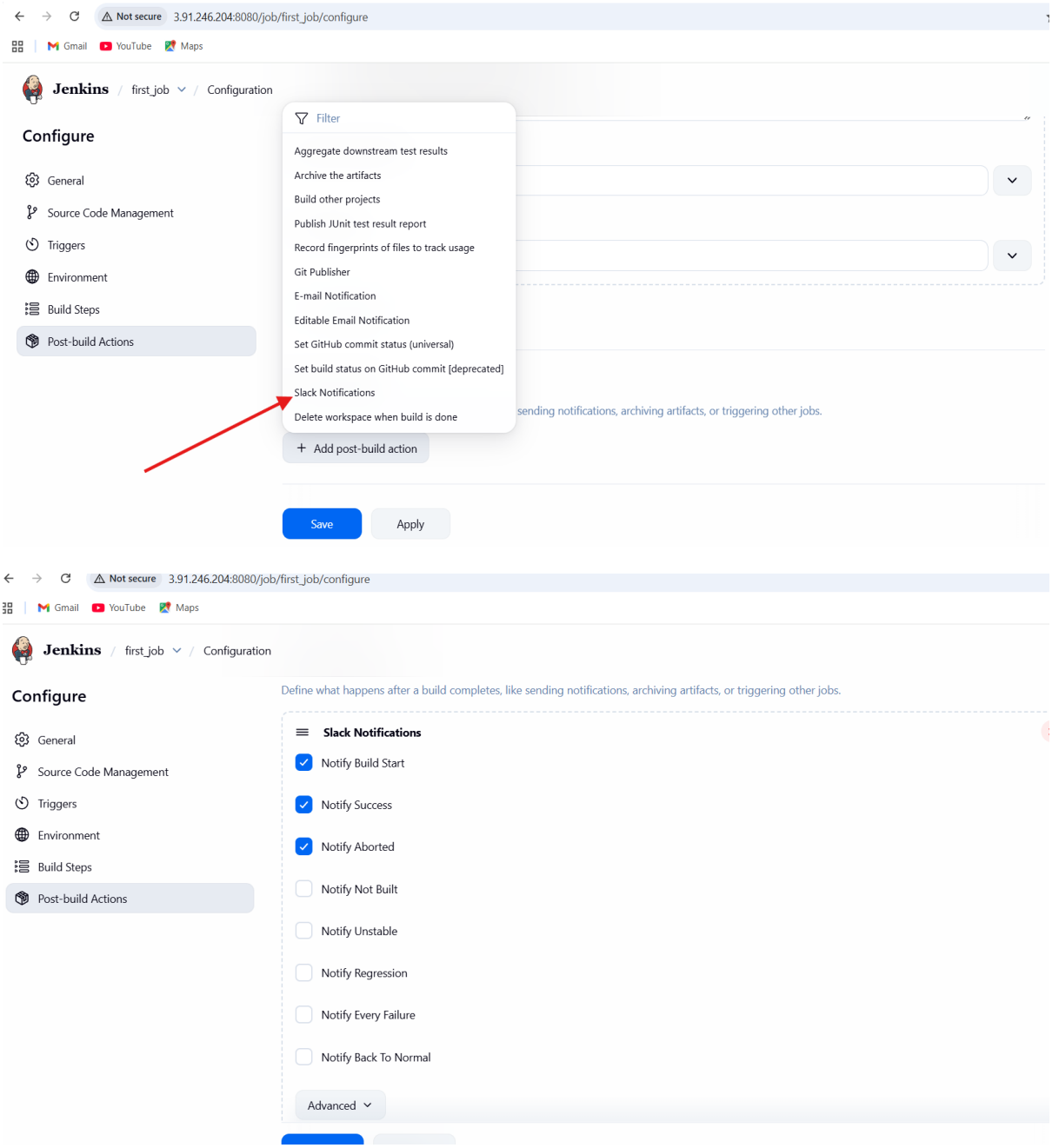
- **stage3**: Slack Integration to send the alerts to slack.

URL: https://github.com/betawins/VProfile-1.git

Go to managed Jenkins and click on plugins download the plugin slack notification.



Go to job and in configuration click on post build actions, select slack notifications.



Task 2

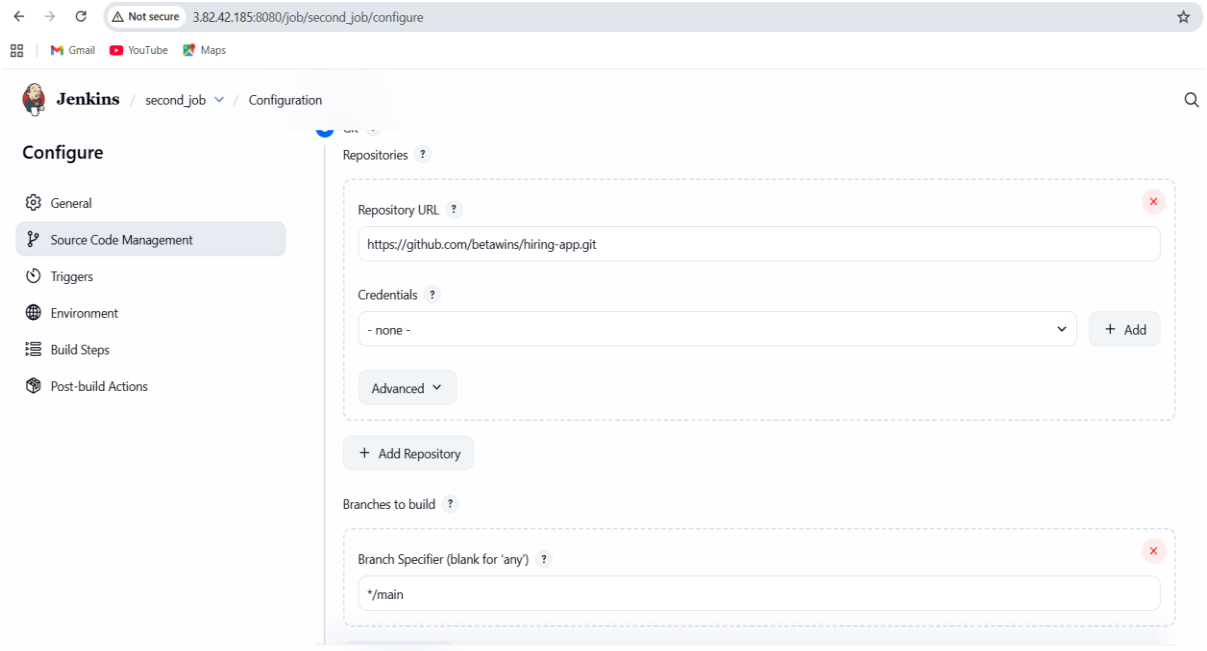
Create one Jenkins job using the below code and create three stages:

- **stage1**: Git clone to download the source code.
- **stage2**: Sonarqube Integration to check the quality of code.
- **stage3**: Slack Integration to send the alerts to slack.

