**Jenkins task-3**

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](https://github.com/betawins/VProfile-1.git" \t "https://app.slack.com/client/T03TRQ064Q0/_blank)

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](https://github.com/betawins/hiring-app.git" \t "https://app.slack.com/client/T03TRQ064Q0/_blank)

IMG_256**GitHub**

**[GitHub - betawins/VProfile-1: The purpose of the research project is for the students to learn how to make CI,CD pipeline by using jenkins and scripting techniques taught in class.](https://github.com/betawins/VProfile-1.git" \t "https://app.slack.com/client/T03TRQ064Q0/_blank)**

The purpose of the research project is for the students to learn how to make CI,CD pipeline by using jenkins and scripting techniques taught in class. - GitHub - betawins/VProfile-1: The purpose o... (49 kB)

IMG_257**GitHub**

**[GitHub - betawins/hiring-app](https://github.com/betawins/hiring-app.git" \t "https://app.slack.com/client/T03TRQ064Q0/_blank)**

Contribute to betawins/hiring-app development by creating an account on GitHub. (33 kB)

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**

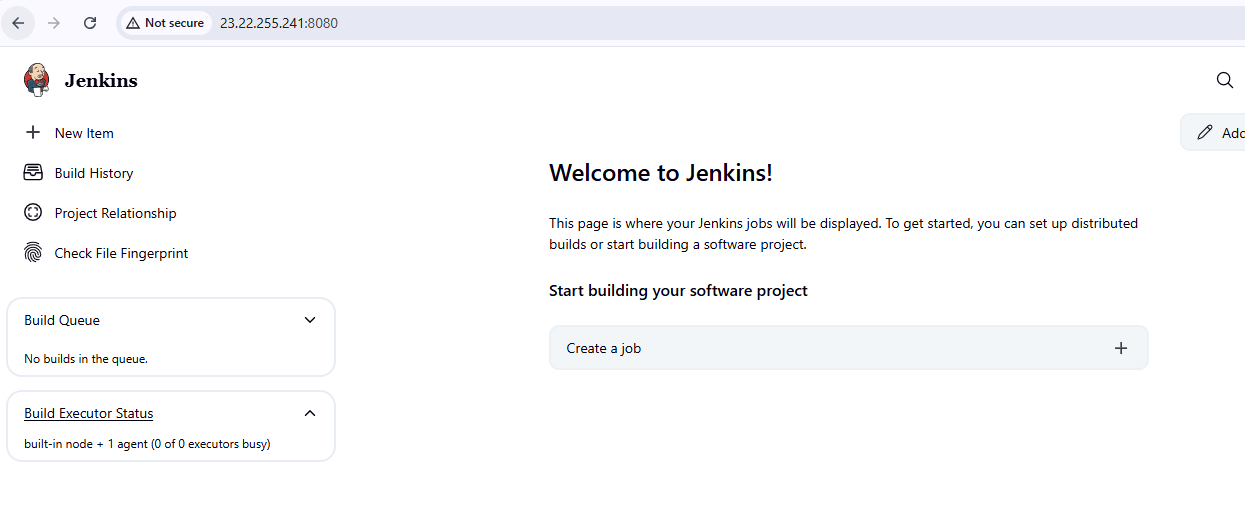
**stage3: Slack Integration to send the alerts to slack.**

**URL: [https://github.com/betawins/VProfile-1.git](https://github.com/betawins/VProfile-1.git" \t "https://app.slack.com/client/T03TRQ064Q0/_blank)**

