**Jenkins-03**

**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**

***stage1: Git clone to download the source code.***

**Step 1: Open Jenkins Dashboard**

* Go to your Jenkins URL → http://<your-server-ip>:8080/

### Step 2: Create a New Job

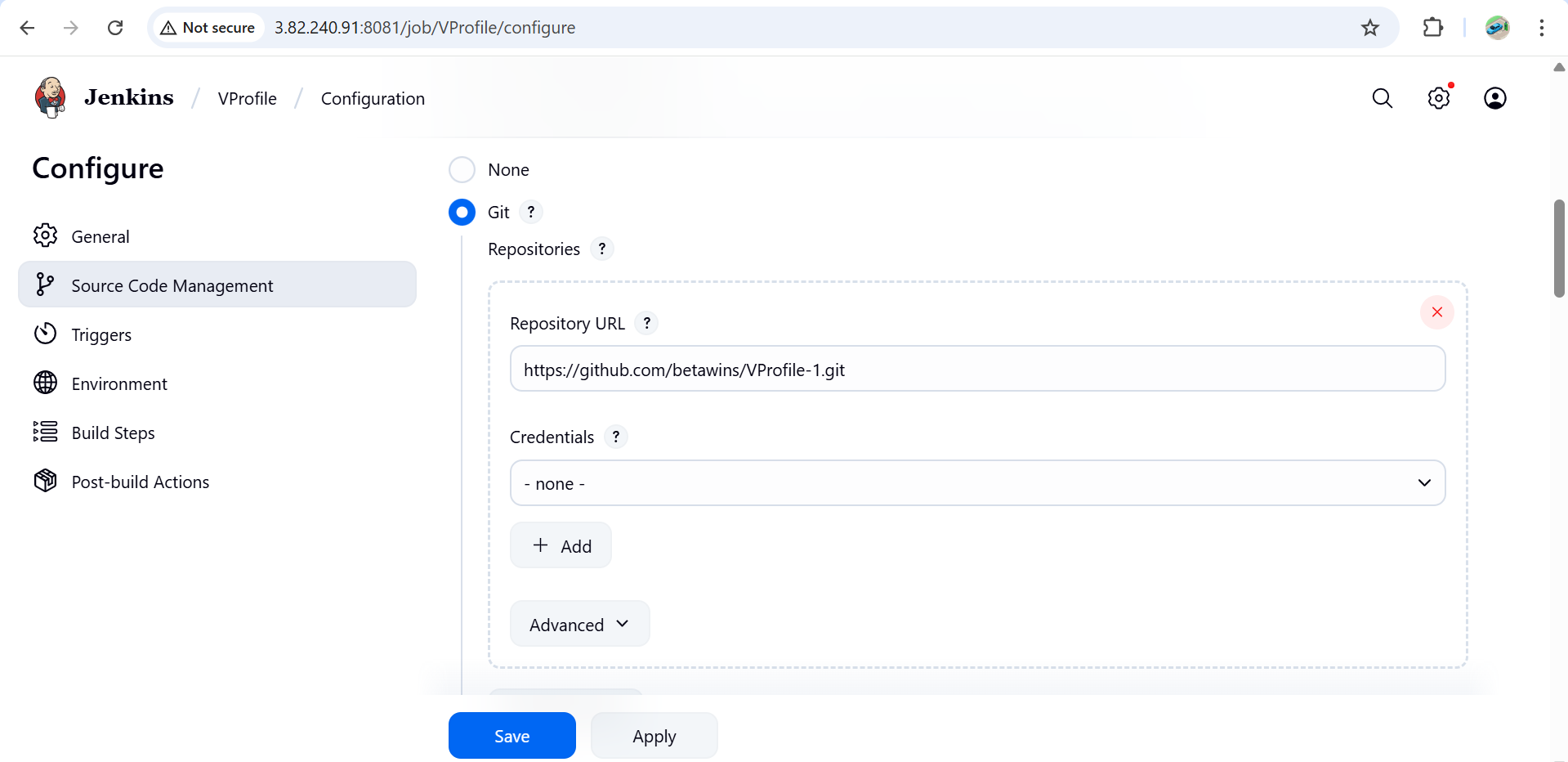
1. Click on **“New Item”**
2. Enter a job name → e.g. - VProfile.
3. Select **“Freestyle project”**.
4. Click **OK**.

### Step 3: Configure Git Repository

1. In the job configuration page:
   * Scroll to **Source Code Management**.
   * Select **Git**.
   * In **Repository URL**, enter your Git repo link:

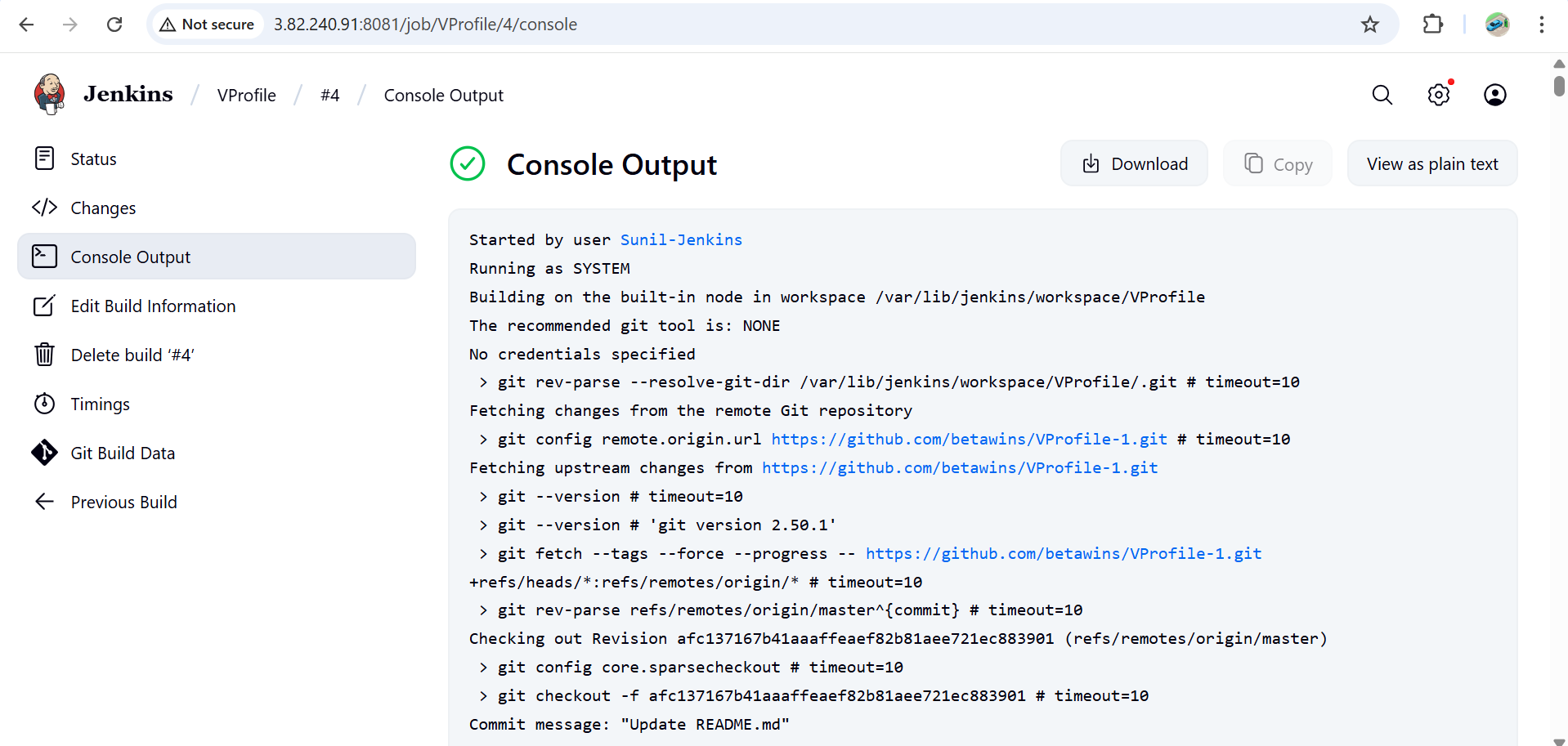
[**https://github.com/betawins/VProfile-1.git**](https://github.com/betawins/VProfile-1.git)

In **Branches to build**, specify the branch (default: \*/main or \*/master).



### Step 4: Save and Build

* Click **Save**.
* Click **Build Now**.
* Check the **Console Output** → It should show Jenkins cloning the repository.



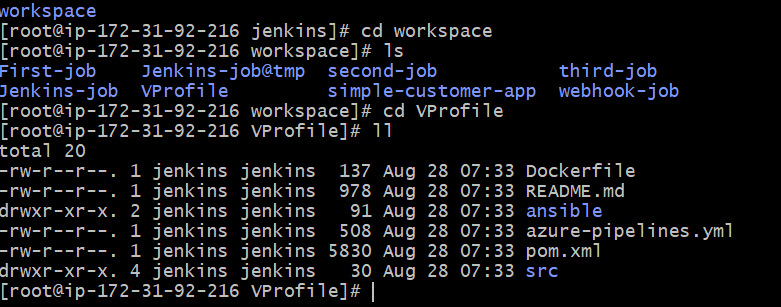
**Now successfully we have created the VProfile job and cloned the source code from git hub using the url**

**We can check by login to your ssh and in**

cd /var/lib/Jenkins

cd workspace

in workspace we ll see the Vprofile



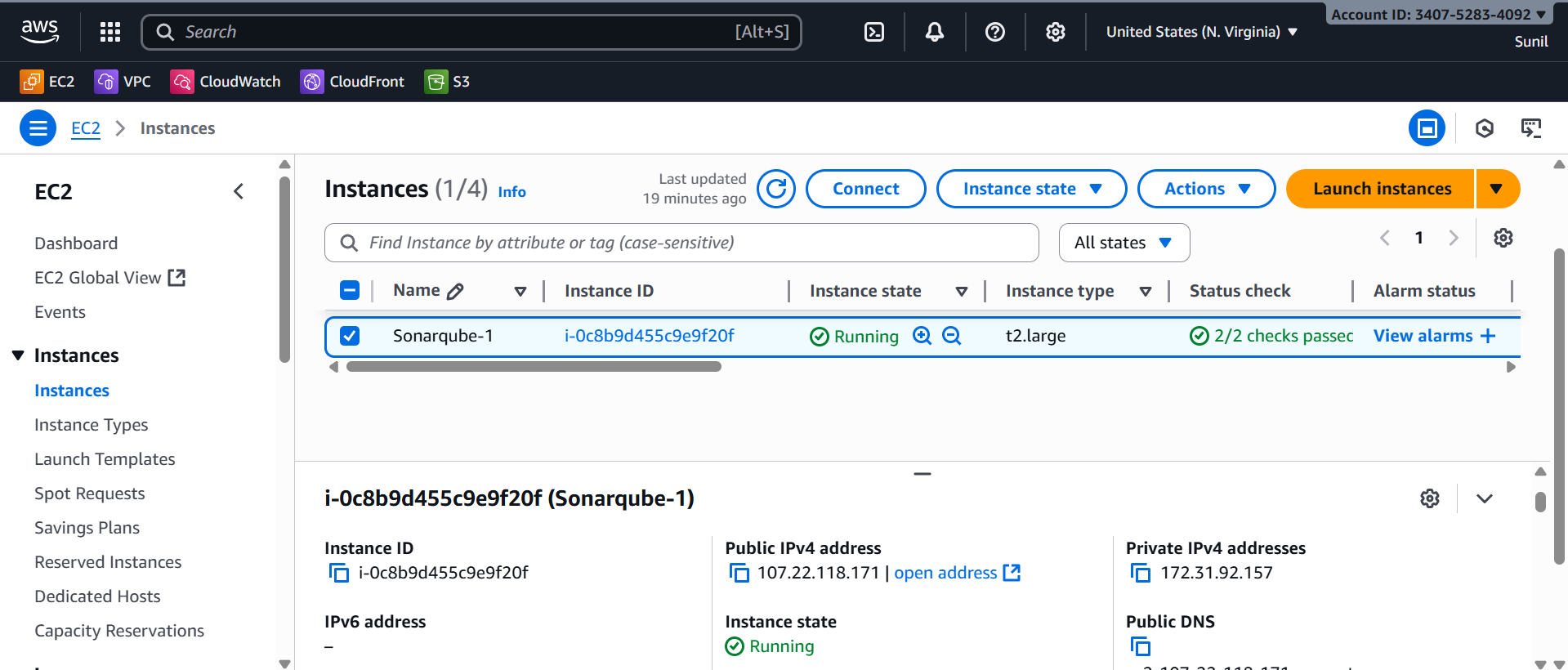
***stage2: Sonarqube Integration to check the quality of code***

### Prerequisites

* + EC2 instance with Java installed
  + Use t2.large with atleast 20gb memory to run sonarqube.
  + MySQL Database Server or MySQL RDS instance.

**1.Step-by-Step SonarQube Installation on EC2**

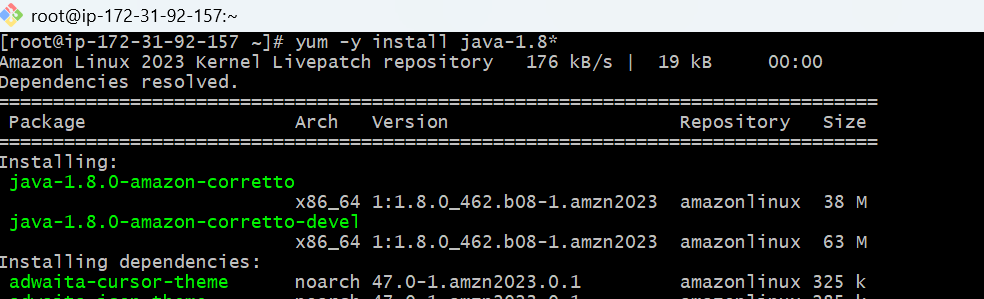
* + Launch EC2 Instance
  + Instance type: **t2.large**
  + Storage: **20 GB** (minimum).

****

**2. Install Java**

Install java 1.8 version

yum install java-1.8\*

****

**3.Install & Configure MySQL**

Add mysql rpm Repository

sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm

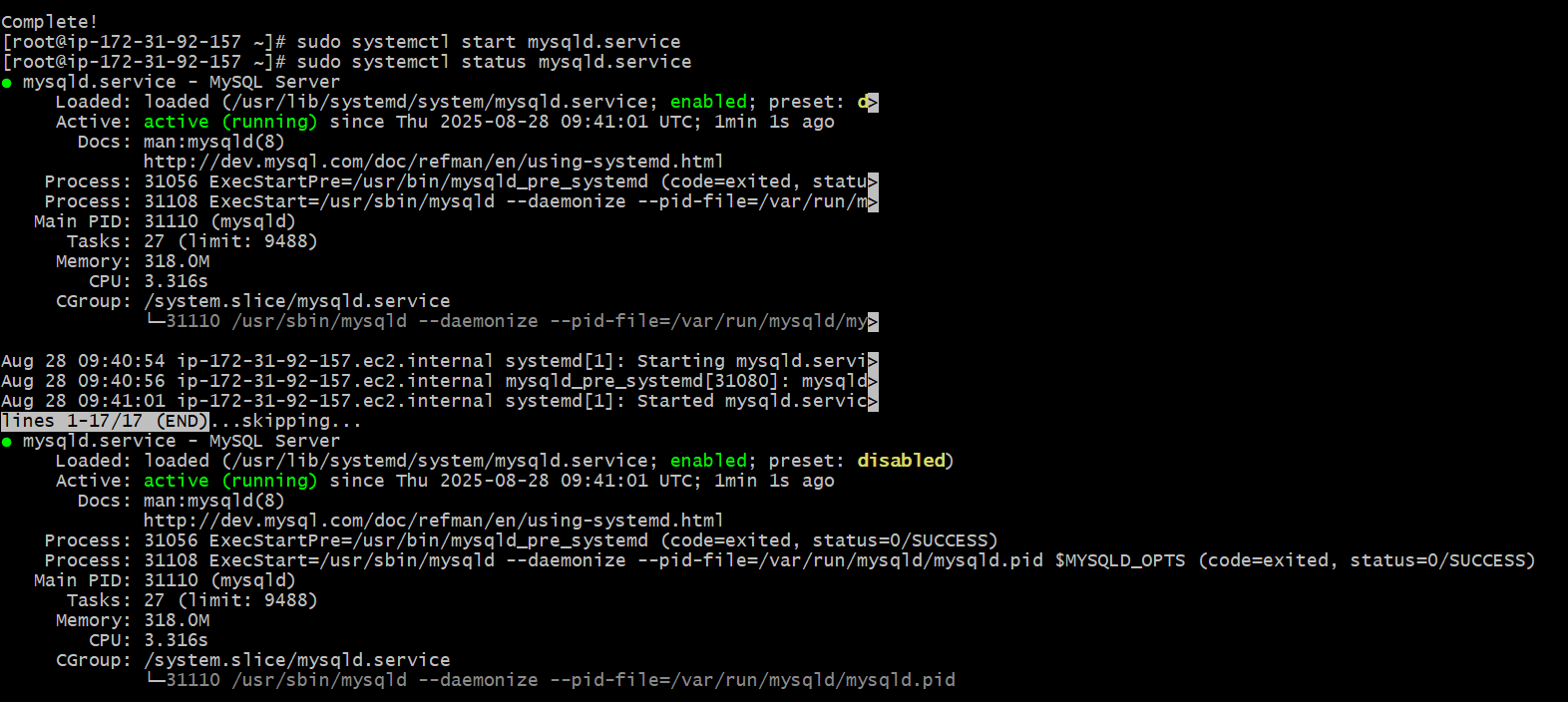
sudo yum localinstall -y mysql57-community-release-el7-11.noarch.rpm

sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022

sudo yum install -y mysql-community-server

****

sudo systemctl start mysqld.service

****

Start MySQL and Enable Start at Boot Time

systemctl start mysqld

systemctl enable mysqld

Check if mysql is running or not

netstat -na | grep 3306

**4. Configure MySQL Root Password**

Configure the MySQL Root Password

You will see default MySQL root password

```sh

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