URL: https://github.com/betawins/hiring-app.git

Stage 1:

Add a new job and give the repository in the git field : https://github.com/betawins/hiring-app.git

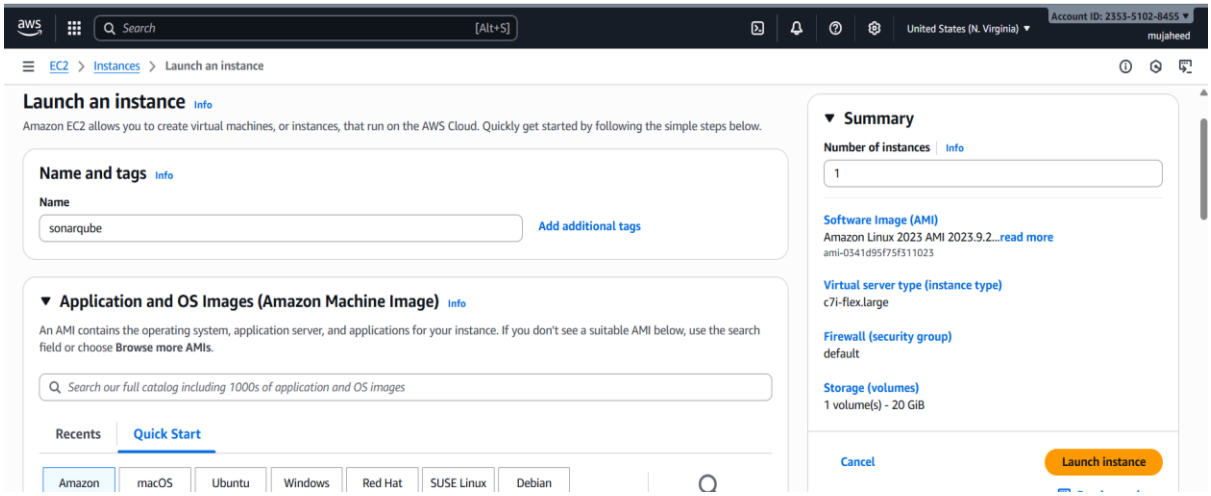


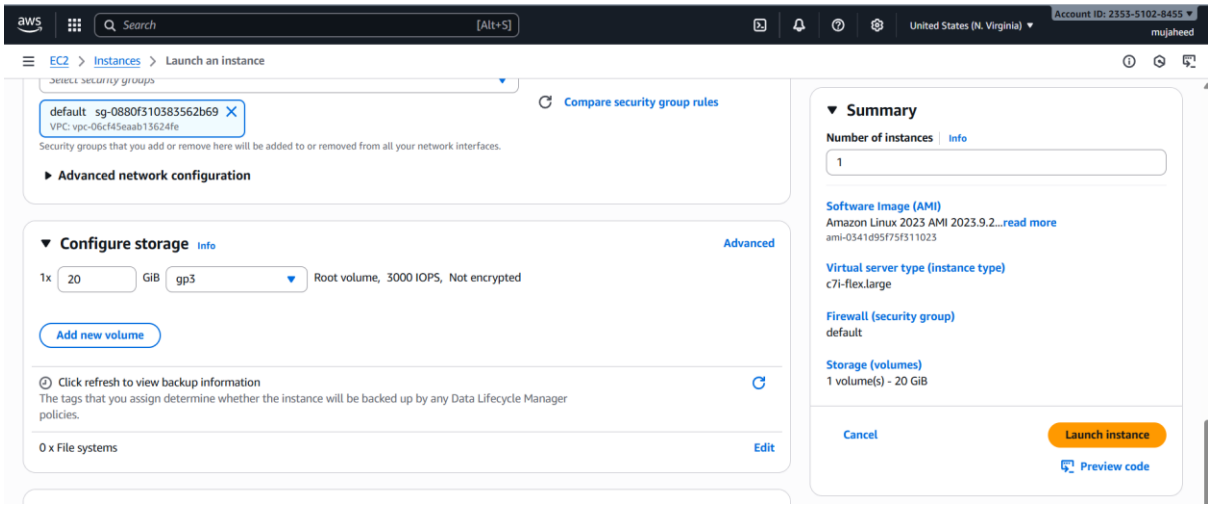
- cd /var/lib/Jenkins/workspace/item

```
[root@ip-172-31-41-85 jenkins]# cd workspace
[root@ip-172-31-41-85 workspace]# ls
first_job  item  second_job
[root@ip-172-31-41-85 workspace]# cd item
[root@ip-172-31-41-85 item]# ls
 Dockerfile   Jenkinsfile   README.md  'Untitled Diagram.drawio'   jenkinsfile-cicd   pom.xml   src
[root@ip-172-31-41-85 item]#
```

Stage 2:

Create an instance named as sonarqube with t2 large and 20 gb of memory.





Install java

- yum install java-1.8*

Add mysql rpm Repository

- yum update
- sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
- sudo yum localinstall mysql57-community-release-el7-11.noarch.rpm
- rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
- sudo yum install mysql-community-server
- sudo systemctl start mysqld.service

Configure the MySQL Root Password

- grep 'temporary' /var/log/mysqld.log

Login to mysql using the default password

- mysql -u root -p

Now replace the default password with a new and strong password

- ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';
- flush privileges;

```
[root@ip-172-31-123-97 ~]# netstat -na | grep 3306
tcp6       0      0 :::3306                 :::*                    LISTEN
[root@ip-172-31-123-97 ~]# sudo systemctl start mysqld.service
[root@ip-172-31-123-97 ~]# netstat -na | grep 3306
tcp6       0      0 :::3306                 :::*                    LISTEN
[root@ip-172-31-123-97 ~]# grep 'temporary' /var/log/mysqld.log
2025-10-23T12:37:14.427678Z 1 [Note] A temporary password is generated for root@localhost: 0rml3qp&yKy/
2025-10-23T12:37:16.842811Z 0 [Note] InnoDB: Creating shared tablespace for temporary tables
[root@ip-172-31-123-97 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.44

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Download stable SonarQube version from below website.

- cd /opt
- wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip
- unzip sonarqube-6.6.zip
- mv /opt/sonarqube-6.6 /opt/sonar