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

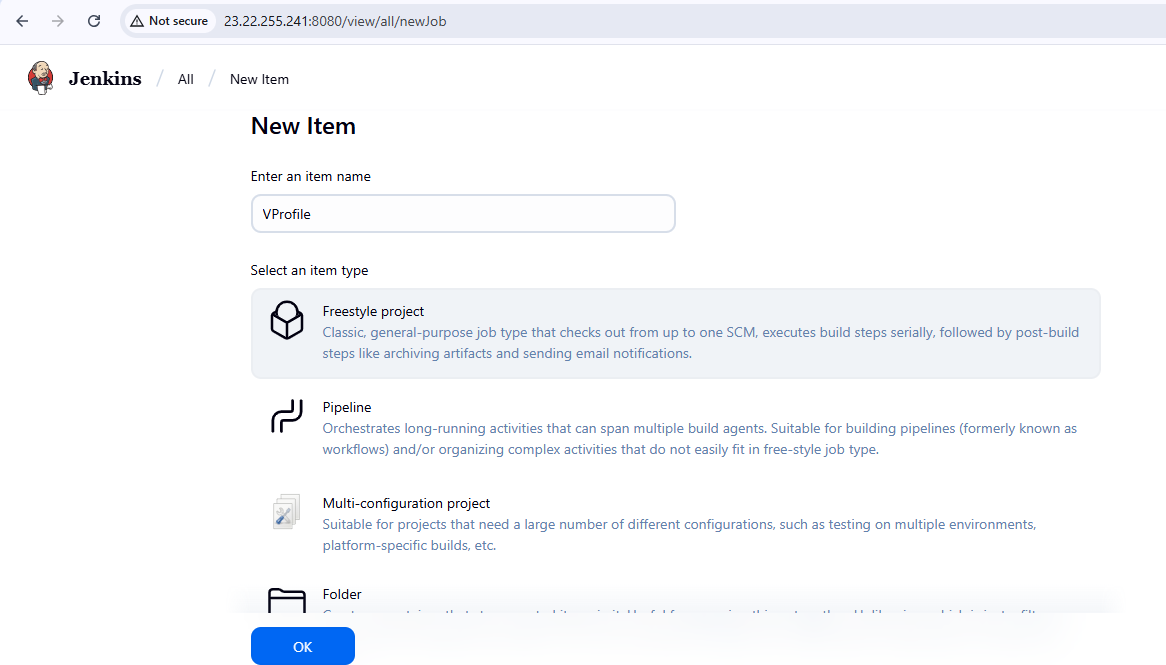
**Step 1: Open Jenkins Dashboard**

* 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**.