```

Login to mysql using the default password

```sh

mysql -u root -p

```

Now replace the default password with a new and strong password

```sh

ALTER USER 'root'@'localhost' IDENTIFIED BY 'Admin@123';

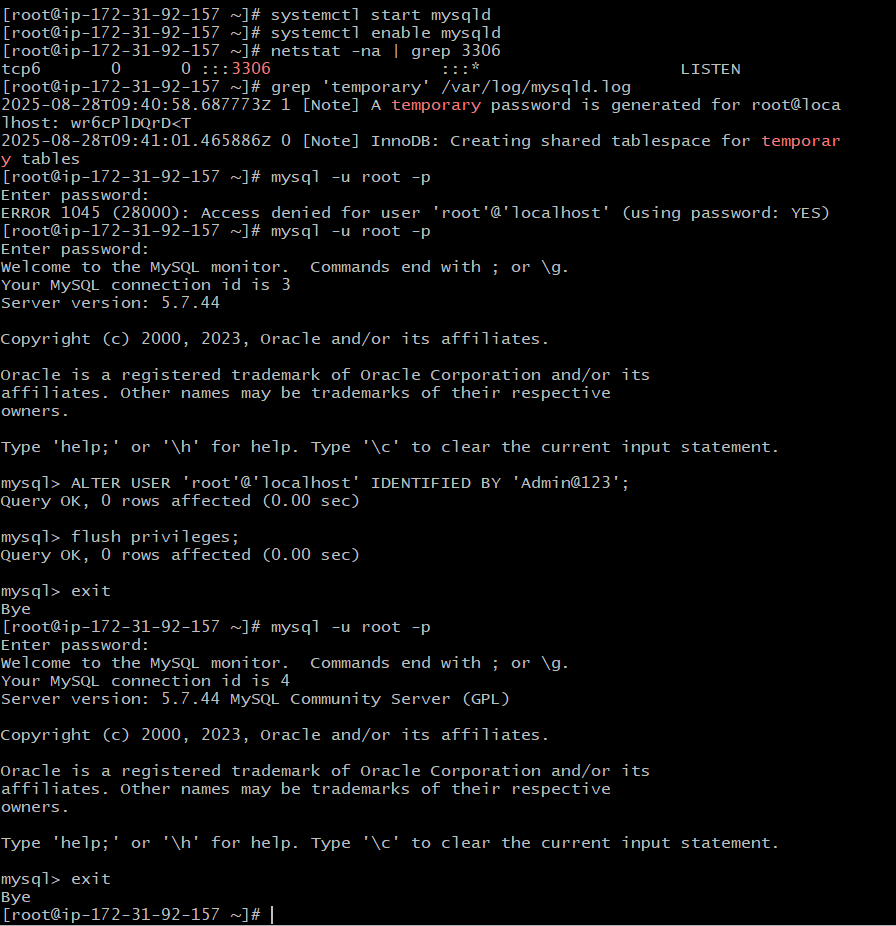
flush privileges;

```

Test Using new password

```sh

mysql -u root -p

****

**Exit**

**And Download the sonarqube using the below commands**

1. **Install SonarQube**

Download stable SonarQube version from below website.

- Website: https://www.sonarqube.org/downloads/

- Note: This Article written for SonarQube6.0

Download & unzip SonarQube 6.0

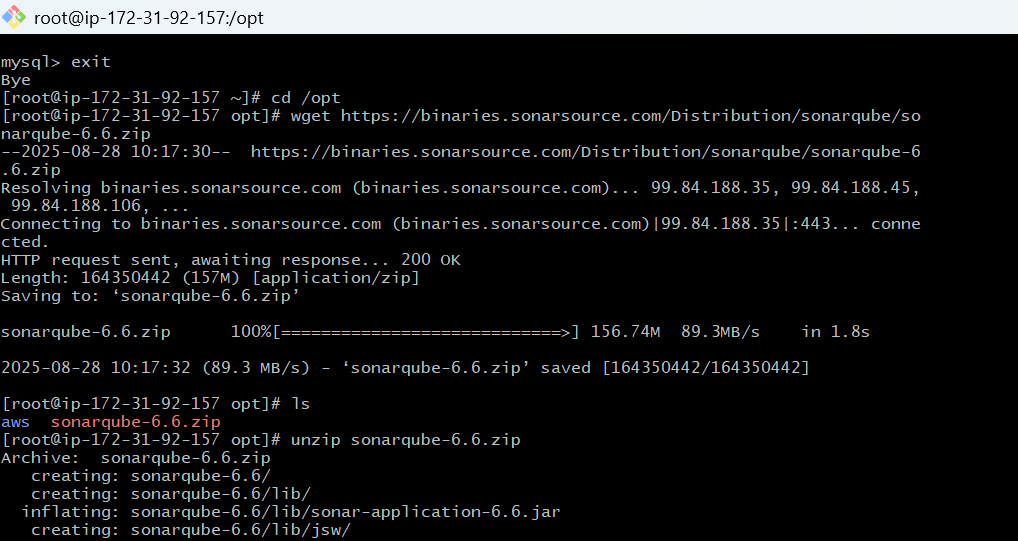
```sh

# cd /opt

# wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.6.zip

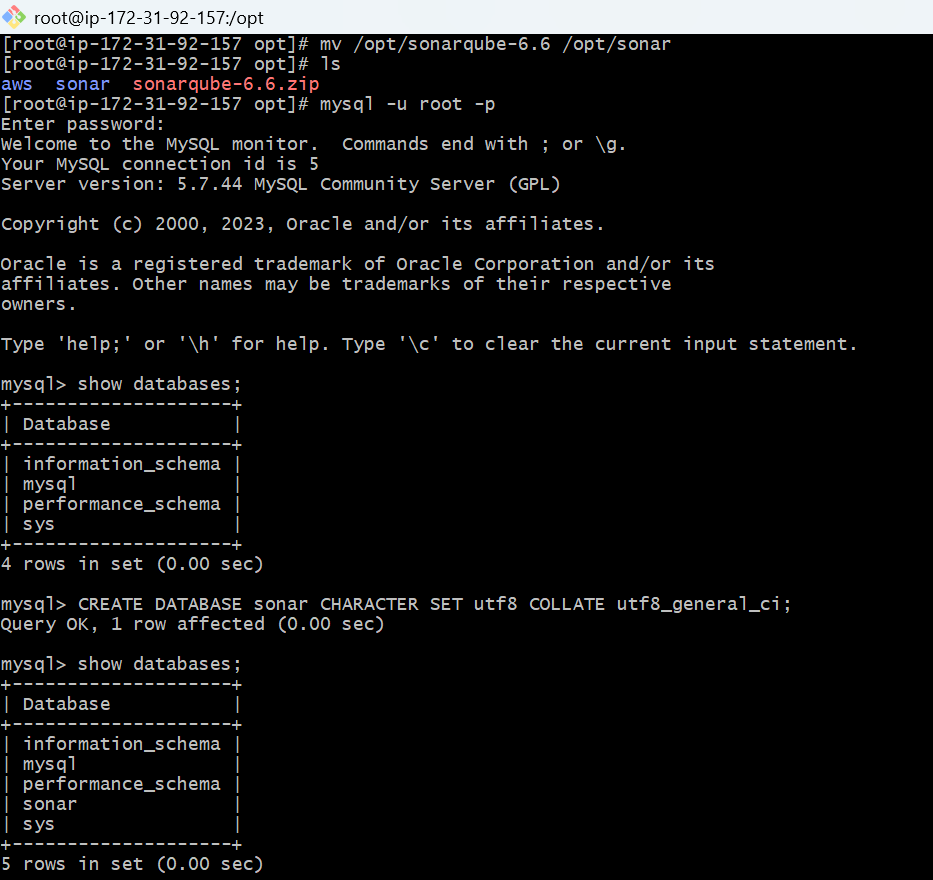
# unzip sonarqube-6.6.zip

# mv /opt/sonarqube-6.6 /opt/sonar

****

Login to mysql ---- mysql -u root -p

Show databases

****

**6.Create SonarQube Database & User**

Create a new sonar database

```sh

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

```

Create a local and a remote user

```sh

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

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

```

Grant database access permissions to users

```sh

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

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

```

check users and databases

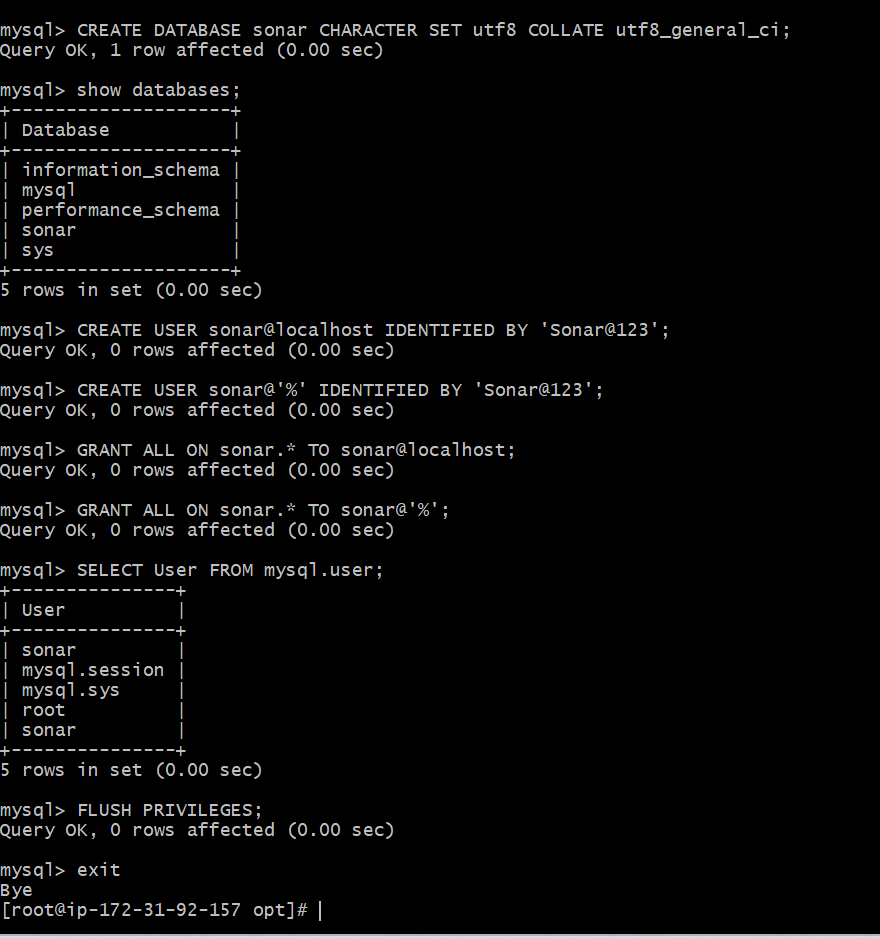
```sh

show databases;

SELECT User FROM mysql.user;

FLUSH PRIVILEGES;

QUIT

****

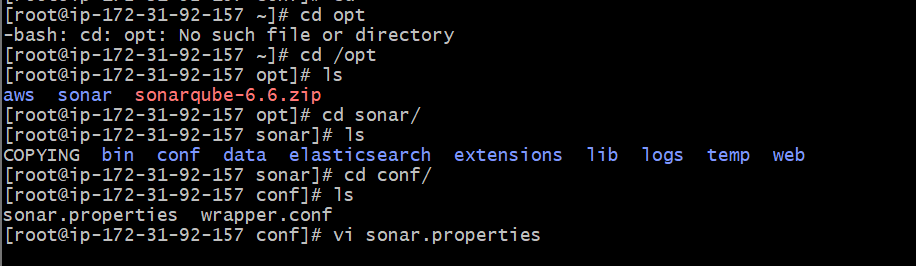
**7. Configure SonarQube with MySQL**

So for you have configured required database information on mysql. Let’s Jump back to your EC2 instance and enable SonarQube properties file to connect his Database.

### ON EC2 Instance

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

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

****

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

Edit properties file:

Vi sonar .properties

Uncomment and update:

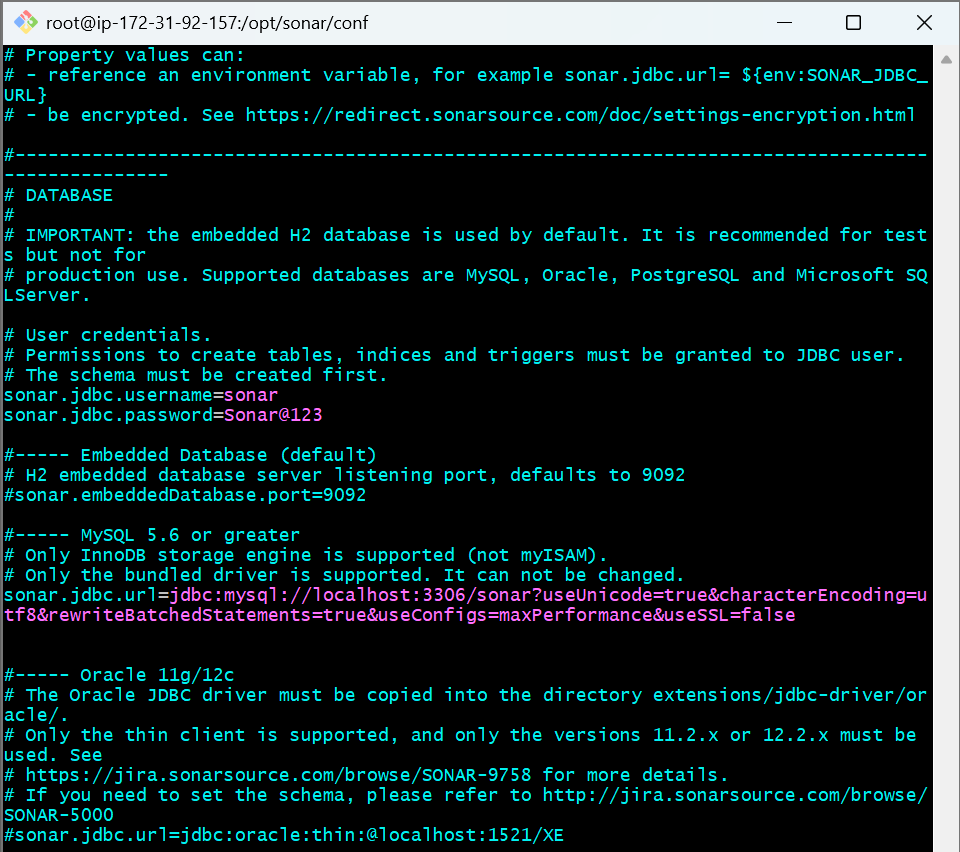
- sonar.jdbc.username=`***sonar`***

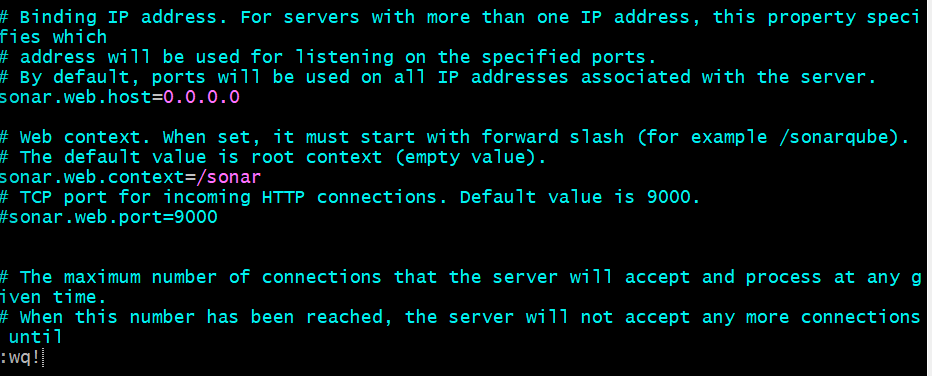
- sonar.jdbc.password=***`Sonar@123`***

- 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`***

****

****

**8. Start SonarQube**

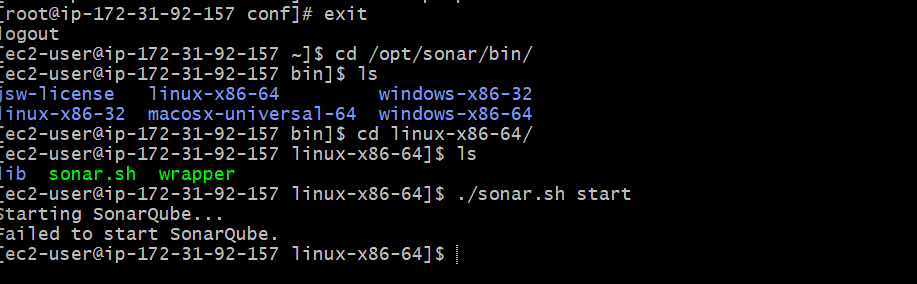
Start SonarQube service