```
[root@ip-172-31-123-97 ~]# cd /opt
[root@ip-172-31-123-97 opt]# wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip
--2025-10-23 13:05:02--  https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip
Resolving binaries.sonarsource.com (binaries.sonarsource.com)... 3.167.37.121, 3.167.37.95, 3.167.37.10, ...
Connecting to binaries.sonarsource.com (binaries.sonarsource.com)|3.167.37.121|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 164350442 (157M) [application/zip]
Saving to: 'sonarqube-6.6.zip'

sonarqube-6.6.zip                       100%[=========================================================================

2025-10-23 13:05:02 (234 MB/s) - 'sonarqube-6.6.zip' saved [164350442/164350442]

[root@ip-172-31-123-97 opt]# unzip sonarqube-6.6.zip
Archive:  sonarqube-6.6.zip
   creating: sonarqube-6.6/
   creating: sonarqube-6.6/lib/
  inflating: sonarqube-6.6/lib/sonar-application-6.6.jar
   creating: sonarqube-6.6/lib/jsw/
  inflating: sonarqube-6.6/lib/jsw/wrapper-3.2.3.jar
   creating: sonarqube-6.6/lib/server/
```

```
  inflating: sonarqube-6.6/extensions/jdbc-driver/oracle/README.txt
  inflating: sonarqube-6.6/COPYING
  inflating: sonarqube-6.6/temp/README.txt
  inflating: sonarqube-6.6/data/README.txt
   creating: sonarqube-6.6/logs/
   creating: sonarqube-6.6/elasticsearch/plugins/
[root@ip-172-31-123-97 opt]# ls
aws  sonarqube-6.6  sonarqube-6.6.zip
[root@ip-172-31-123-97 opt]# mv /opt/sonarqube-6.6 /opt/sonar
[root@ip-172-31-123-97 opt]# ls
aws  sonar  sonarqube-6.6.zip
[root@ip-172-31-123-97 opt]#
```

Login to mysql

- mysql -u root -p

Create a new sonar database

- CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;

```
[root@ip-172-31-123-97 opt]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.7.44 MySQL Community Server (GPL)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;
Query OK, 1 row affected (0.00 sec)

mysql>
```

Create a local and a remote user

- CREATE USER sonar@localhost IDENTIFIED BY 'Sonar@123';
- CREATE USER sonar@'%' IDENTIFIED BY 'Sonar@123';

Grant database access permissions to users

- GRANT ALL ON sonar.* TO sonar@localhost;
- GRANT ALL ON sonar.* TO sonar@'%';

```
mysql> CREATE USER sonar@localhost IDENTIFIED BY 'Sonar@123';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE USER sonar@'%' IDENTIFIED BY 'Sonar@123';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL ON sonar.* TO sonar@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL ON sonar.* TO sonar@'%';
Query OK, 0 rows affected (0.00 sec)

mysql>
```

check users and databases

- show databases;
- SELECT User FROM mysql.user;
- FLUSH PRIVILEGES;
- QUIT

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sonar              |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> SELECT User FROM mysql.user;
+---------------+
| User          |
+---------------+
| sonar         |
| mysql.session |
| mysql.sys     |
| root          |
| sonar         |
+---------------+
5 rows in set (0.00 sec)
```

Go to root user and change the sonar file

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

* sonar.jdbc.username=`sonar`

* sonar.jdbc.password=`Sonar@123`

*sonar.jdbc.url=jdbc:mysql://`localhost:3306`/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false

* sonar.web.host=`0.0.0.0`

* sonar.web.context=`/sonar`

```
[root@ip-172-31-123-97 opt]# ls
aws  sonar  sonarqube-6.6.zip
[root@ip-172-31-123-97 opt]# cd sonar
[root@ip-172-31-123-97 sonar]# ls
COPYING  bin  conf  data  elasticsearch  extensions  lib  logs  temp  web
[root@ip-172-31-123-97 sonar]# cd conf
[root@ip-172-31-123-97 conf]# ls
sonar.properties  wrapper.conf
[root@ip-172-31-123-97 conf]# vi sonar.properties
```

```
#--------------------------------------------------------------------------------------
# DATABASE
#
# IMPORTANT: the embedded H2 database is used by default. It is recommended for tests but not for
# production use. Supported databases are MySQL, Oracle, PostgreSQL and Microsoft SQLServer.

# User credentials.
# Permissions to create tables, indices and triggers must be granted to JDBC user.
# The schema must be created first.
sonar.jdbc.username='sonar'
sonar.jdbc.password='sonar@123'

#----- Embedded Database (default)
# H2 embedded database server listening port, defaults to 9092
#sonar.embeddedDatabase.port=9092

#----- MySQL 5.6 or greater
# Only InnoDB storage engine is supported (not myISAM).
# Only the bundled driver is supported. It can not be changed.
sonar.jdbc.url=jdbc:mysql://localhost:3306/sonar?useUnicode=true&characterEncoding=utf8&rewriteBatchedStatements=true&useConfigs=maxPerformance&useSSL=false


#----- Oracle 11g/12c
```

```
#sonar.web.javaAdditionalOpts=

# Binding IP address. For servers with more than one IP address, this property speci
# address will be used for listening on the specified ports.
# By default, ports will be used on all IP addresses associated with the server.
sonar.web.host=0.0.0.0

# Web context. When set, it must start with forward slash (for example /sonarqube).
# The default value is root context (empty value).
sonar.web.context='/sonar'
# TCP port for incoming HTTP connections. Default value is 9000.
#sonar.web.port=9000
```