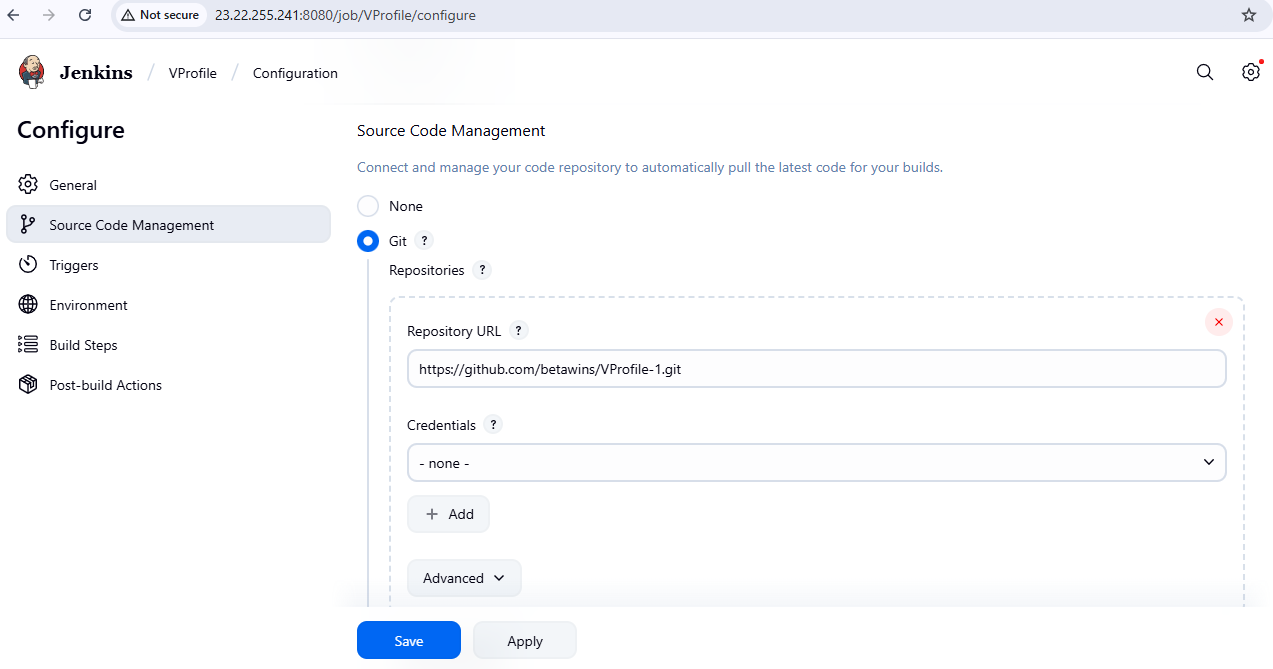


### Step 3: Configure Git Repository

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

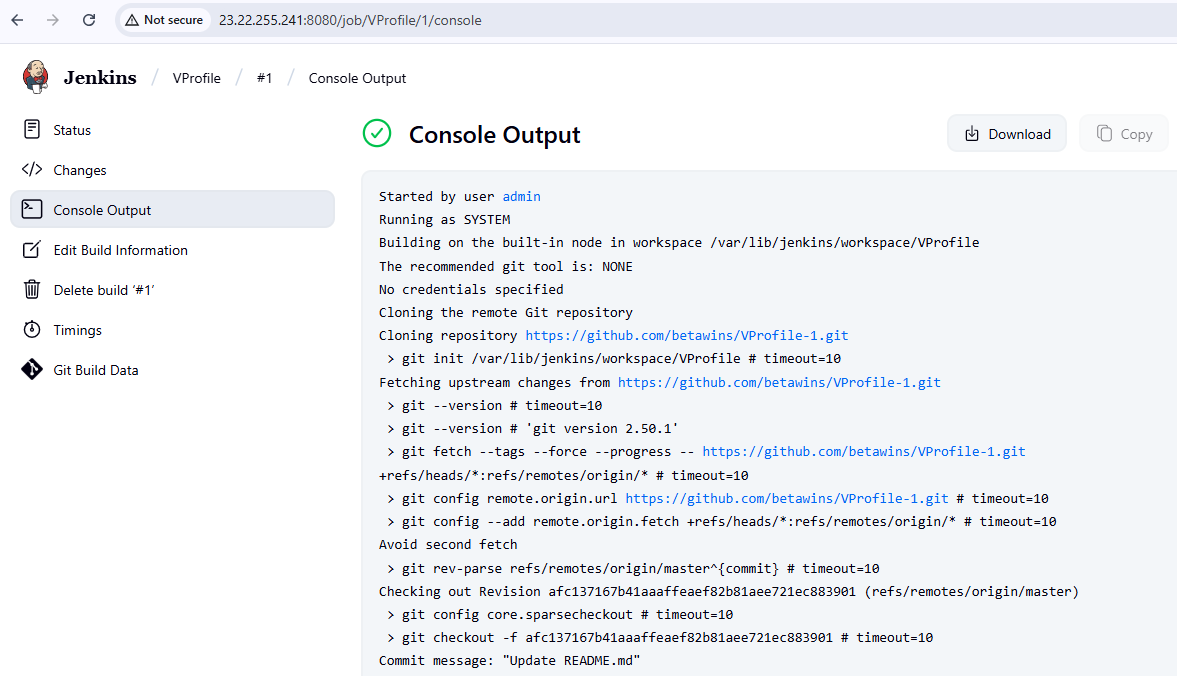
**<https://github.com/betawins/VProfile-1.git>**

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.



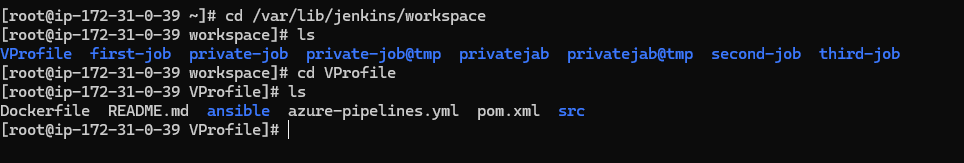
**Now successfully we have created the VProfile job and cloned the source code from github using the url**

**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**

# SonarQube Installation

SonarQube provides the capability to not only show health of an application but also to highlight issues newly introduced. With a Quality Gate in place, you can fix the leak and therefore improve code quality systematically.

### Prerequisites

**1. EC2 instance with Java installed**

**2. Use t2.large with atleast 20gb memory to run sonarqube.**

**3. MySQL Database Server or MyQL RDS instance.**

### Installation

Install java 1.8 version

```sh

yum install java-1.8\*

```

Add mysql rpm Repository

```sh

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

```

Start MySQL and Enable Start at Boot Time

```sh

systemctl start mysqld

systemctl enable mysqld

```

Check if mysql is running or not

```sh

netstat -na | grep 3306

```

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

```

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

```sh

mysql -u root -p

```

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

```

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

- 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`

# Sonar version 7 and higher cannot be run using root user,please switch to any other user and change the permissions to sonar directory and start sonar.

Create new user

```sh

# useradd sabear

# passwd sabear

```

Add user sabear to sudoers file

```sh

vi /etc/sudoers

## Same thing without a password

# %wheel ALL=(ALL) NOPASSWD: ALL

sabear ALL=(ALL) NOPASSWD: ALL

```

Restart the sshd service

```sh

# systemctl restart sshd

```

Switch to newly created user

```sh

# sudo su sabear

```

Start SonarQube service

```sh

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

# ./sonar.sh start

```

##### Run SonarQube as a default service

Implement SonarQube server as a service

```sh

Copy sonar.sh to etc/init.d/sonar and modify it according to your platform.

# sudo cp /opt/sonar/bin/linux-x86-64/sonar.sh /etc/init.d/sonar

# sudo vi /etc/init.d/sonar

```

Add below values to your /etc/init.d/sonar file

```sh

Insert/modify below values

SONAR\_HOME=/opt/sonar

PLATFORM=linux-x86-64

WRAPPER\_CMD="${SONAR\_HOME}/bin/${PLATFORM}/wrapper"

WRAPPER\_CONF="${SONAR\_HOME}/conf/wrapper.conf"

PIDDIR="/var/run"

```

Start SonarQube server

```sh

# service sonar start

```

SonarQube application uses port 9000. access SonarQube from browser

```sh

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

```

Default credentials for sonarqube.

Username : admin

Password: admin

### NOTES

1) Check whether you enabled port 9000 in EC2 instance security group

### Important Points:

1) mysql port number : 3306

2) sonarqube port: 9000

3) sonarqube logs : /opt/sonar/logs

4) We can find four different logs

a)access.log

b)es.log aka elastic search

c)sonar.log

d)web.log