```sh

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

# ./sonar.sh start

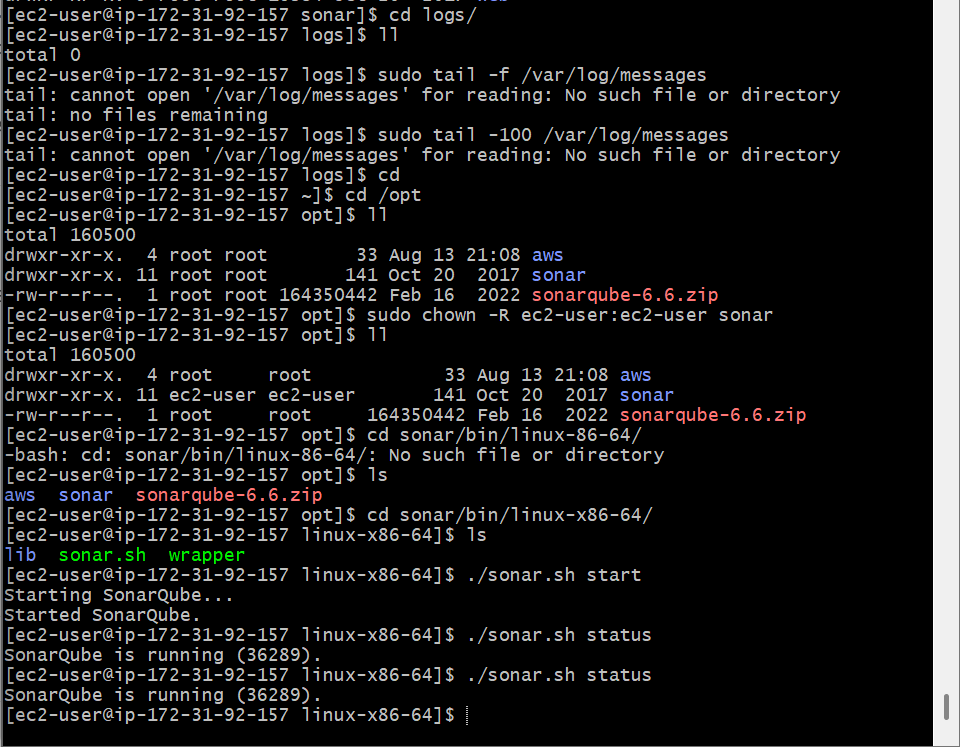
```

****

**While starting the Sonarqube it failed to start**

**use this command to give permissions and the start again**

sudo chown -R ec2-user:ec2-user sonar

****

**9. Access SonarQube UI**

SonarQube application uses port 9000. access SonarQube from browser