Change the ownership of sonar

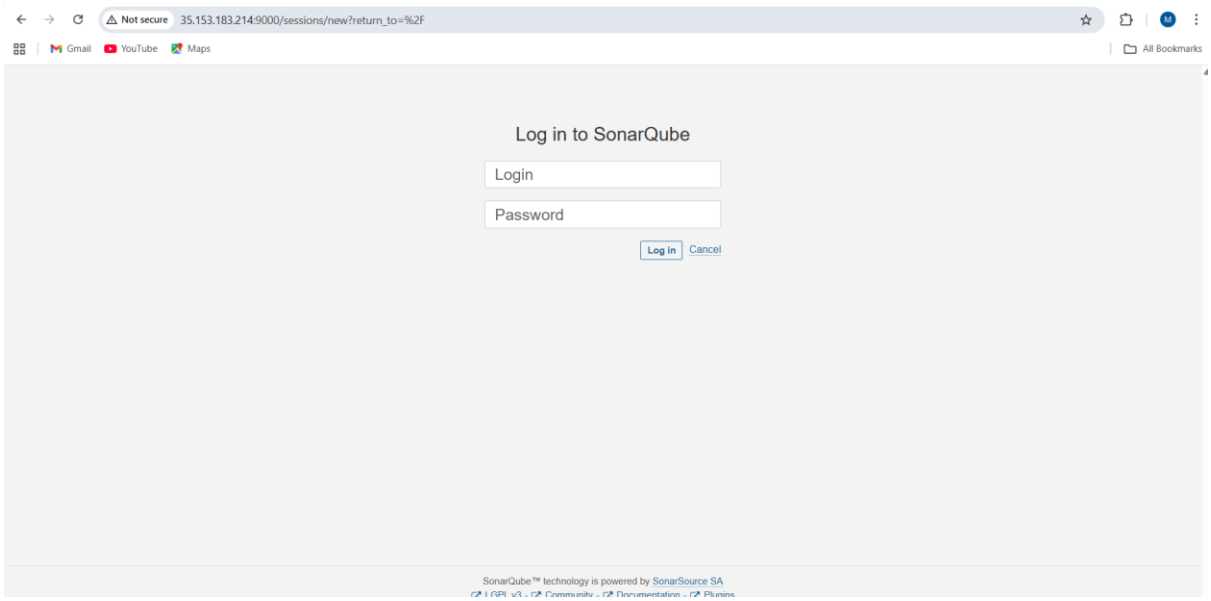
- cd /opt
- sudo chown -R ec2-user:ec2-user sonar

```
[ec2-user@ip-172-31-123-97 ~]$ cd /opt
[ec2-user@ip-172-31-123-97 opt]$ ll
total 160500
drwxr-xr-x.  4 root root        33 Oct  8 23:55 aws
drwxr-xr-x. 11 root root       141 Oct 20  2017 sonar
-rw-r--r--.  1 root root 164350442 Feb 16  2022 sonarqube-6.6.zip
[ec2-user@ip-172-31-123-97 opt]$ sudo chown -R ec2-user:ec2-user sonar
[ec2-user@ip-172-31-123-97 opt]$
```

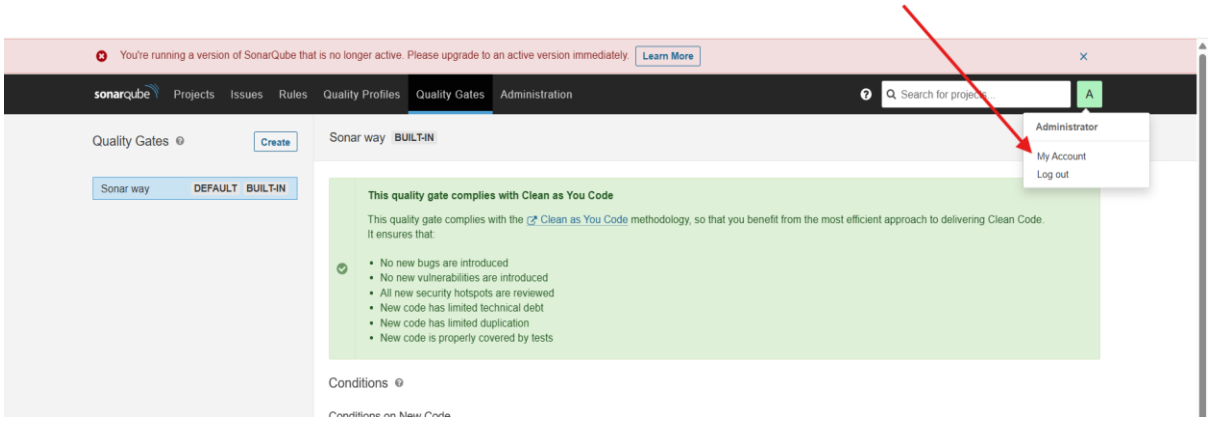
Start sonarqube

- cd /opt/sonar/bin/linux-x86-64/
- ./sonar.sh start

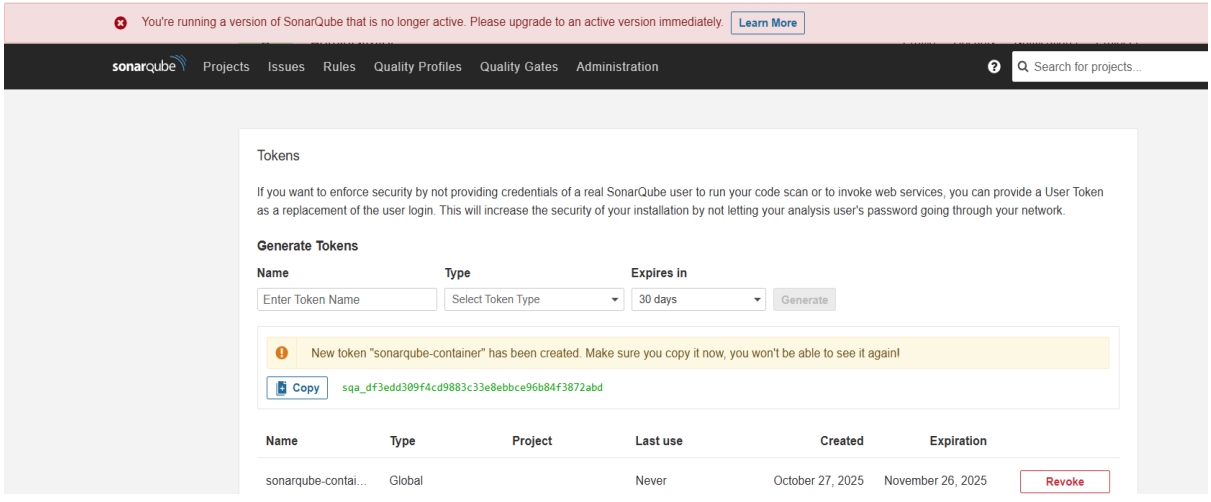
```
[ec2-user@ip-172-31-123-97 opt]$ cd /opt
[ec2-user@ip-172-31-123-97 opt]$ ls
aws  sonar  sonarqube-6.6.zip
[ec2-user@ip-172-31-123-97 opt]$ cd sonar
[ec2-user@ip-172-31-123-97 sonar]$ ls
COPYING  bin  conf  data  elasticsearch  extensions  lib  logs  temp  web
[ec2-user@ip-172-31-123-97 sonar]$ cd bin
[ec2-user@ip-172-31-123-97 bin]$ ls
jsw-license  linux-x86-32  linux-x86-64  macosx-universal-64  windows-x86-32  windows-x86-64
[ec2-user@ip-172-31-123-97 bin]$ cd linux-x86-64
[ec2-user@ip-172-31-123-97 linux-x86-64]$ ls
lib  sonar.sh  wrapper
[ec2-user@ip-172-31-123-97 linux-x86-64]$ ./sonar.sh start
Starting SonarQube...
Started SonarQube.
[ec2-user@ip-172-31-123-97 linux-x86-64]$
```



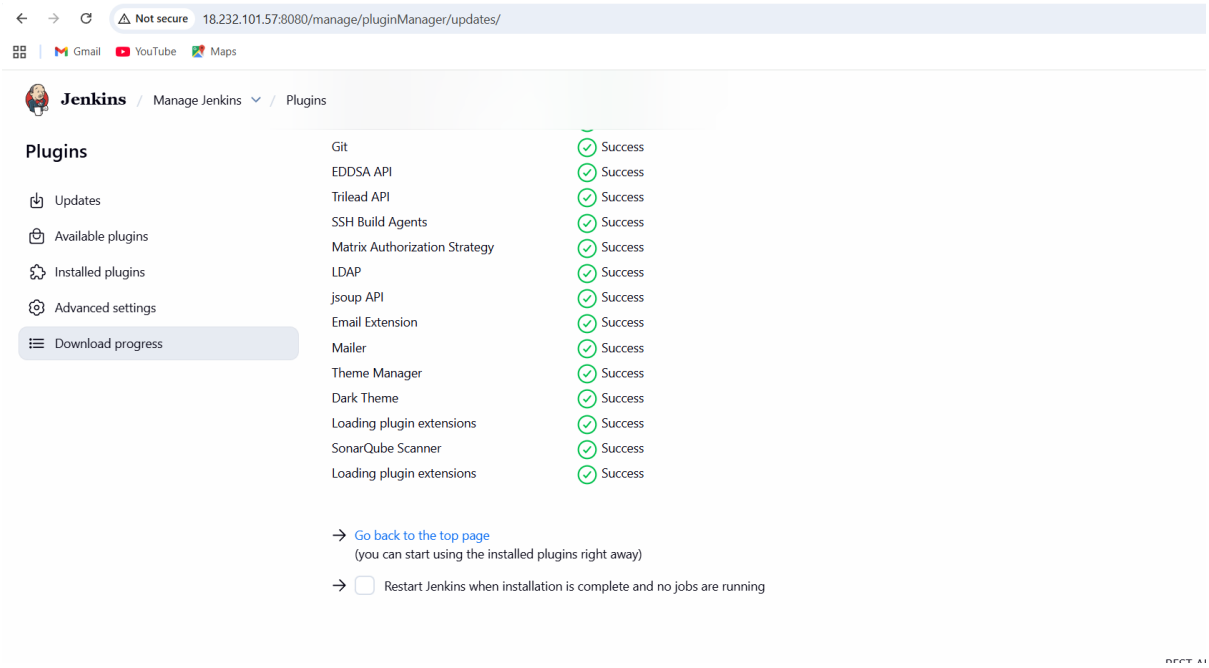
Go to my accounts



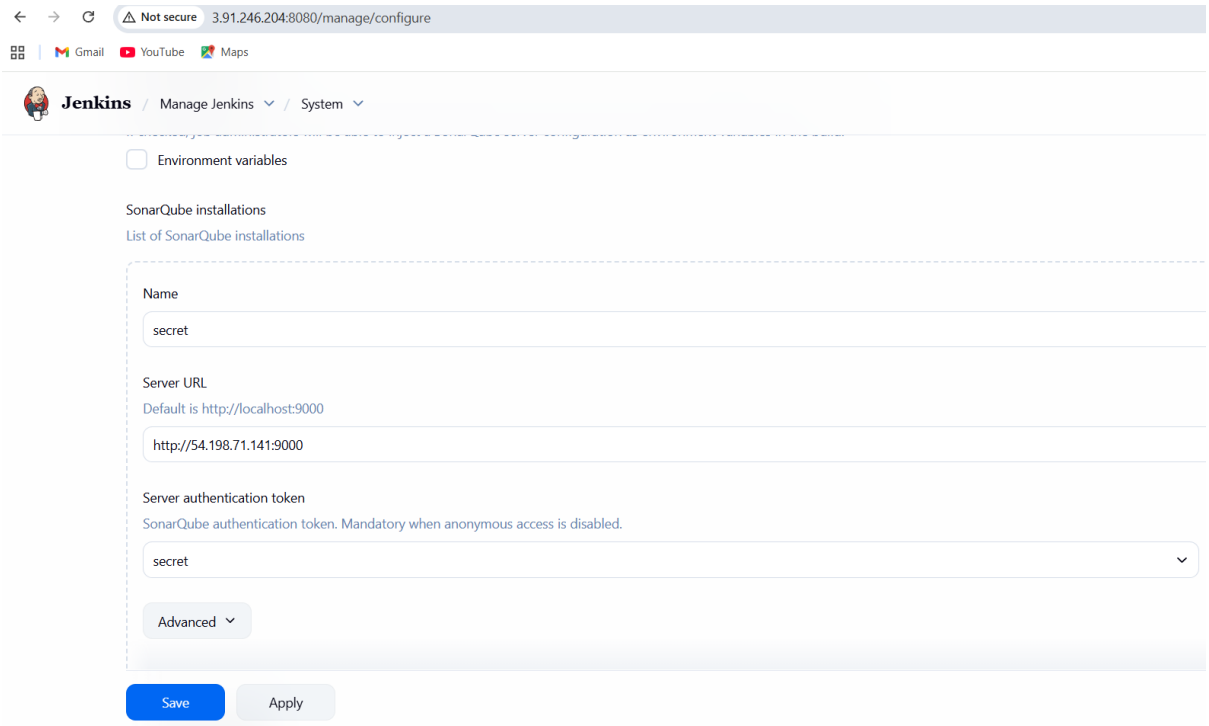
Generate a token in the security field.