```sh

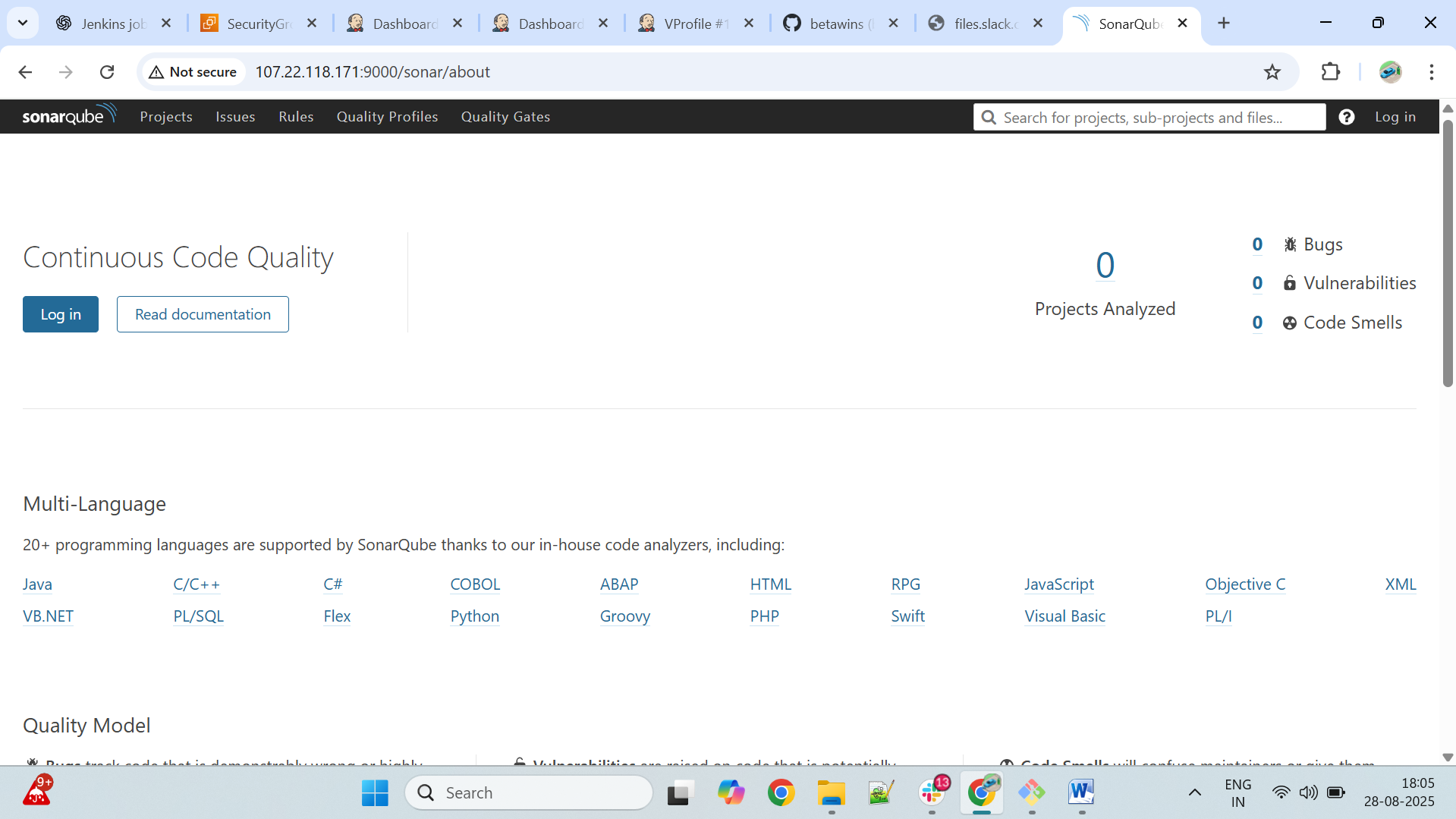
http://<EC2\_PUBLIC\_IP>:9000/sonar

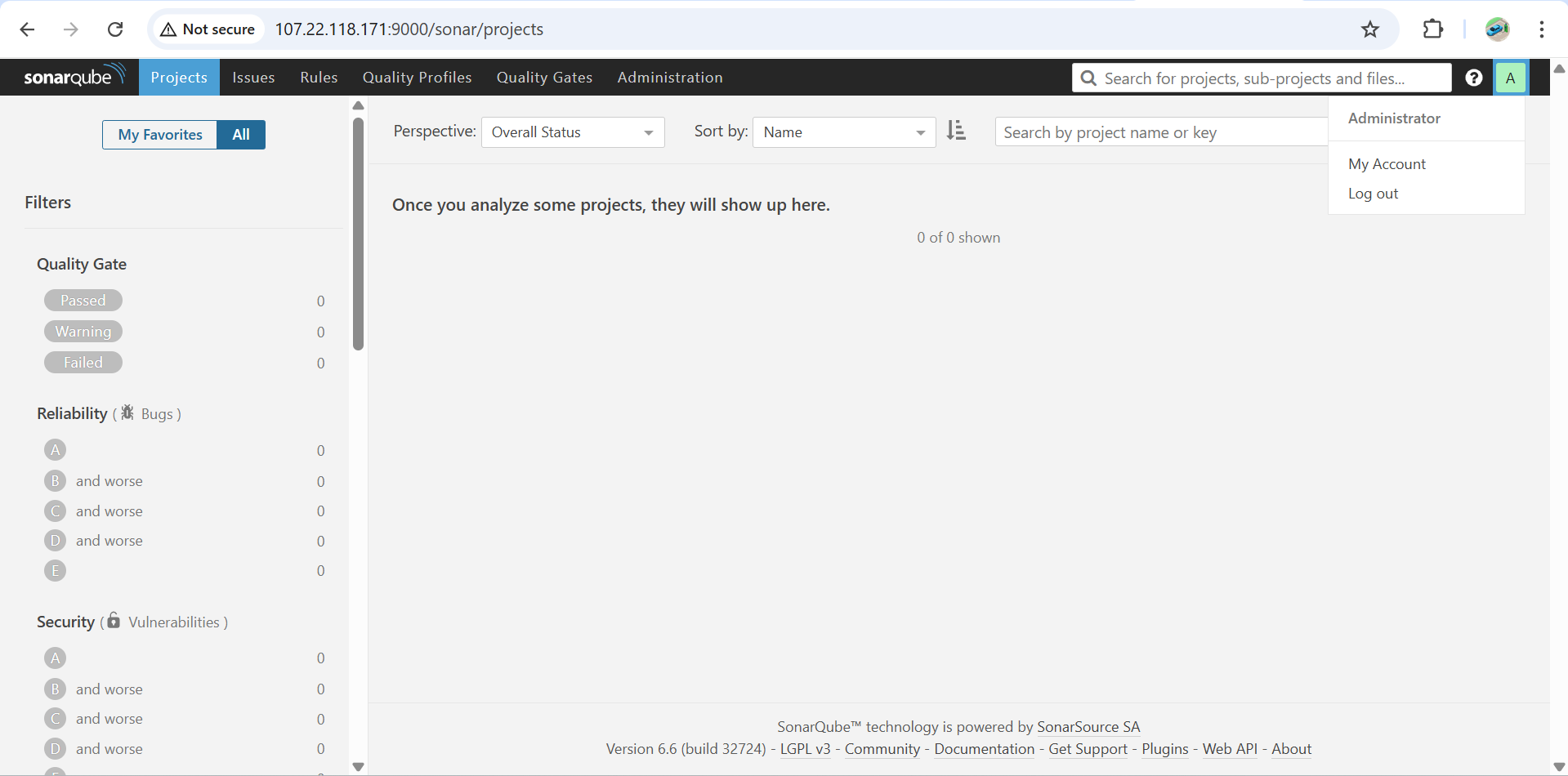
```

Default credentials for sonarqube.

Username : admin

Password: admin

****

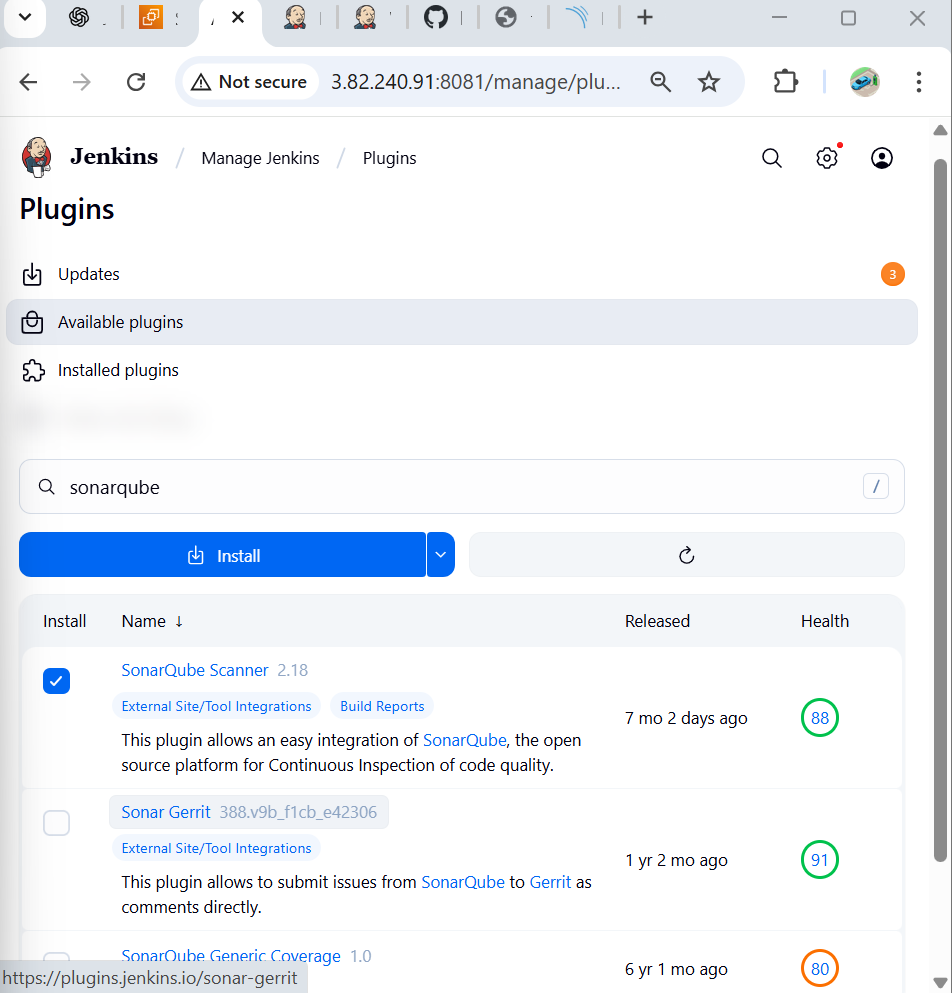
****

***Jenkins + SonarQube Integration***

**1.Install SonarQube Scanner Plugin in Jenkins**

**Login to Jenkins GUI console and install " SonarQube scanner" plugin**

**- `Manage Jenkins` > `Manage Plugins` > `Avalable` > `SonarQube scanner`**

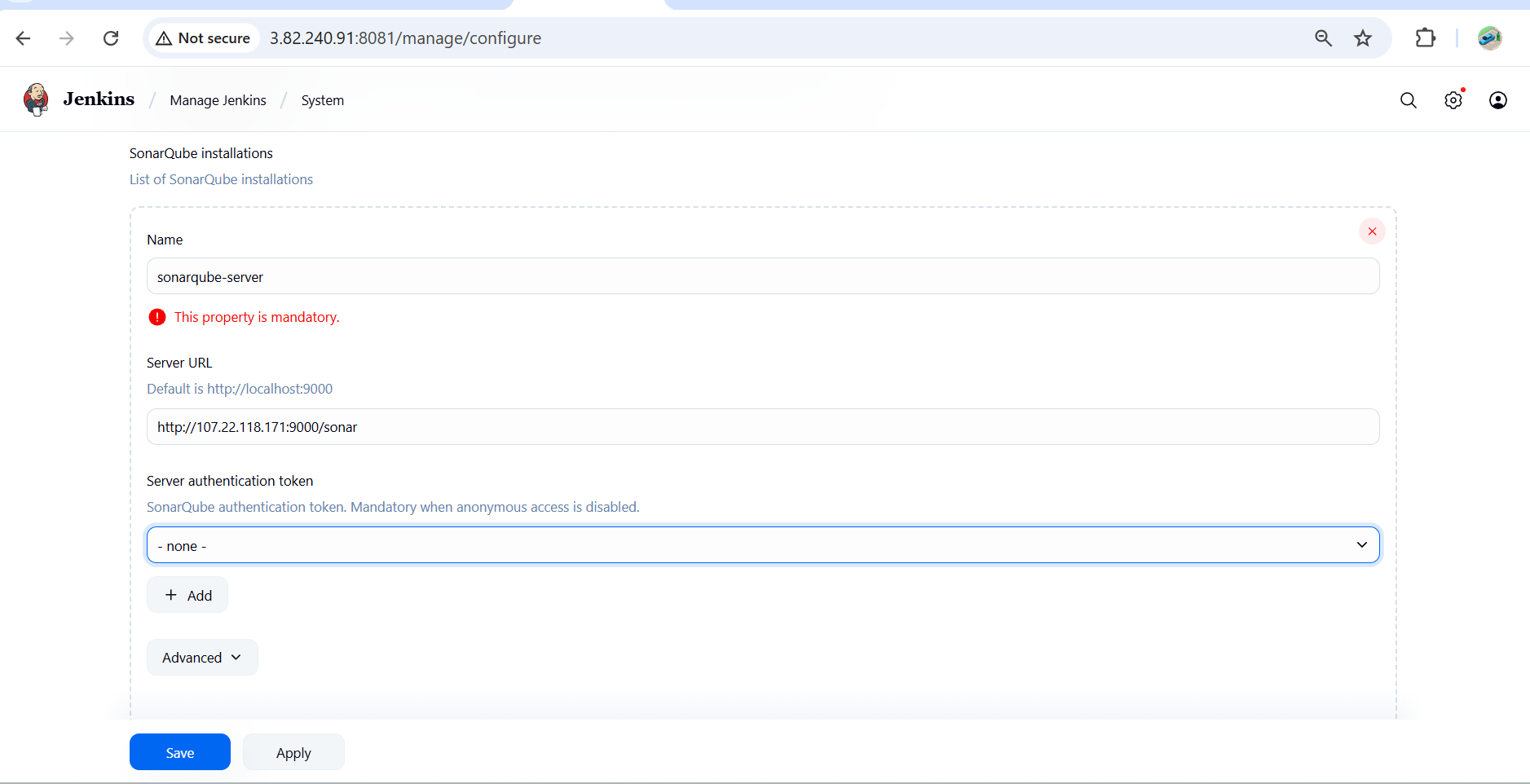
****

**2.Configure SonarQube Server in Jenkins**

Go to:  
**Manage Jenkins → Configure System**.

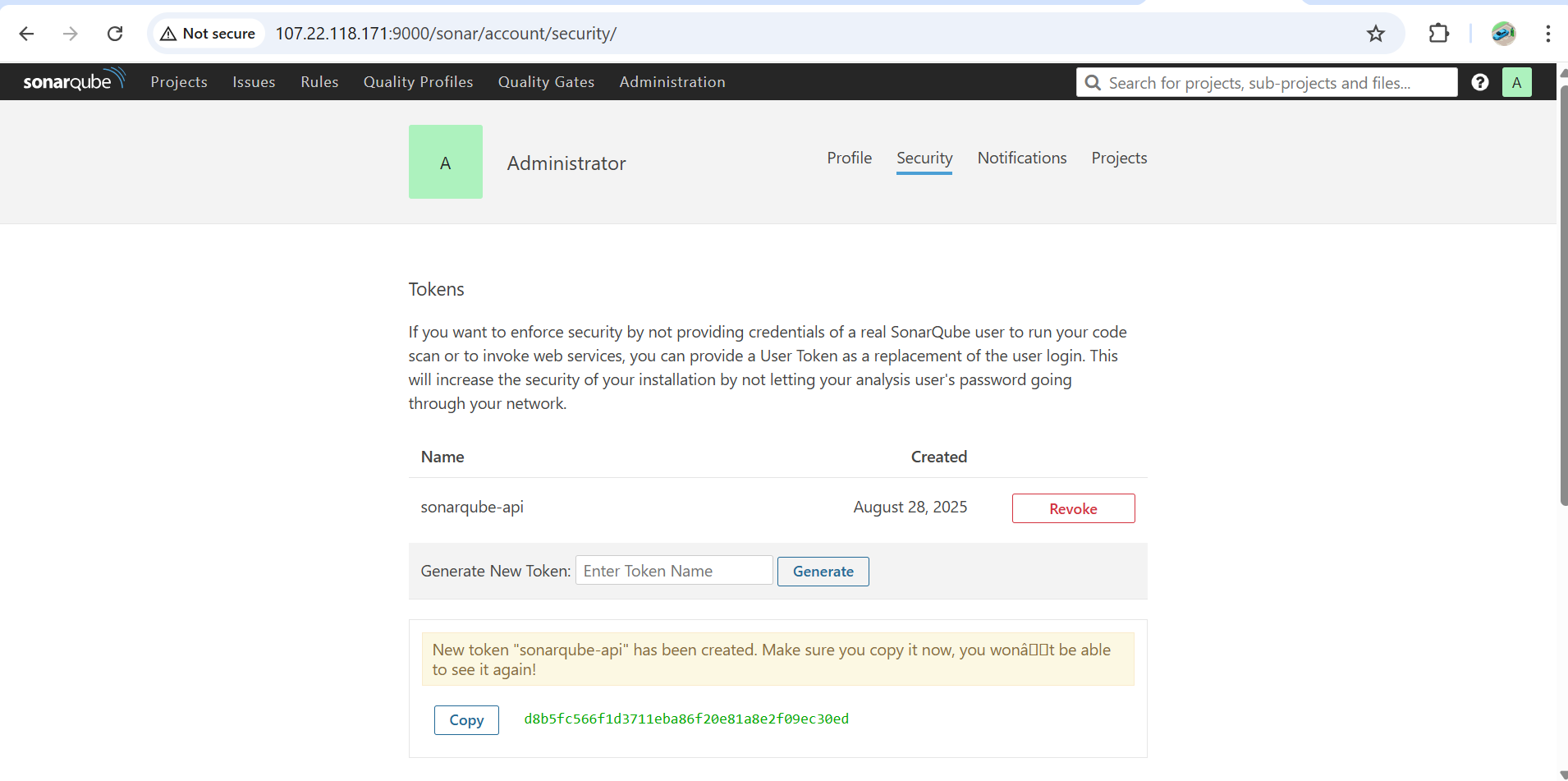
Scroll down to **SonarQube servers** → **Add SonarQube**.

* **Name:** SonarQube
* **Server URL:** http://<Sonarqube\_server>:9000/sonar

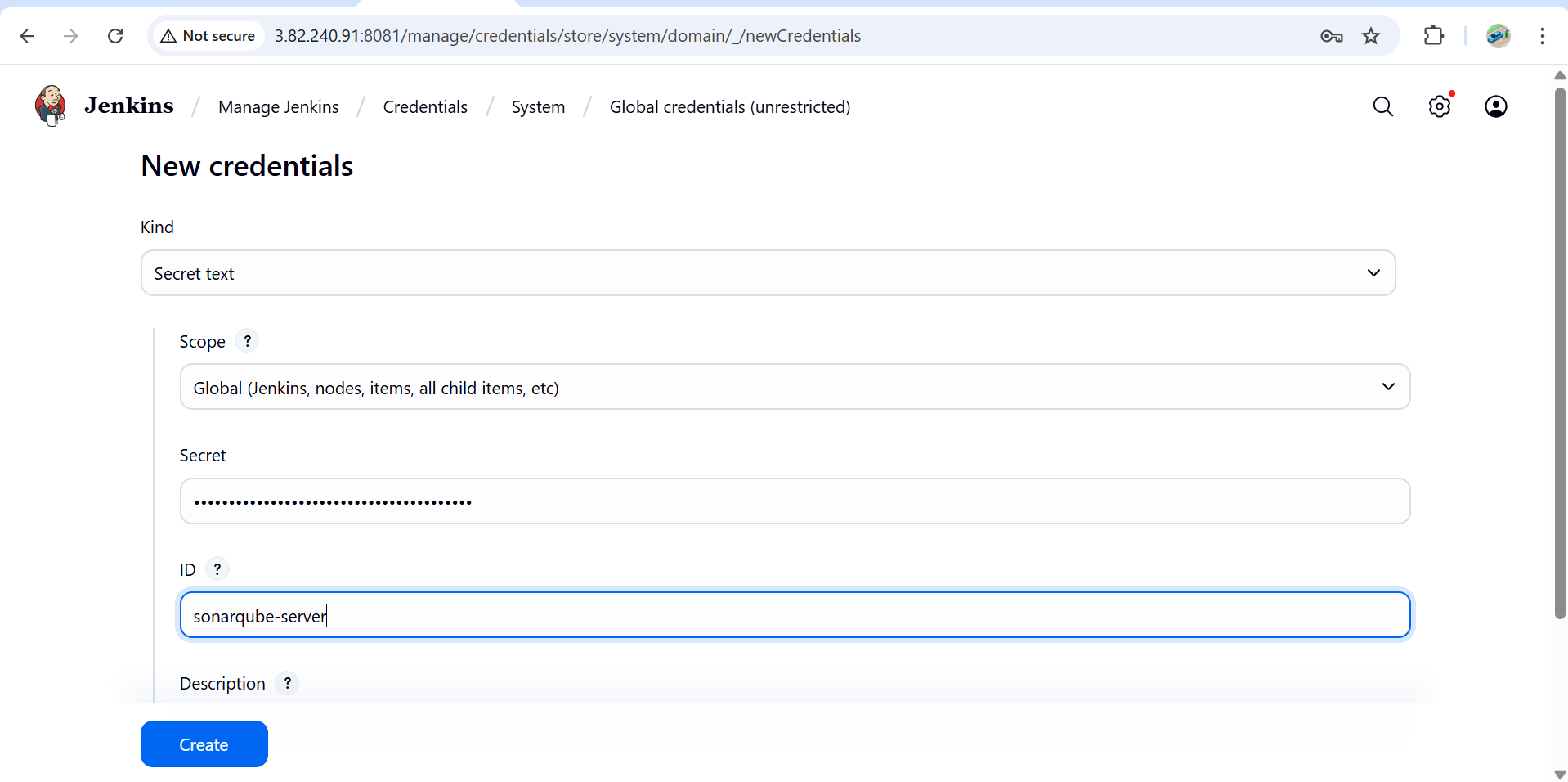
****

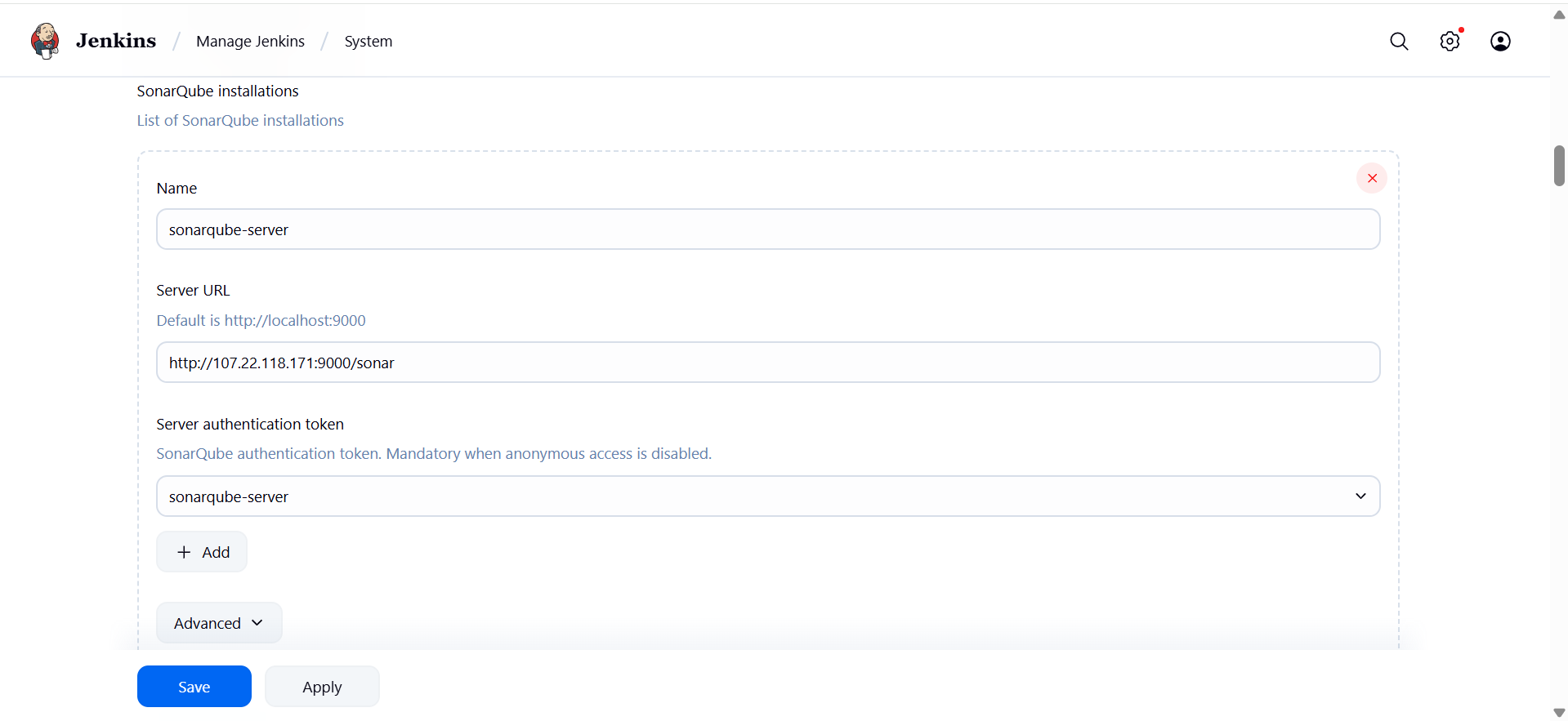
### 3.Generate Token from SonarQube:

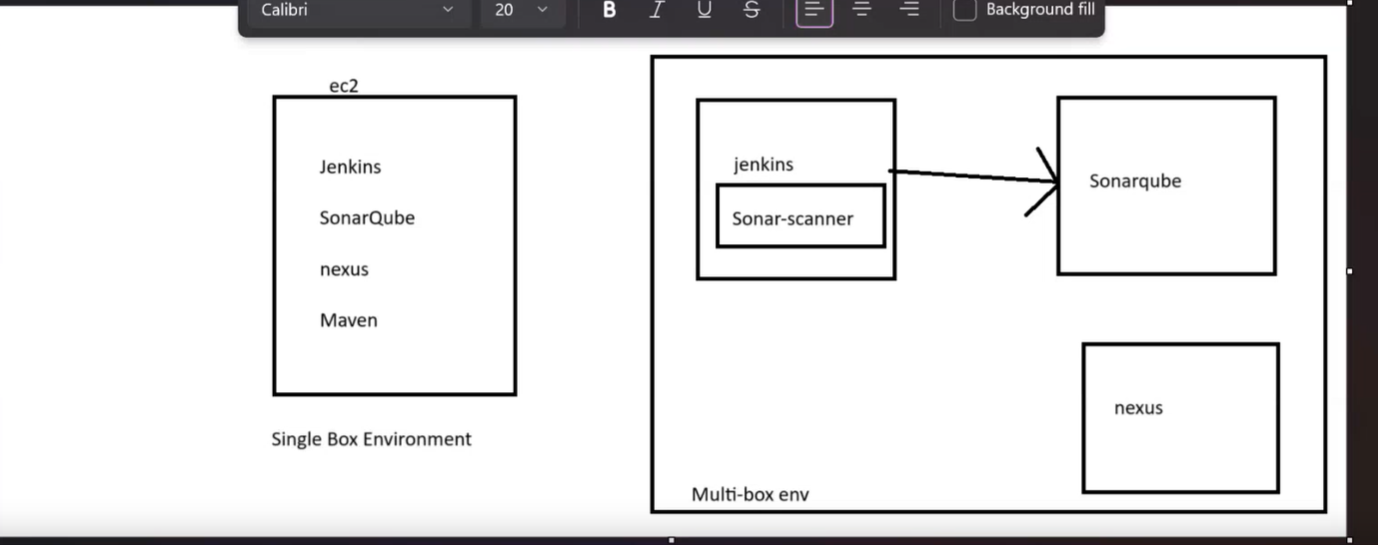
* Login to SonarQube → Top-right corner **My Account → Security → Generate Tokens**.

****

Copy the token and paste it into Jenkins.

****

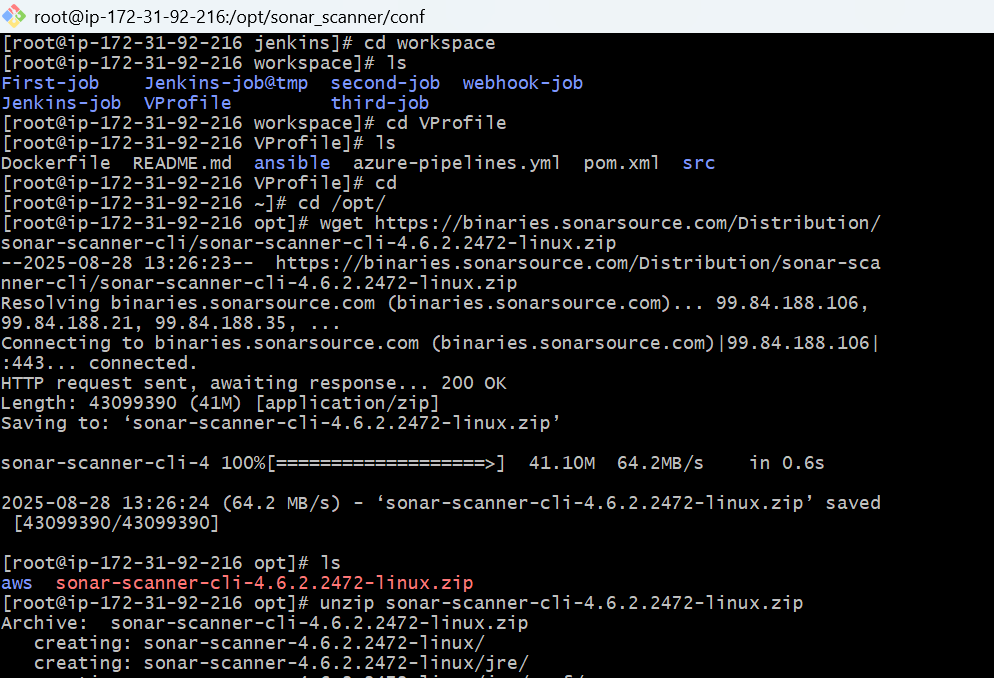
****

****

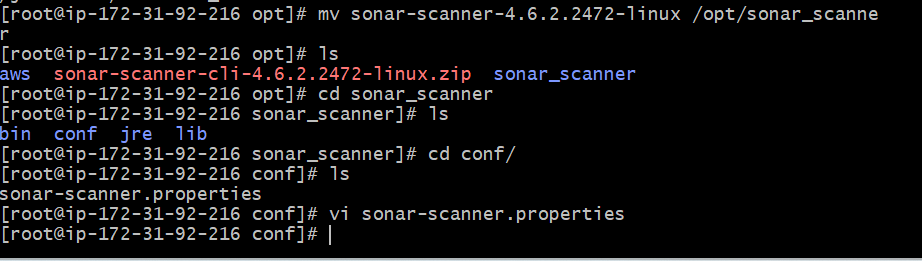
**4.Login to Jenkins server and install sonarqube scanner.**

**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**

****

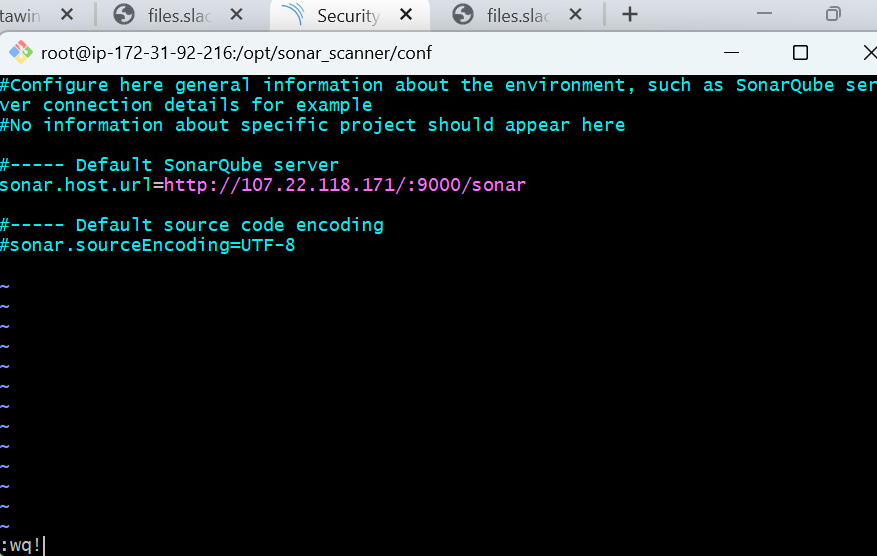
**# mv sonar-scanner-4.6.2.2472-linux /opt/sonar\_scanner**

****

**Set SonarQube server details in sonar-scanner property file**

**- Sonar properties file: /opt/sonar\_scanner/conf/sonar-scanner.properties**

**- sonar.host.url=http://`<SONAR\_SERVER\_IP>`:9000**

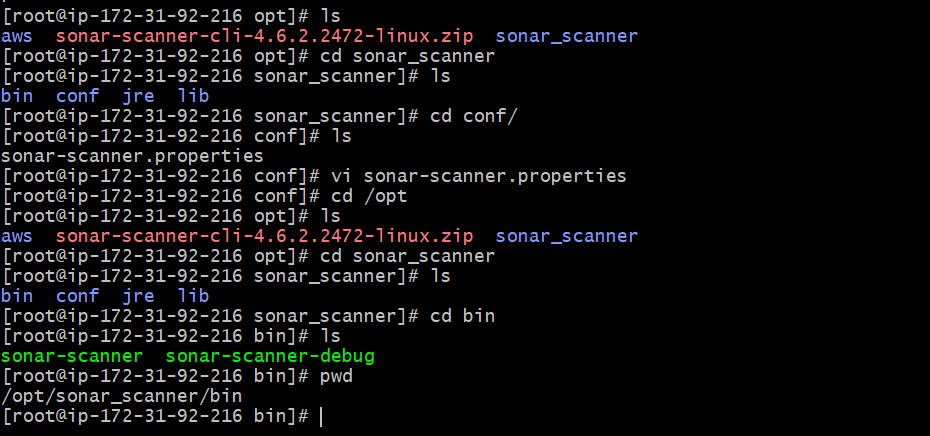
****

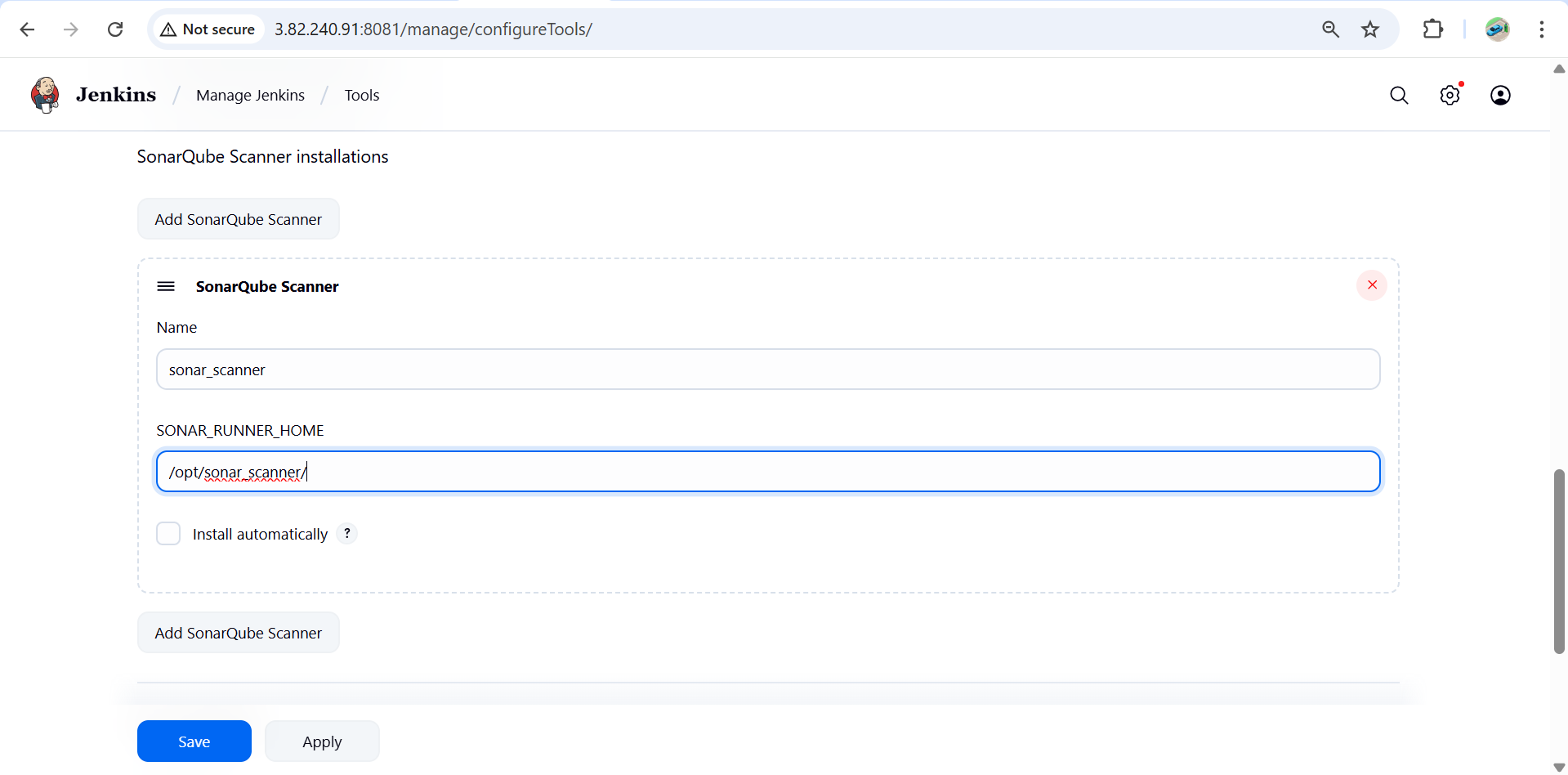
**5.Configure SonarQube Scanner in Jenkins**

**- `Manage Jenkins` > `Global Tool Configuration` > `SonarQube Scanner`**

**- Name : `sonar\_scanner`**

**- SONAR\_RUNNER\_HOME : `/opt/sonar\_scanner`**

****

****

**6.Create a Jenkins Job to Test SonarQube**

Alredy we have created a VProfile Job

Configure that job

Create a job to test SonarQube. Provide below sonar properties details in the job under build

-SCM:

<https://github.com/betawins/VProfile-1.git>

- Build:

- `Invoke top-level Maven targets`

`Goals : clean install`

- `Execute SonarQube Scanner` > `Analysis properties` (it is mandatary).