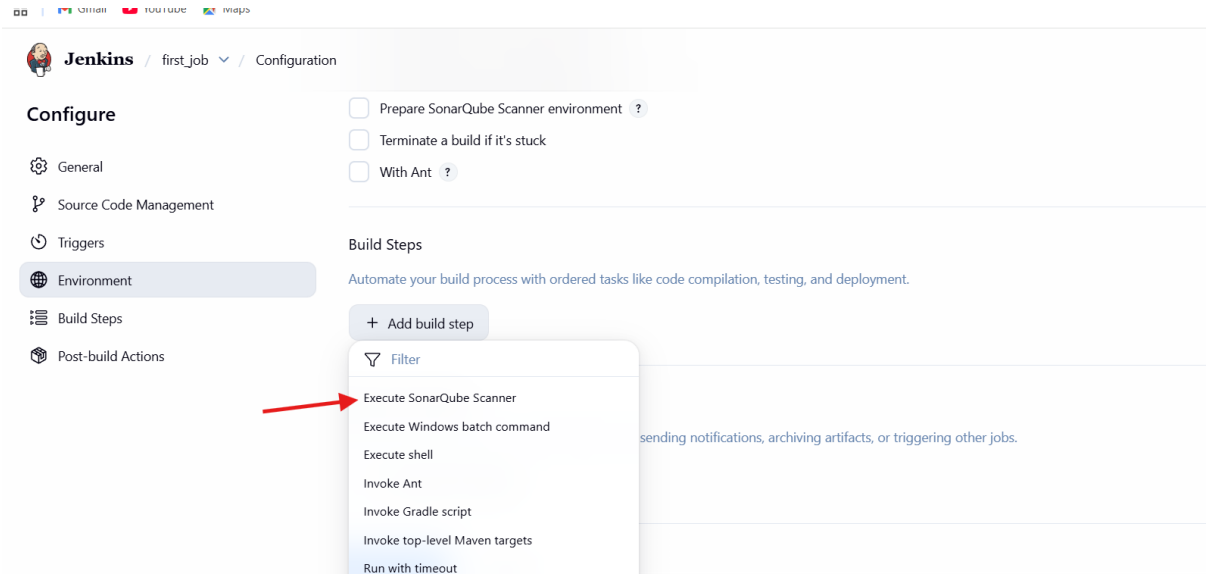
Install sonarqube scanner in plugins.



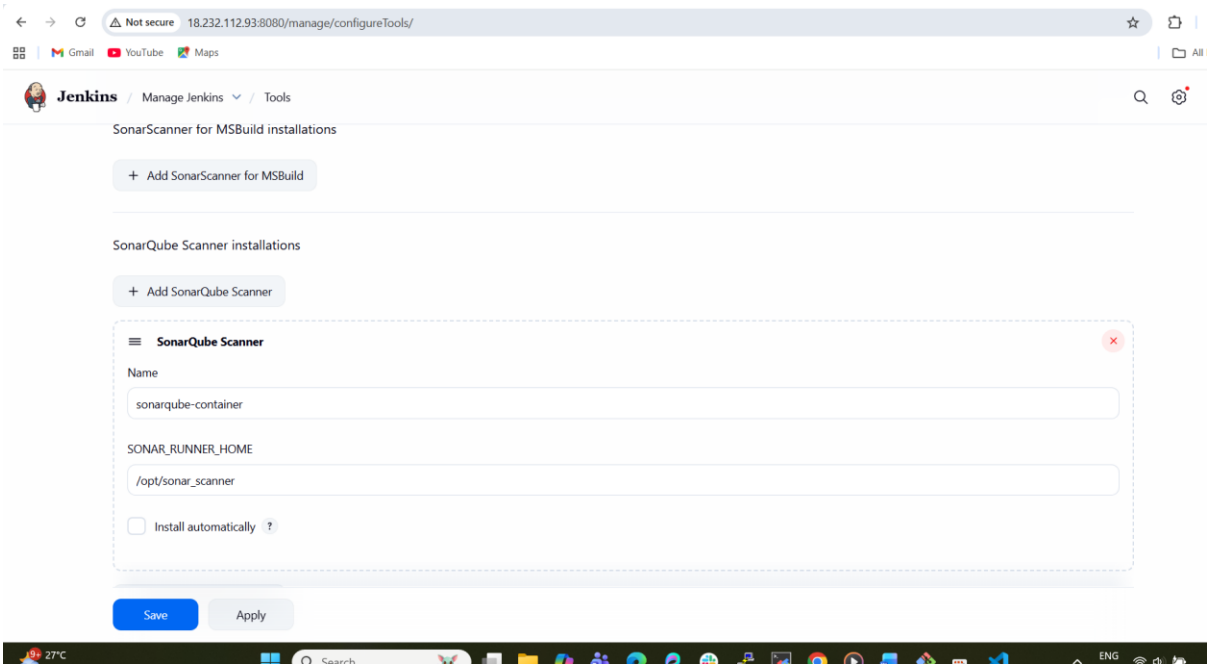
Go to managed Jenkins and in sonarqube server give the name and url of your sonarqube and add credentials with secret text copy the token and paste there and save .



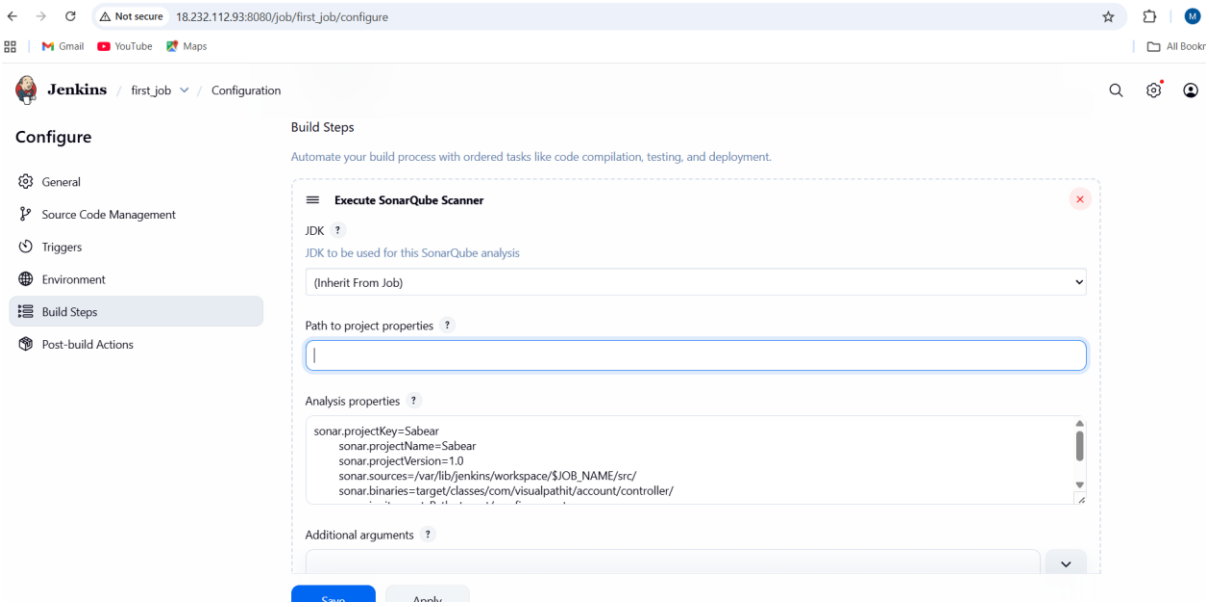
Go to your job and go to build steps click on execute sonarqube scanner and give the analysis properties of your code and give the credentials and save it.



Go to manage Jenkins click on tools click on add sonarqube scanner and give your sonar scanner path.



Go to job , configure and write the code in analysis properties.



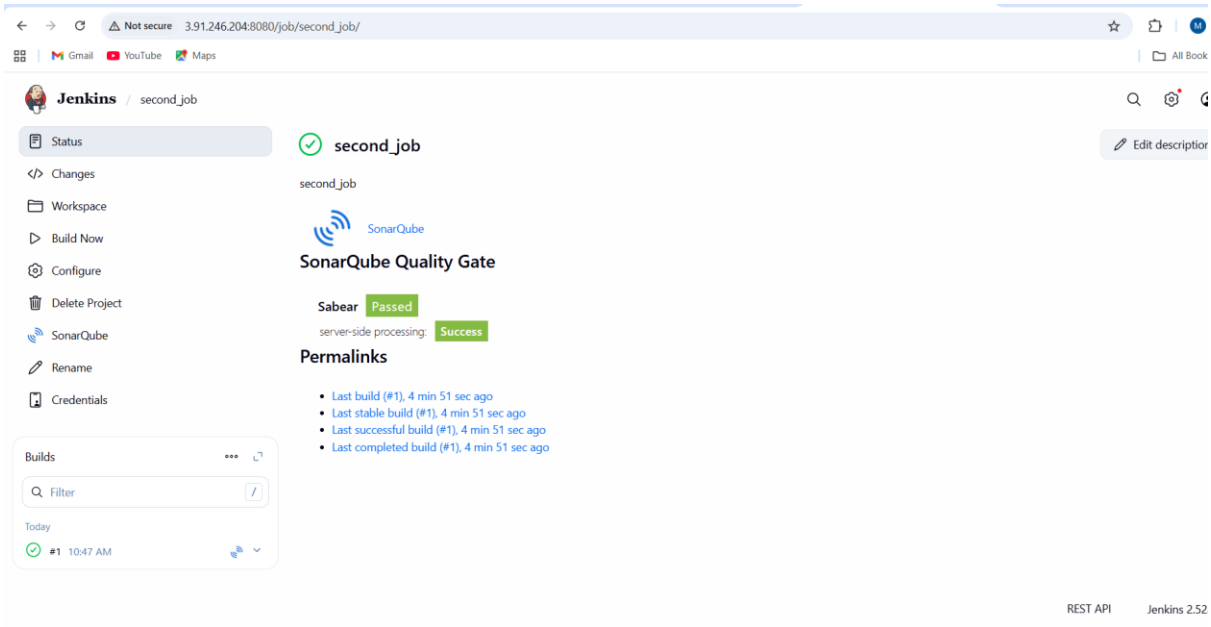
Download sonar scanner in the Jenkins server

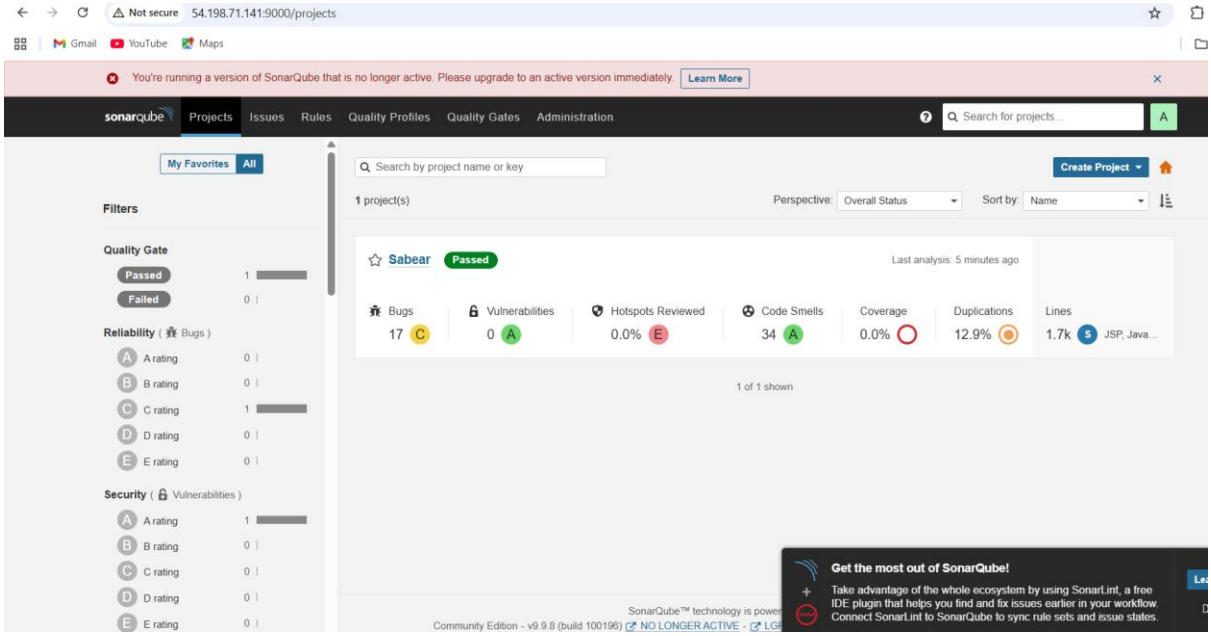
- wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-4.6.2.2472-linux.zip
- unzip sonar-scanner-cli-4.6.2.2472-linux.zip
- mv sonar-scanner-4.6.2.2472-linux /opt/sonar_scanner

change the ownership

- sudo chown -R jenkins:jenkins /opt/sonar_scanner
- sudo chmod -R 755 /opt/sonar_scanner

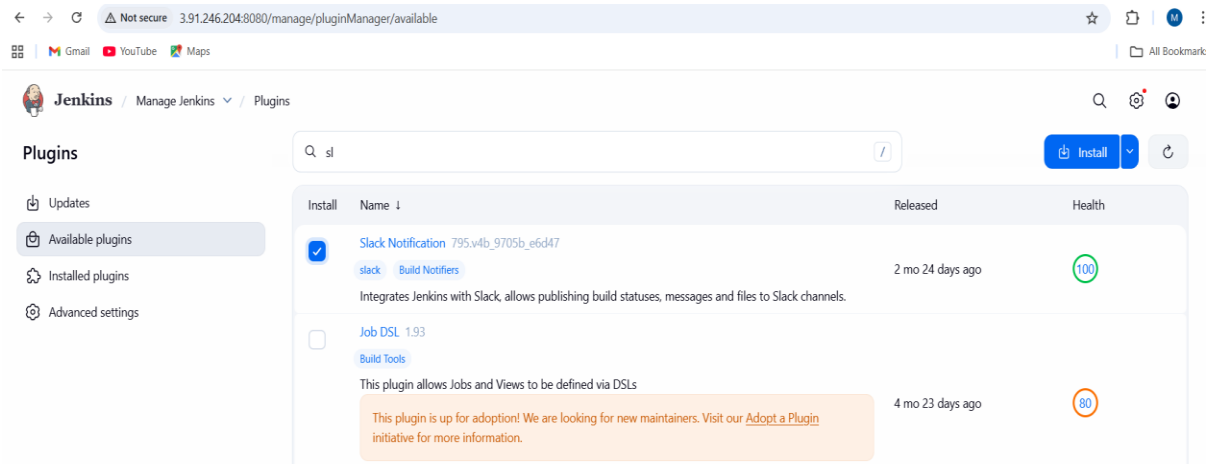
then go to the job and run the build now.