```sh

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/

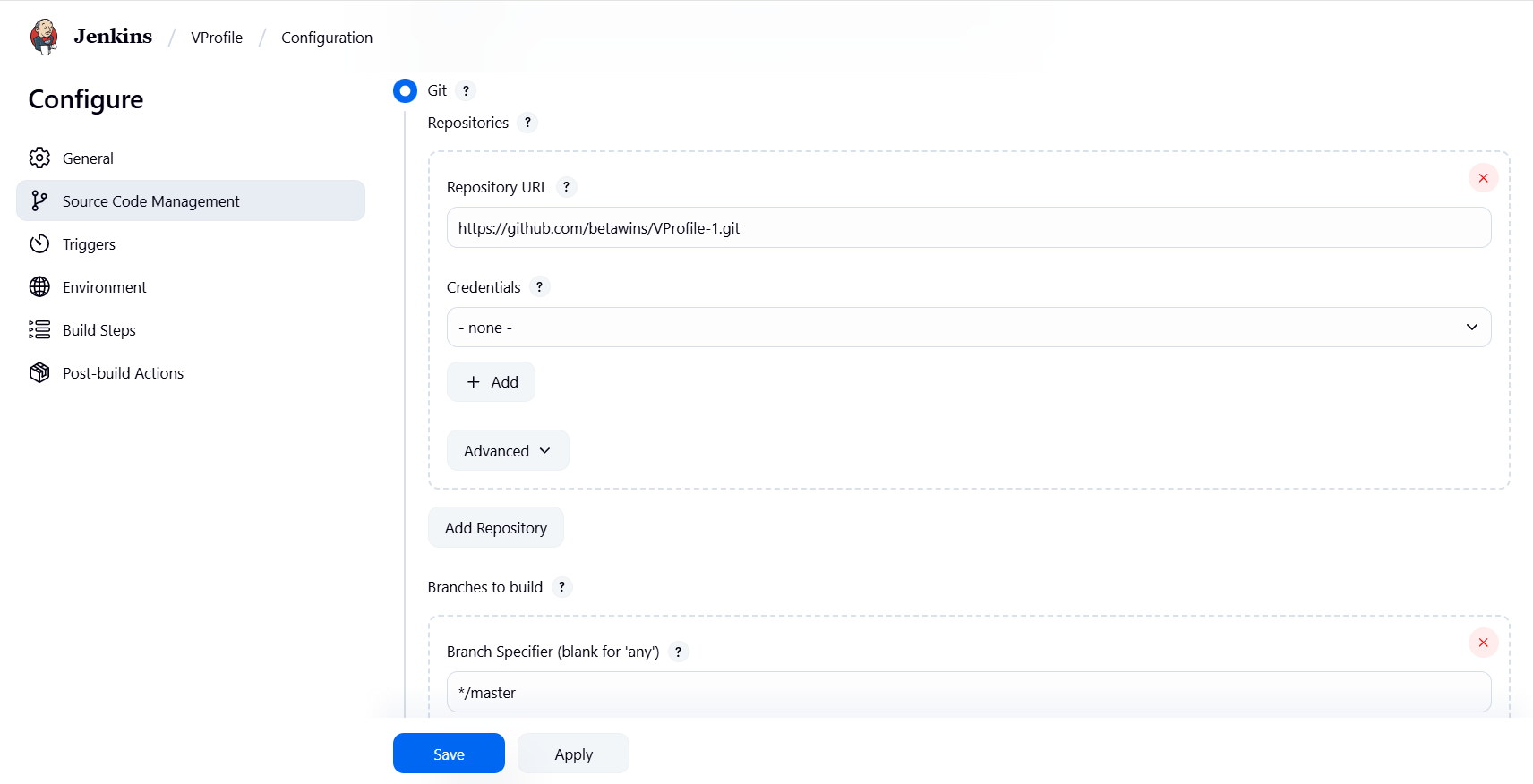
sonar.junit.reportsPath=target/surefire-reports

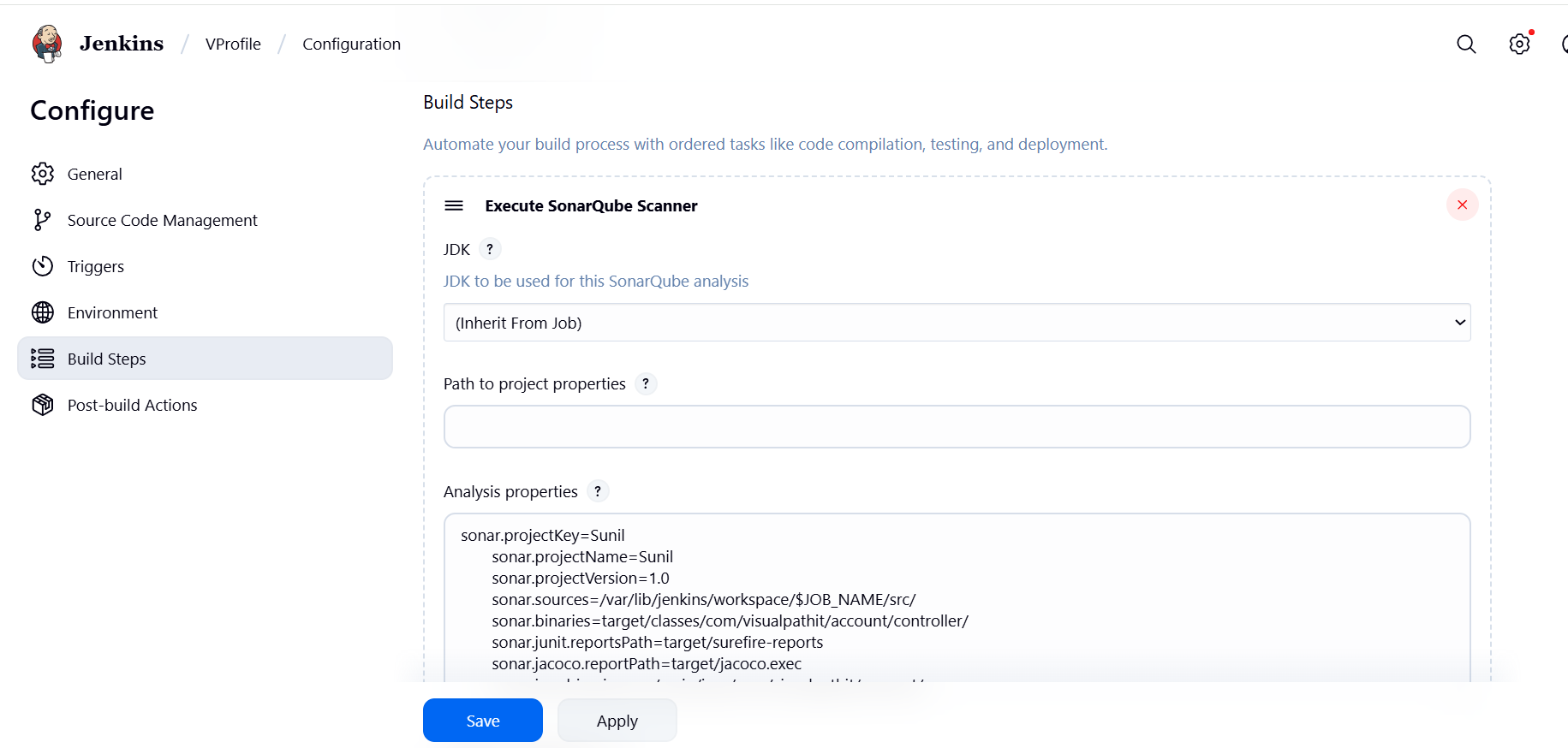
sonar.jacoco.reportPath=target/jacoco.exec