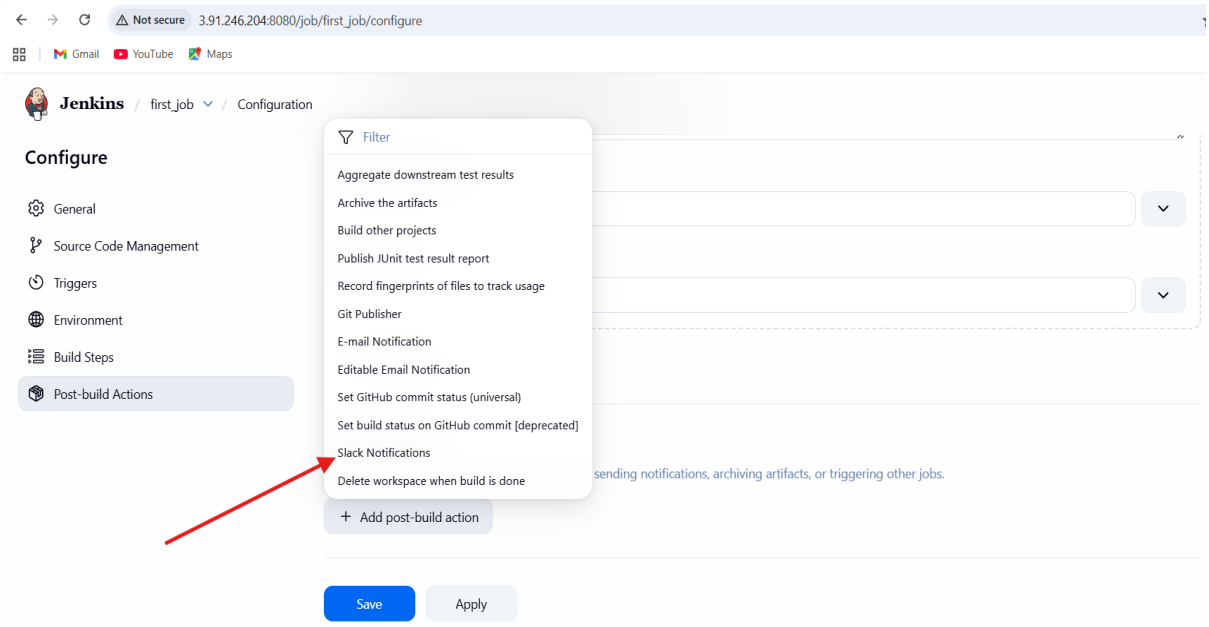


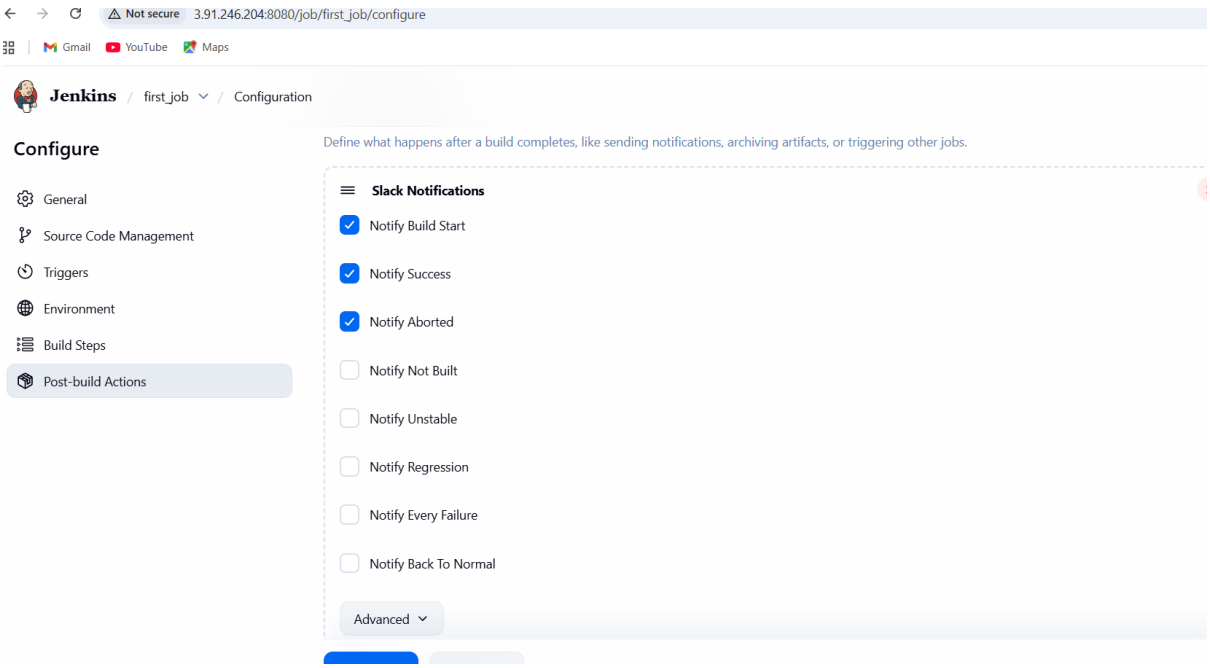
- **stage3**: Slack Integration to send the alerts to slack.

Go to managed Jenkins and click on plugins download the plugin slack notification.



Go to job and in configuration click on post build actions, select slack notifications.





Docker method:

- sudo su –
- yum install java
- yum install docker
- systemctl start docker
- docker pull sonarqube
- docker pull postgres
- sudo docker network create sonarnet
- sudo docker network connect sonarnet sonardb
- sudo docker network connect sonarnet sonarqube
- docker ps-a
- check with ip address in browser
- do the integration same as traditional method.