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

```

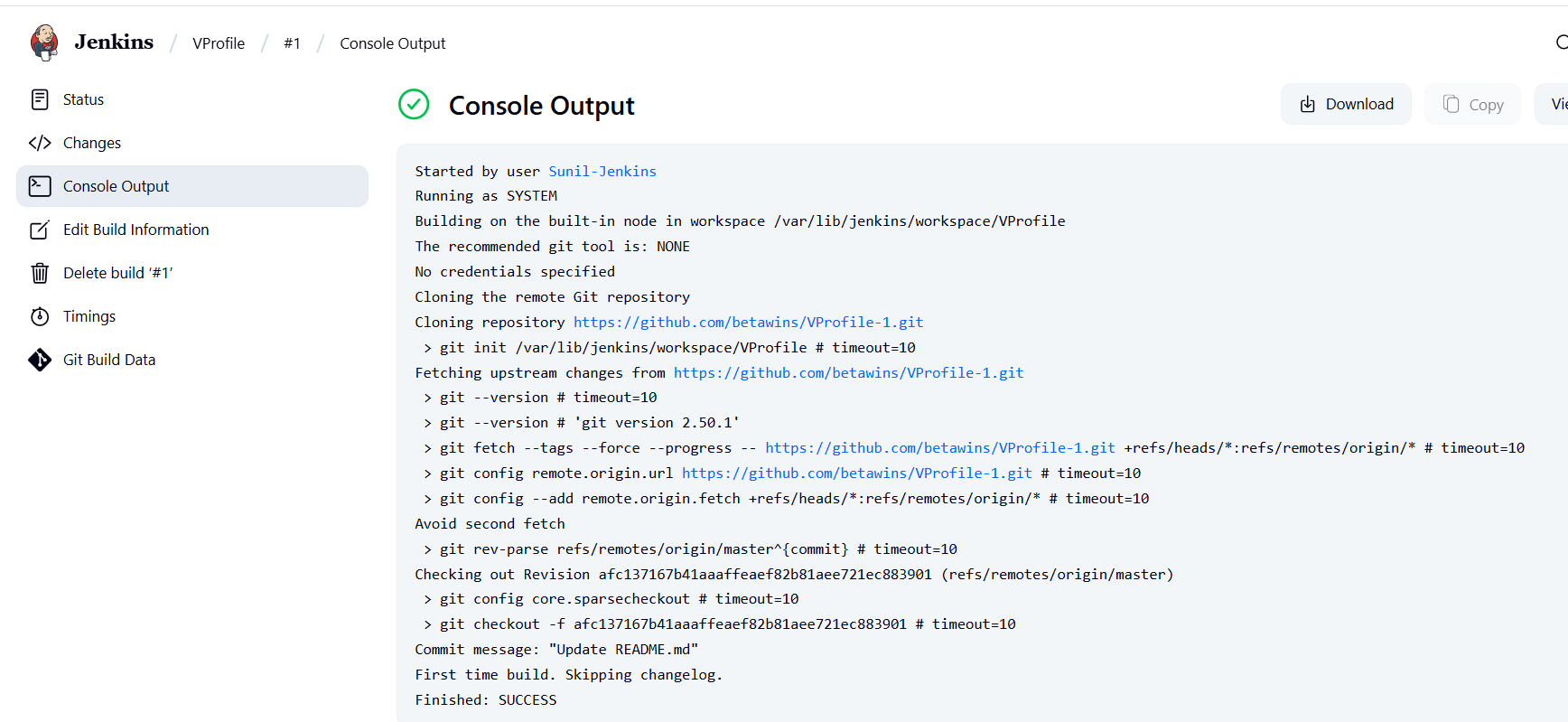
Execute job to get analysis report.

****

****

**7. Run the Jenkins Job**

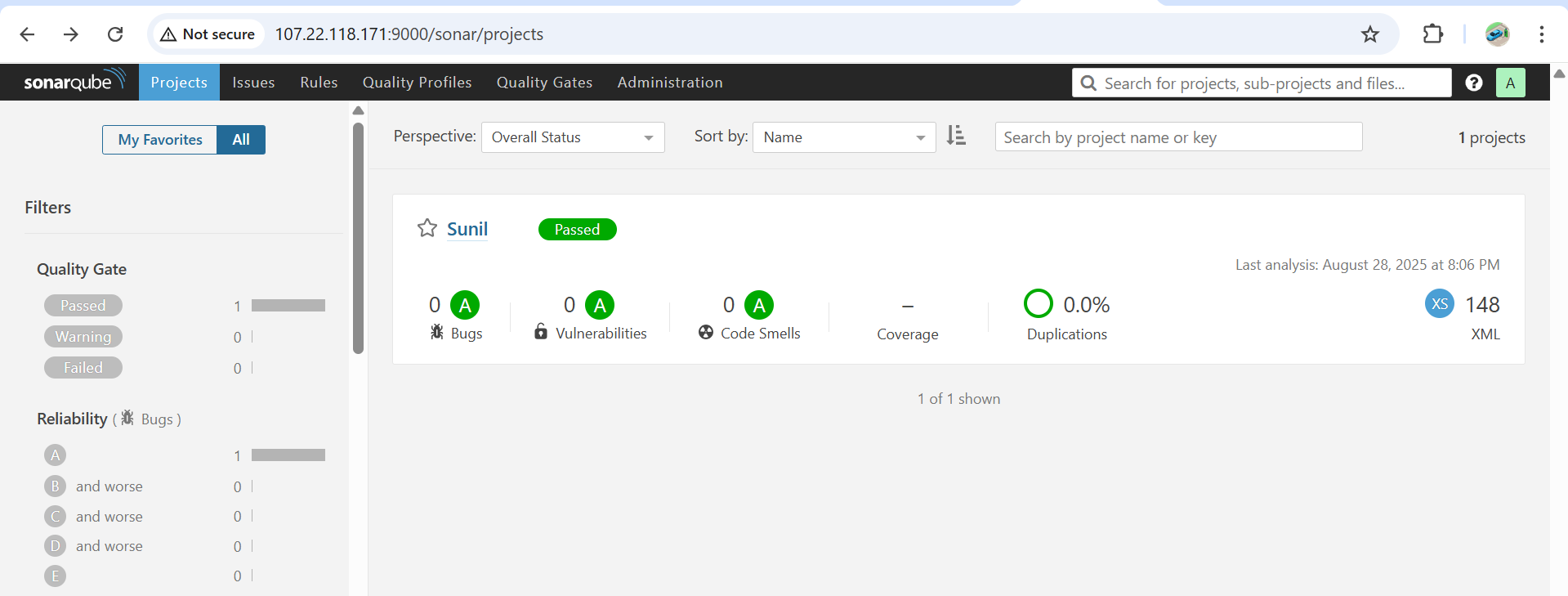
****

****

**8. Verify in SonarQube**

**Goto sonarqube and refresh**

**http://<Sonarqube\_server>:9000/sonar**

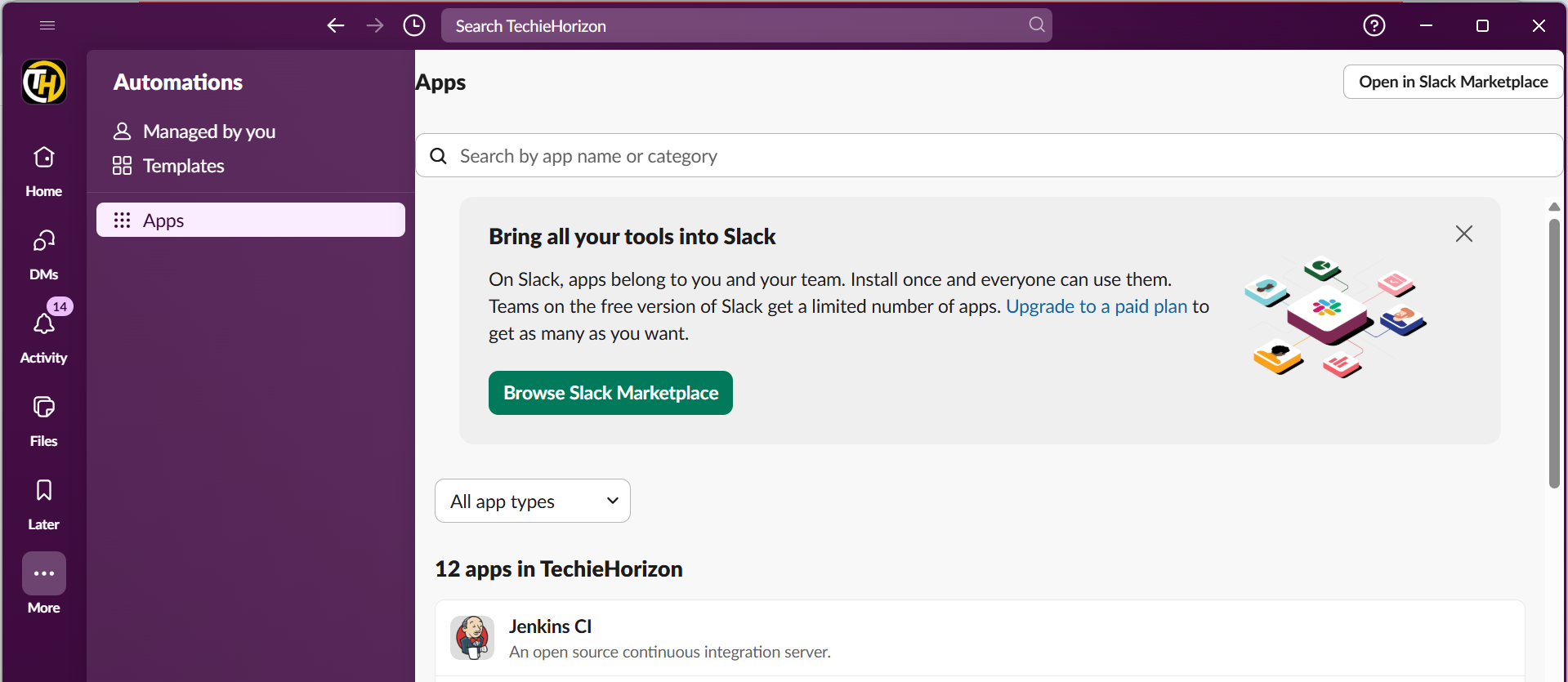
****

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

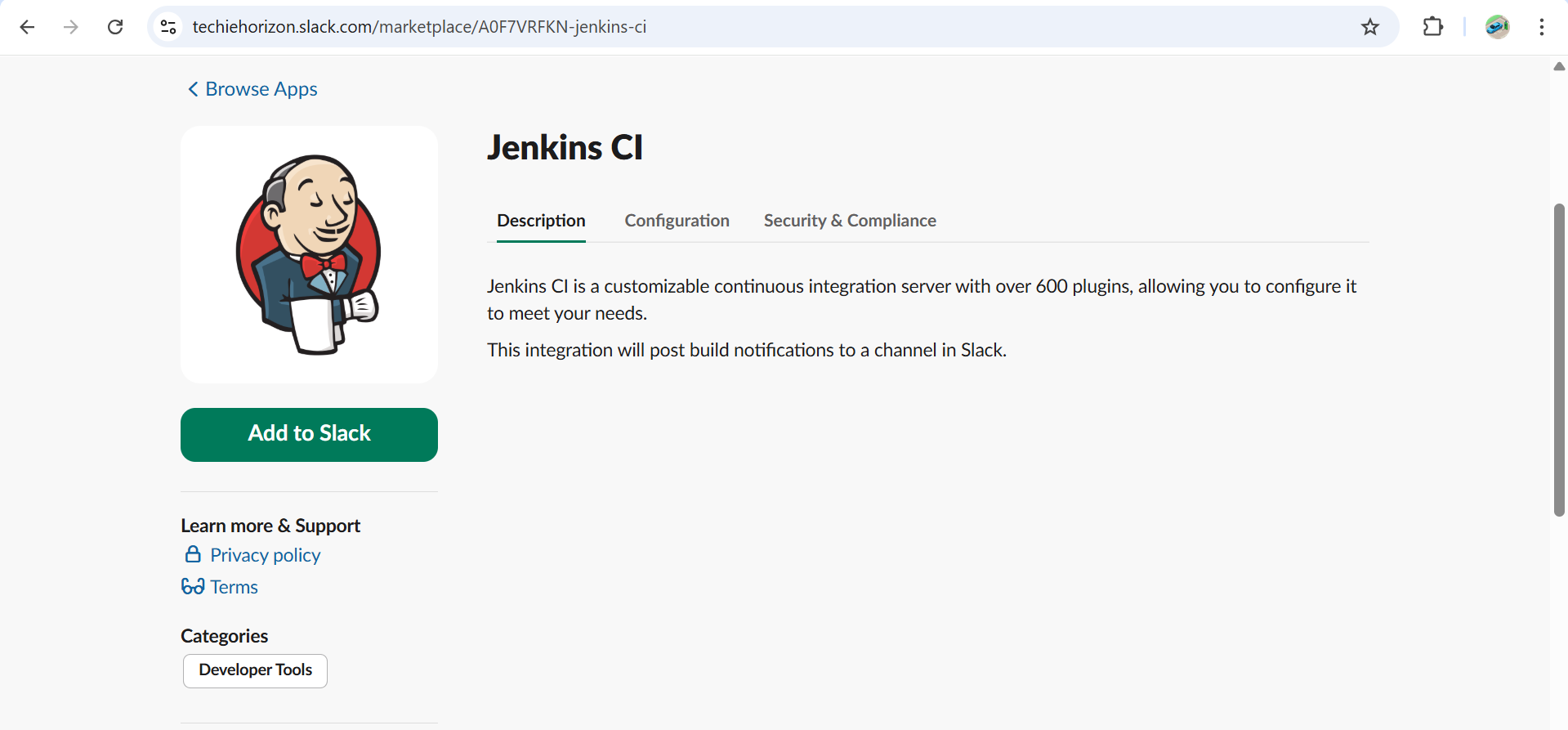
**1.Jenkins + Slack Integration**

Goto slack open apps

Select Jenkins cli

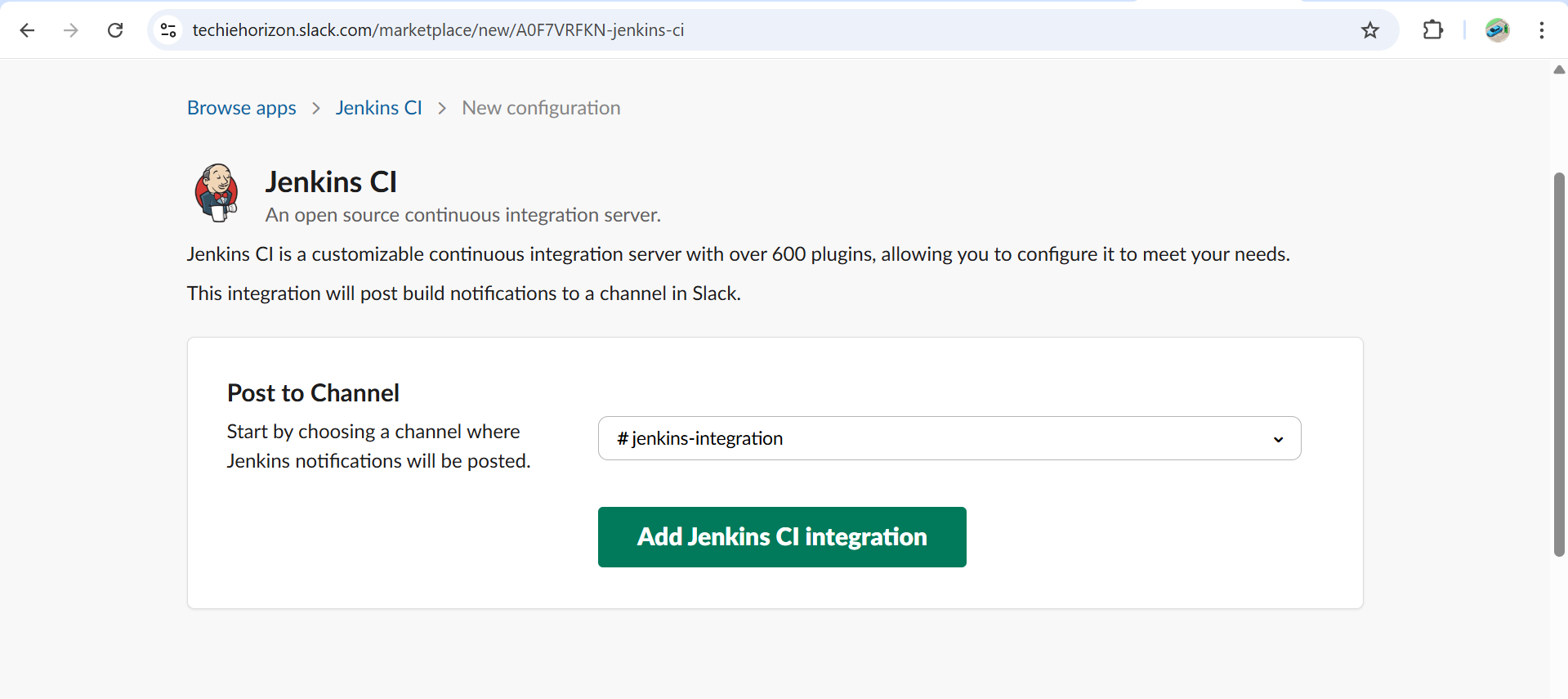
****

Click on Add to slack

****

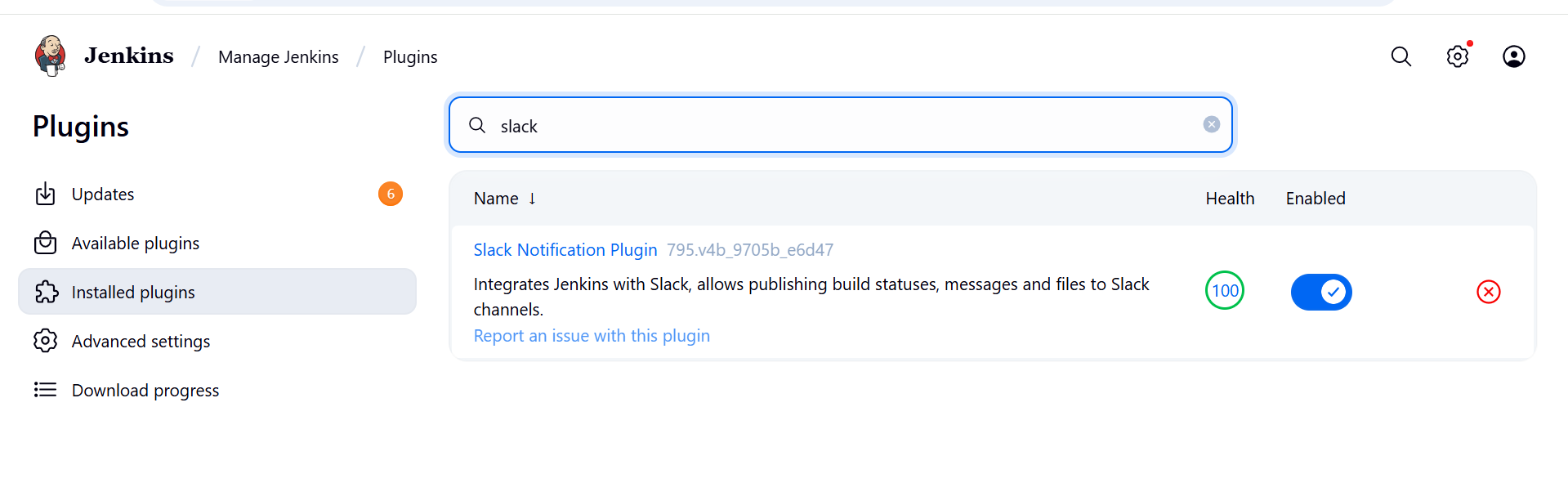
**Select the channel #jenkins-integration**

**And add**

****

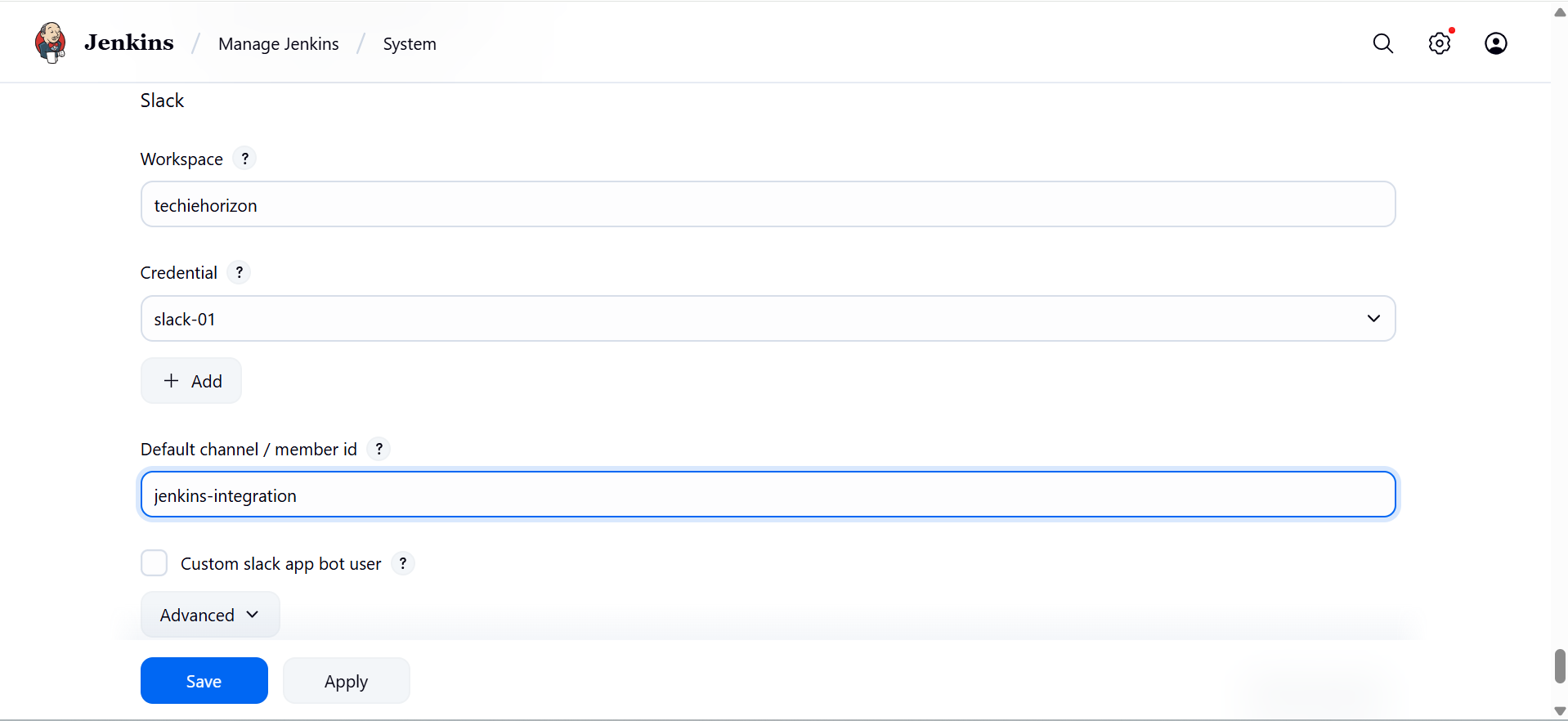
## 2. Install Slack Plugin in Jenkins

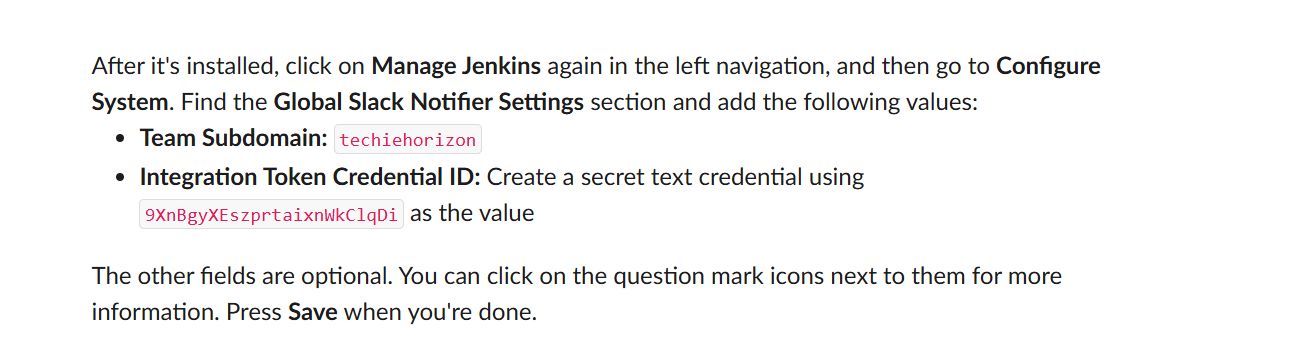
1. Go to Jenkins GUI:  
   **Manage Jenkins → Manage Plugins → Available → Search “Slack Notification” → Install without restart**.
2. Once installed, go to:  
   **Manage Jenkins → Configure System → Slack**.



## 3. Configure Slack in Jenkins

* **Workspace:** your Slack team domain (e.g., myworkspace.slack.com).
* **Integration Token Credential ID:** Add a new credential → Kind = “Secret Text” → Paste URL.
* **Default Channel:** #jenkins-integration
* Save.

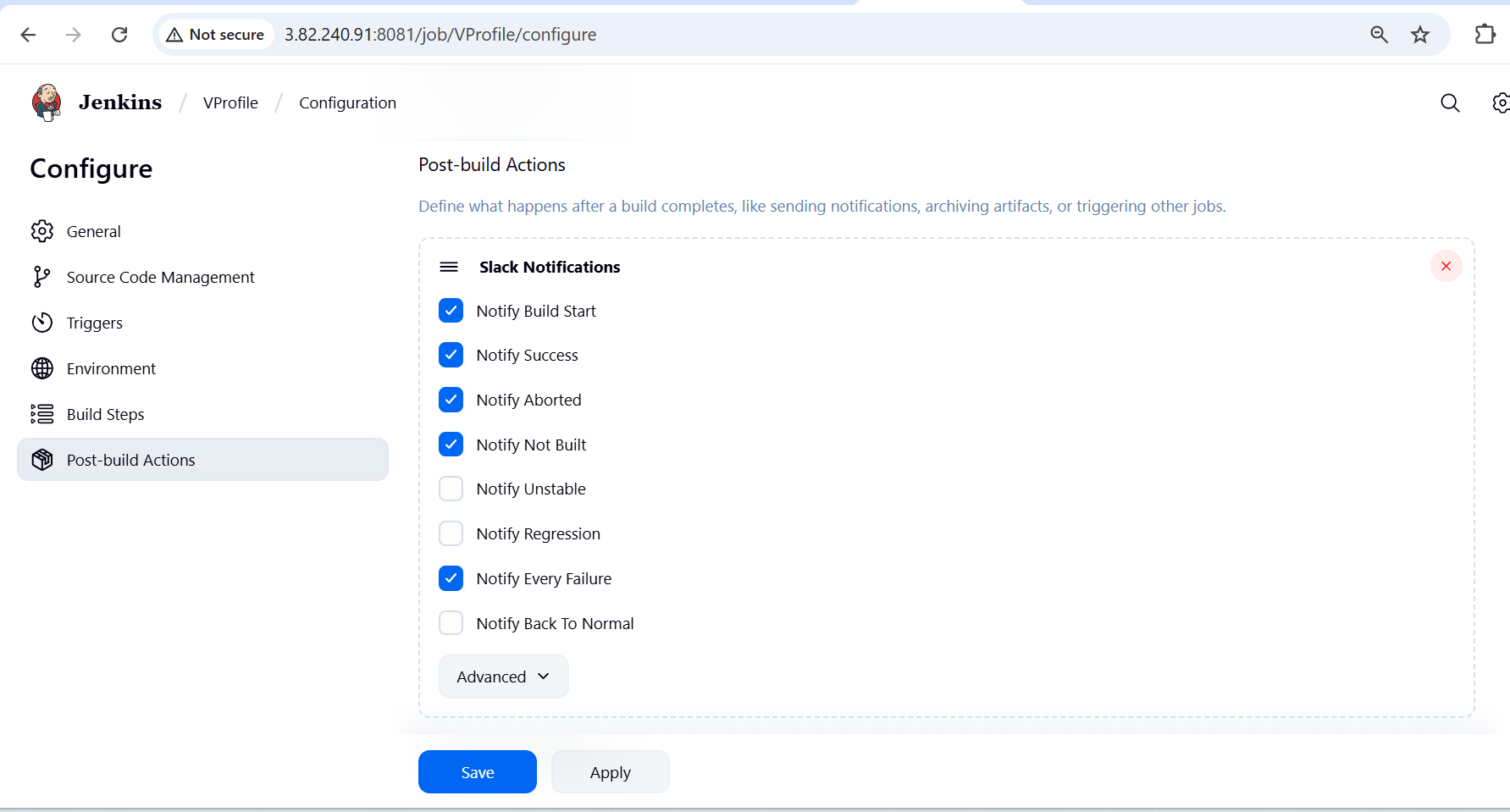


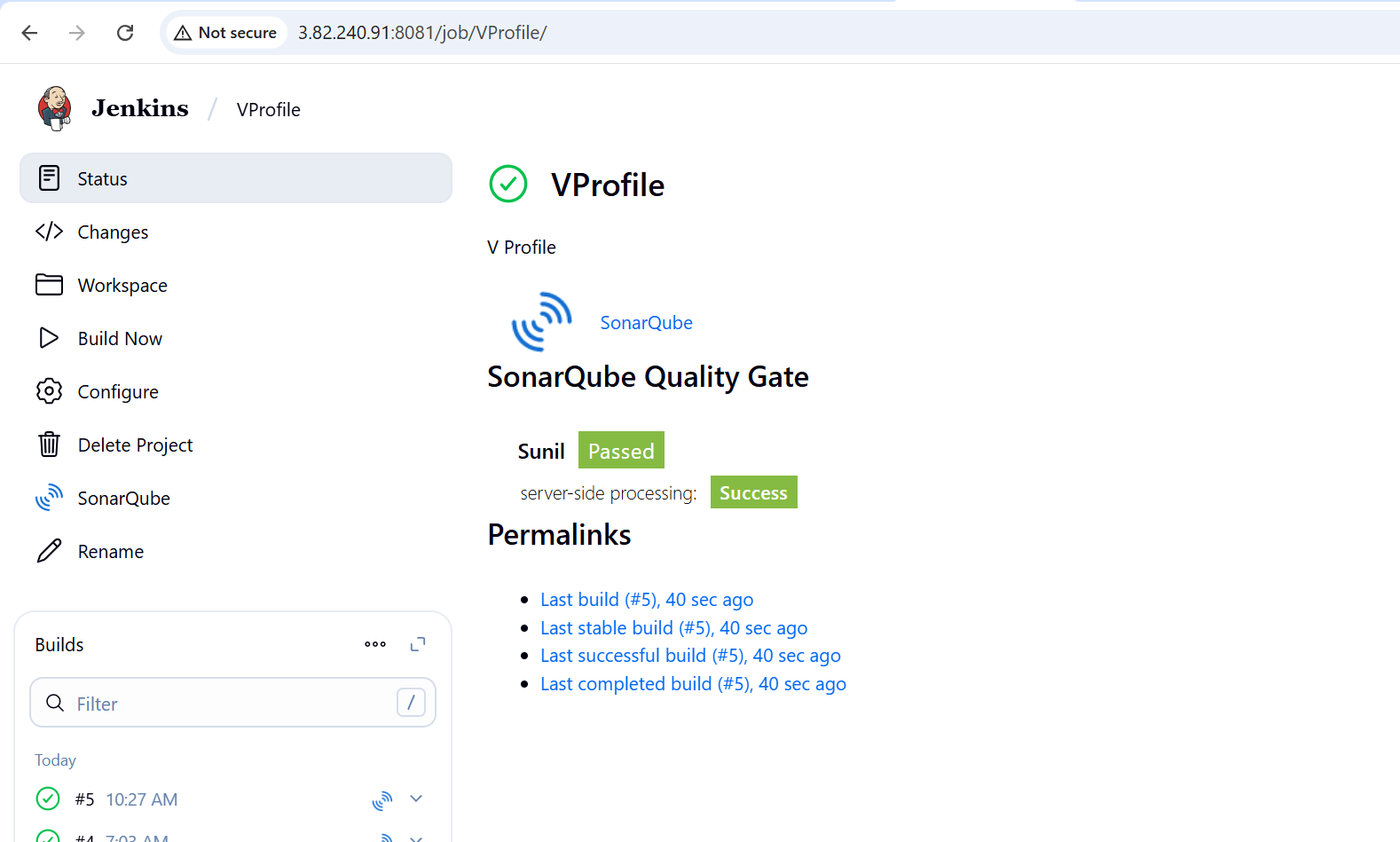
****

## 4. Add Slack Notifications to Jenkins Job

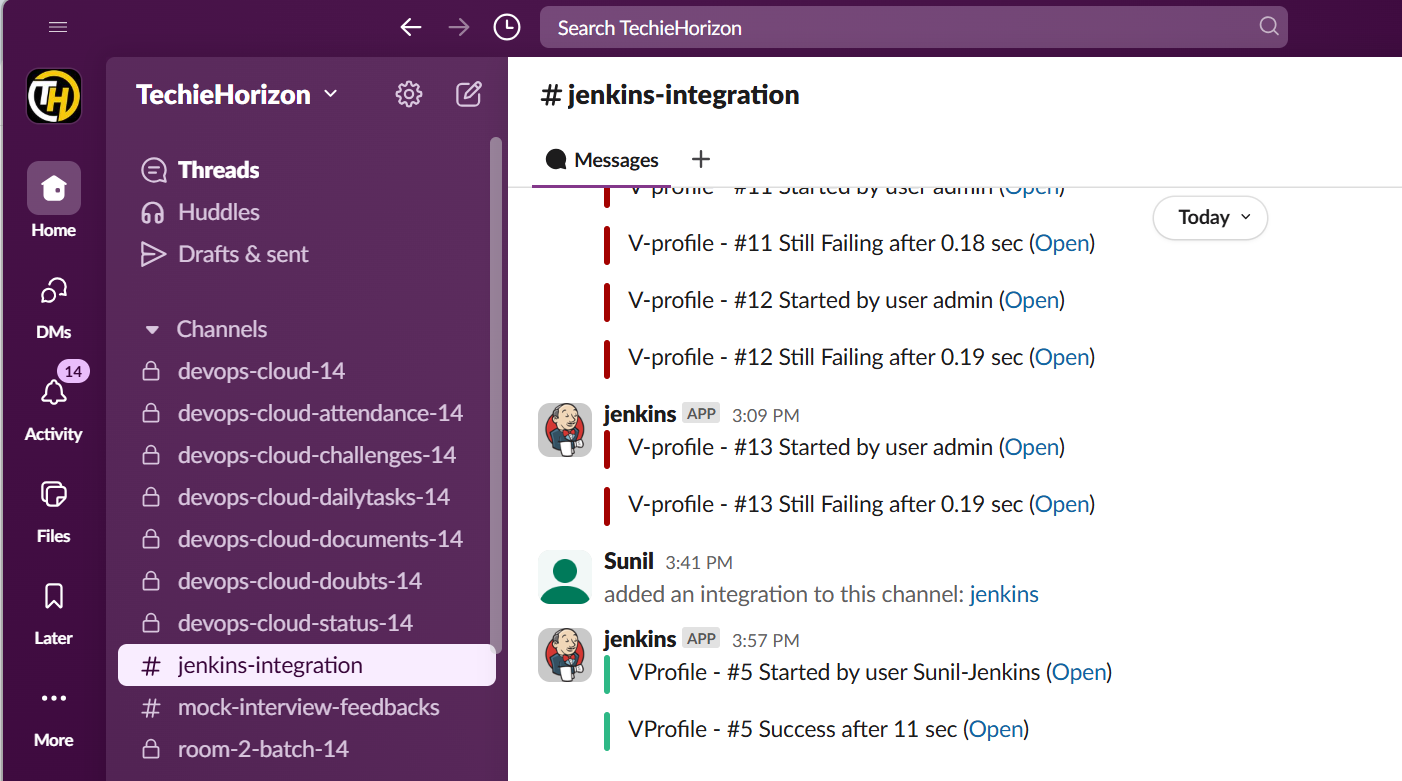
Open your existing **SonarQube Jenkins job** → **Post-build Actions**:

1. Select **Add post-build action** → **Slack Notifications**.
2. Choose when to notify:
   * Notify Success
   * Notify Failure
   * Notify Unstable Builds
   * Notify Back to Normal
3. Save job.





Now when you build, Jenkins will push notifications to Slack.

****

**2) 1) Create one jenkins job using the below code and create three stages.**

**stage1: Git clone to download the source code.**

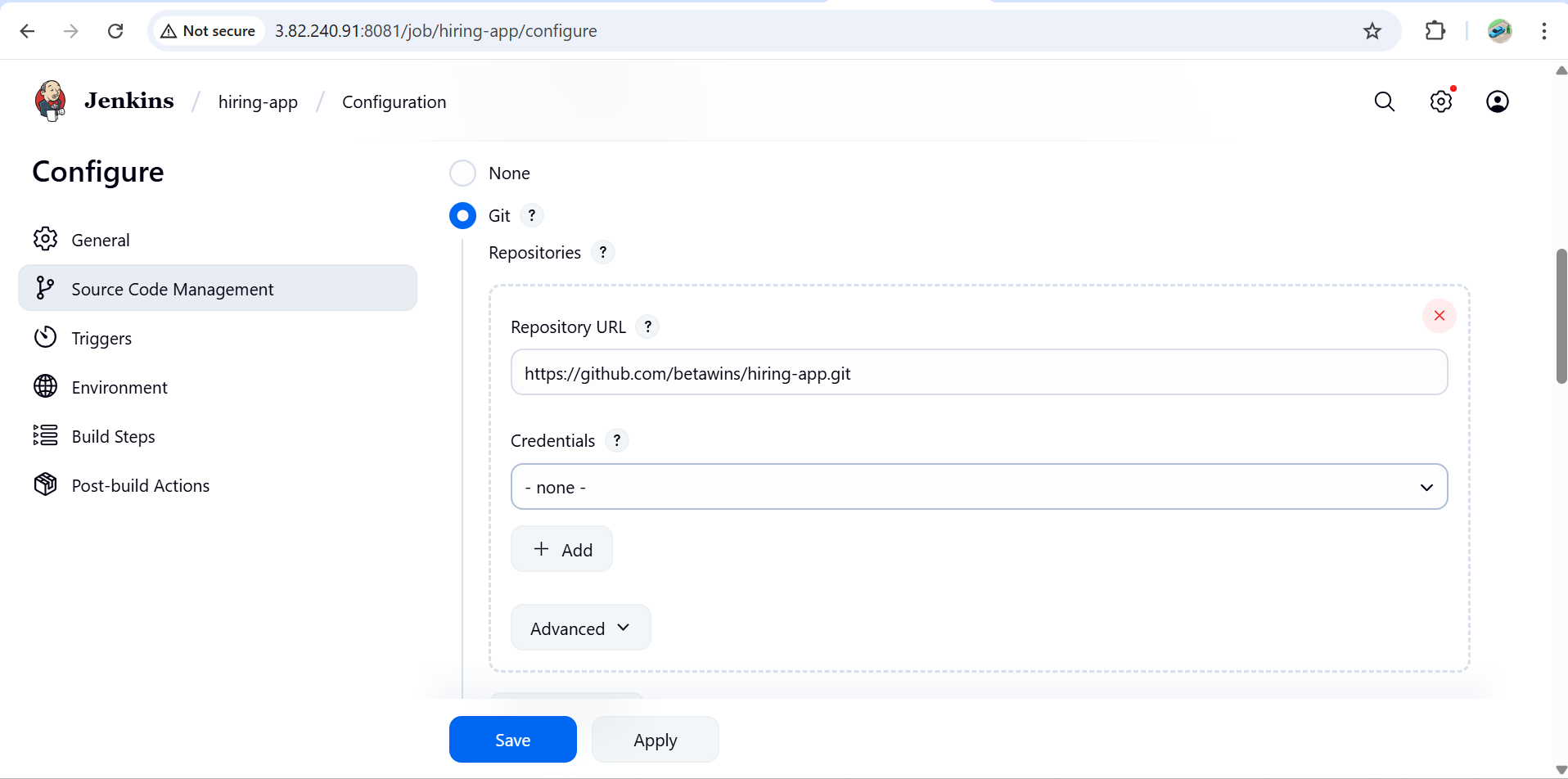
**stage2: Sonarqube Integration to check the quality of code**

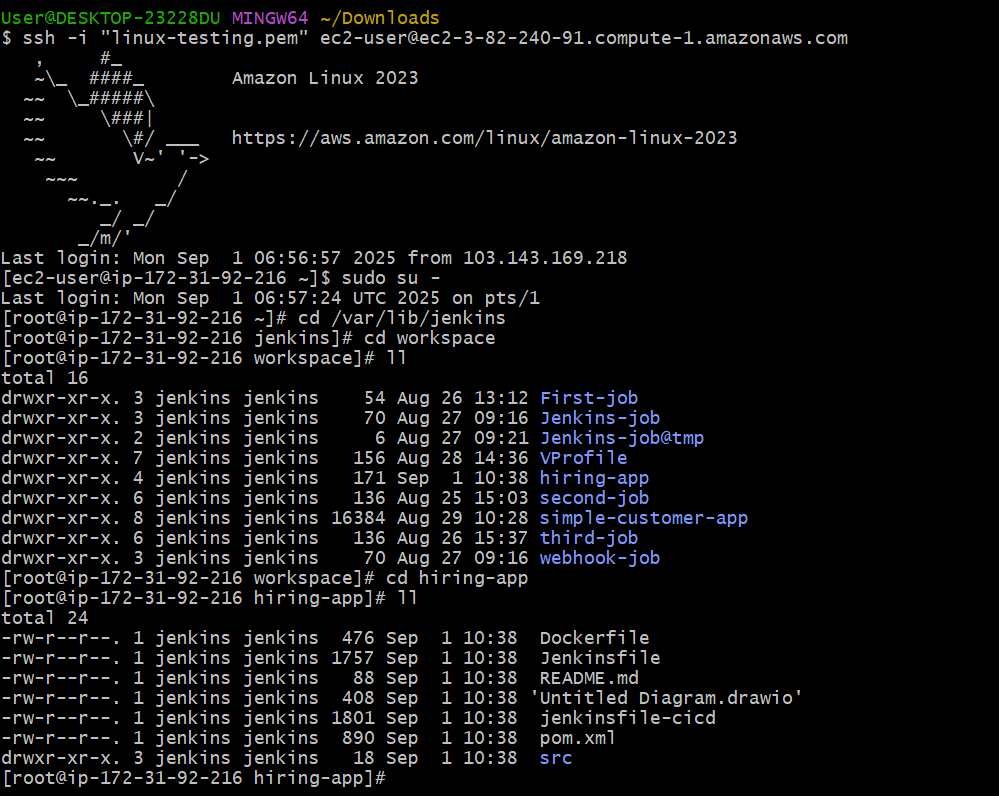
**stage3: Slack Integration to send the alerts to slack.**

**URL:** [**https://github.com/betawins/hiring-app.git**](https://github.com/betawins/hiring-app.git)

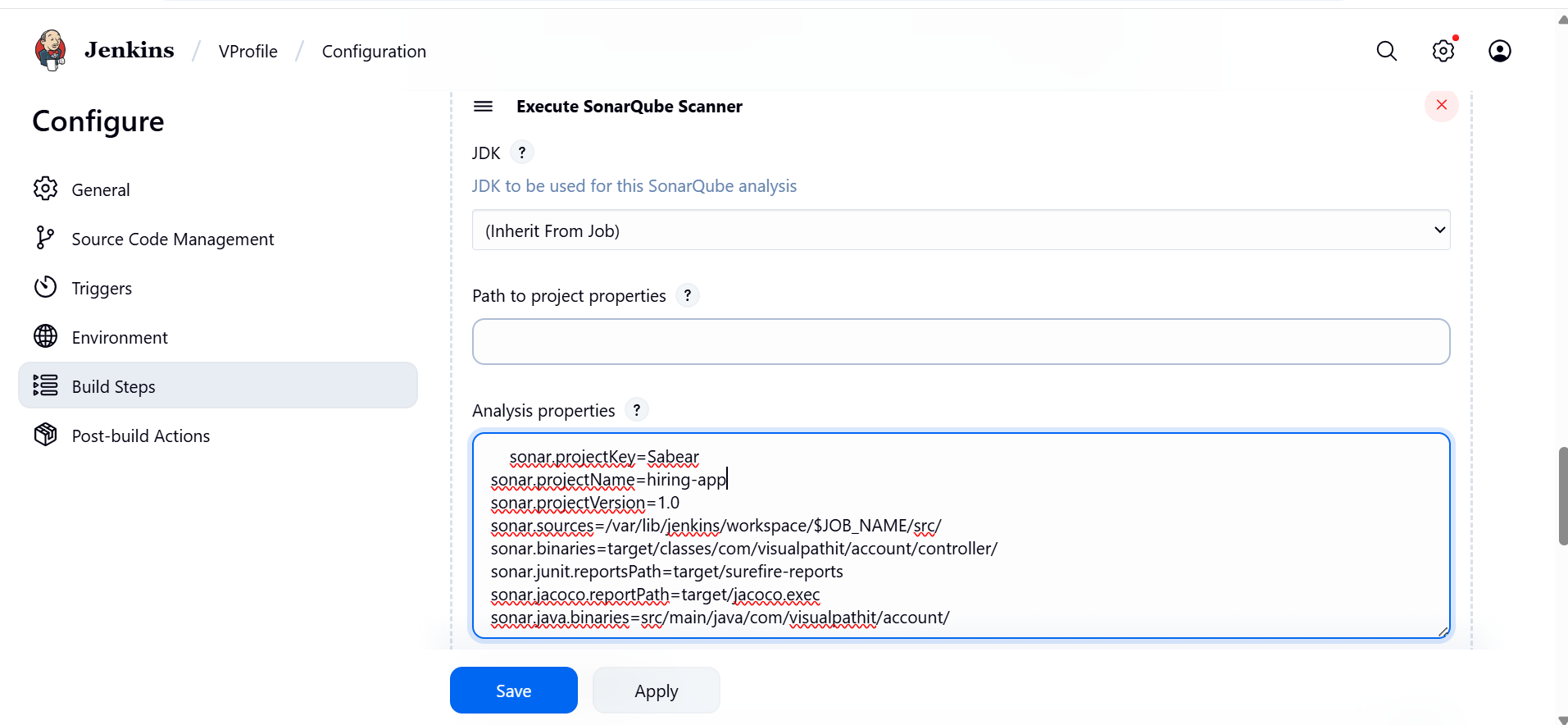
***stage1: Git clone to download the source codes***

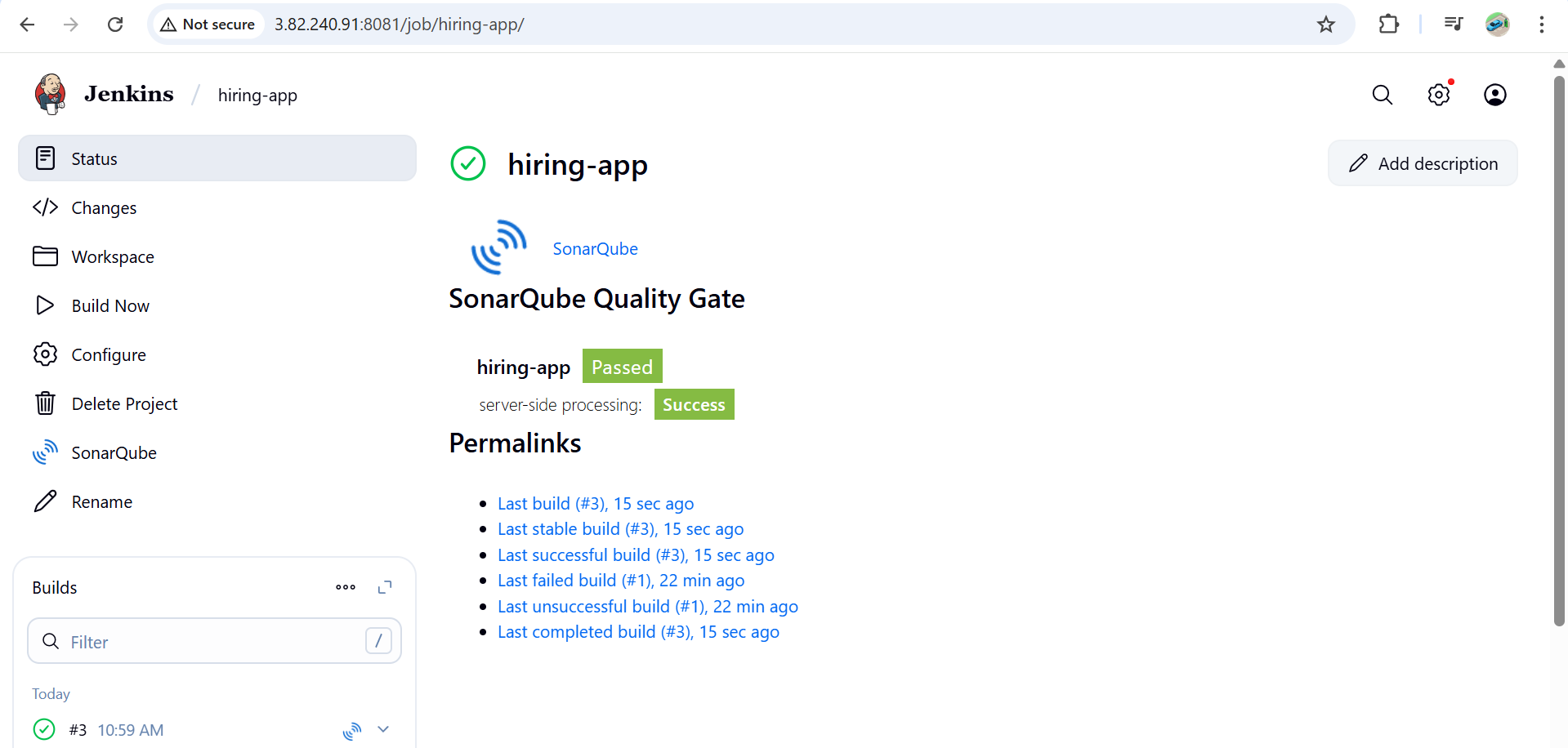
**Follow the similar steps and configurations as above task 1**

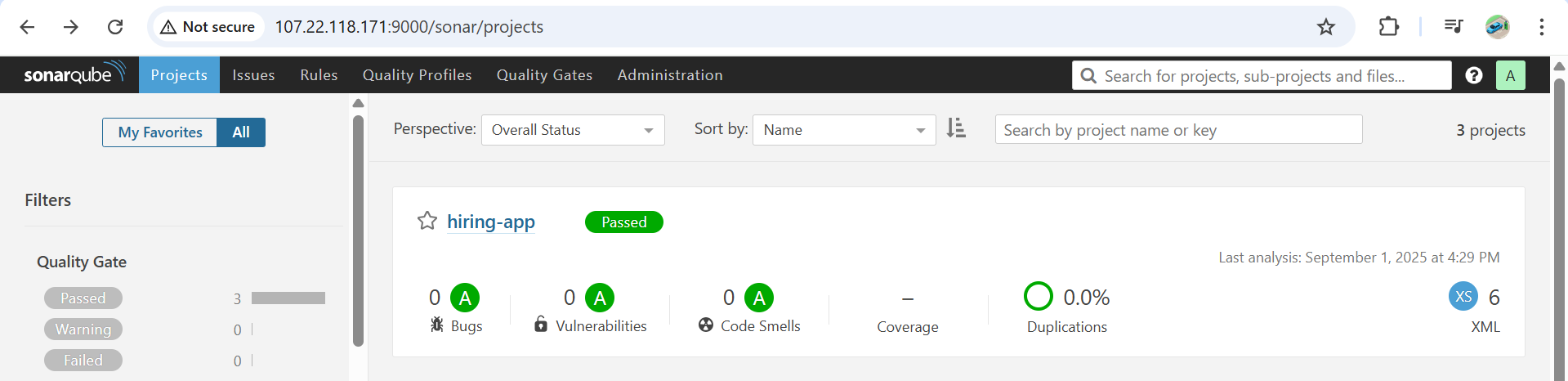
****

****

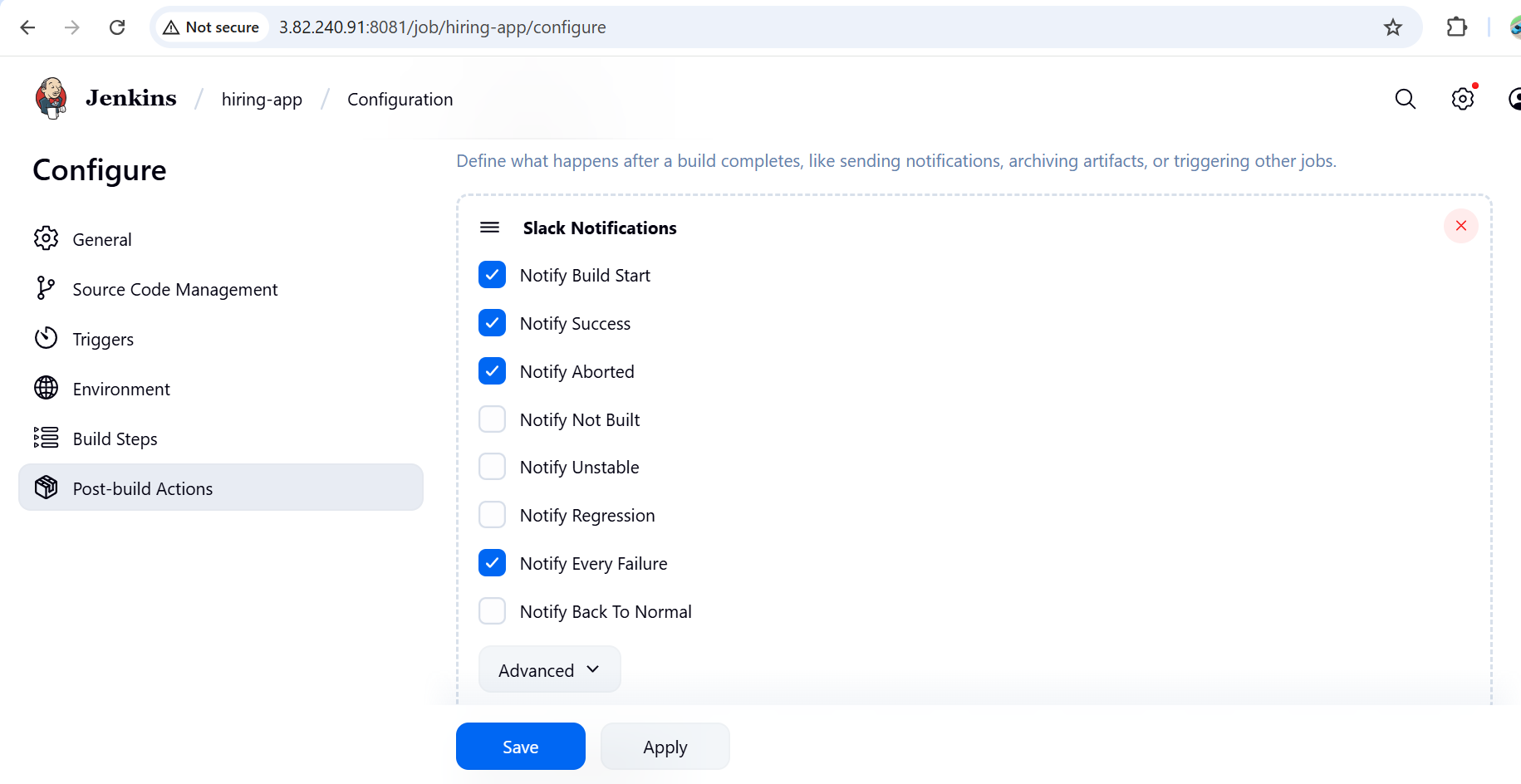
**stage2: Sonarqube Integration to check the quality of code**

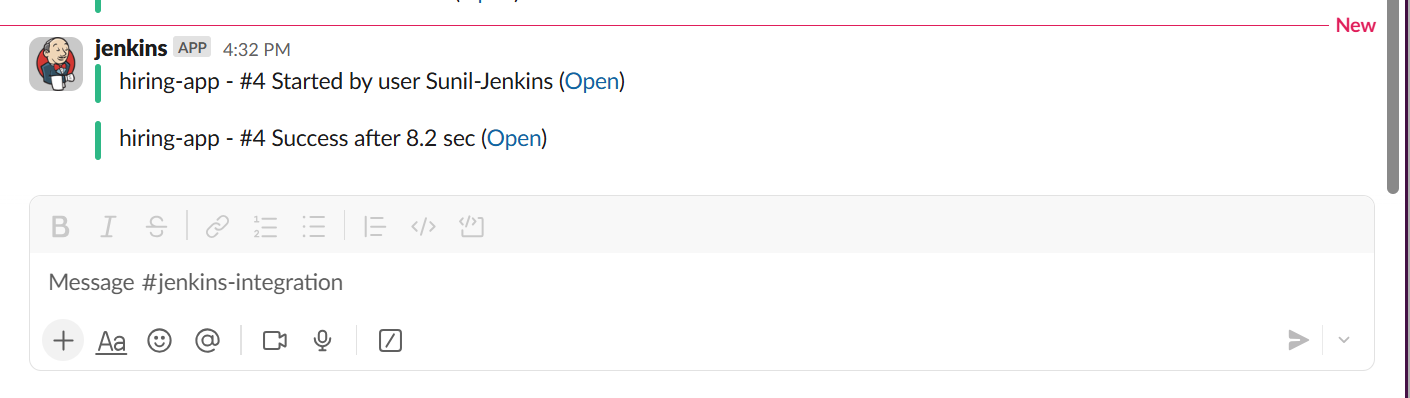
****

****

****

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

****

****