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>

STEPS:

**Update System Packages:**

sudo yum update -y

**Install Java (OpenJDK 11 - Amazon Corretto):**

sudo yum install java-11-amazon-corretto -y

# Verify Java version

java -version

**Install MySQL Client (if not already present):**

sudo yum install mysql -y

**Configure MySQL Database (on your MySQL server):**

**(Connect to your MySQL server as root or an admin user)**

CREATE DATABASE sonar CHARACTER SET utf8mb4 COLLATE utf8mb4\_general\_ci;

CREATE USER 'sonar'@'localhost' IDENTIFIED BY 'sonar'; -- Use a strong password!

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

FLUSH PRIVILEGES;

EXIT;

**Download and Extract SonarQube 7.8:**

# Navigate to /opt directory

cd /opt/

# Download SonarQube 7.8 (replace URL if different, verify latest 7.8 download link from SonarQube archives)

sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-7.8.zip

# Unzip the downloaded archive

sudo unzip sonarqube-7.8.zip

# Rename the extracted directory for easier access

sudo mv sonarqube-7.8 sonar

**Create SonarQube System User:**

sudo useradd sabear

sudo passwd sabear

# Enter and confirm your password for 'sabear'

**Grant Sudo Privileges to sabear (Optional but useful for troubleshooting):**

sudo vi /etc/sudoers

**(Add the following line below root ALL=(ALL) ALL)**

sabear ALL=(ALL) NOPASSWD: ALL

**save and exit vi**

sudo chown -R sabear:sabear /opt/sonar

**Edit SonarQube Configuration File: :**

sudo vi /opt/sonar/conf/sonar.properties

#----- MySQL

sonar.jdbc.username=sonar

sonar.jdbc.password=your\_db\_password

make sure mysql should (like version )then change it

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

Configure Web Server Port :

sonar.web.host=0.0.0.0

sonar.web.context=/sonar

sonar.web.port=9000

**System Service Setup:**

sudo vi /etc/init.d/sonar

**#!/bin/sh**

#

# Copyright (c) 1999, 2006 Tanuki Software Inc.

#

# Java Service Wrapper sh script. Suitable for starting and stopping

# wrapped Java applications on UNIX platforms.

#

#-----------------------------------------------------------------------------

# These settings can be modified to fit the needs of your application

# Default values for the Application variables, below.

#

# NOTE: The build for specific applications may override this during the resource-copying

# phase, to fill in a concrete name and avoid the use of the defaults specified here.

**DEF\_APP\_NAME="SonarQube"**

**DEF\_APP\_LONG\_NAME="SonarQube"**

# Application

**APP\_NAME="${DEF\_APP\_NAME}"**

**APP\_LONG\_NAME="${DEF\_APP\_LONG\_NAME}"**

# ADD THESE TWO LINES HERE:

**SONAR\_HOME=/opt/sonar**

**PLATFORM=linux-x86-64**

# Wrapper

#WRAPPER\_CMD="./wrapper"

#WRAPPER\_CONF="../../conf/wrapper.conf"

#SHUTDOWNER\_LIB\_DIR="../../lib"

# Wrapper (ENSURE these are uncommented and use the variables above)

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

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

**SHUTDOWNER\_LIB\_DIR="${SONAR\_HOME}/lib" # Ensure this uses SONAR\_HOME too**

# Location of the pid file (change to a standard system location)

**PIDDIR="/var/run/sonar"**

# Priority at which to run the wrapper. See "man nice" for valid priorities.

# nice is only used if a priority is specified.

**PRIORITY=**

# Location of the pid file.

#PIDDIR="."

# If uncommented, causes the Wrapper to be shutdown using an anchor file.

# When launched with the 'start' command, it will also ignore all INT and

# TERM signals.

#IGNORE\_SIGNALS=true

# If specified, the Wrapper will be run as the specified user.

# IMPORTANT - Make sure that the user has the required privileges to write

# the PID file and wrapper.log files. Failure to be able to write the log

# file will cause the Wrapper to exit without any way to write out an error

# message.

# NOTE - This will set the user which is used to run the Wrapper as well as

# the JVM and is not useful in situations where a privileged resource or

# port needs to be allocated prior to the user being changed.

**RUN\_AS\_USER=sabear**

# The following two lines are used by the chkconfig command. Change as is

# appropriate for your application. They should remain commented.

# chkconfig: 2345 20 80

# description: Test Wrapper Sample Application

# Do not modify anything beyond this point

#-----------------------------------------------------------------------------

**change permmisions:**

sudo chmod 755 /etc/init.d/sonar

**Create PID Directory and Set Permissions:**

sudo mkdir -p /var/run/sonar

sudo chown sabear:sabear /var/run/sonar

sudo systemctl daemon-reload

sudo sysctl -w vm.max\_map\_count=262144

**go to vi conf:**

sudo vi /etc/sysctl.conf

Add the following line to the end of the file)

vm.max\_map\_count=262144

**Troubleshooting & Resolution Steps :**

sudo service sonar stop # Ensure SonarQube is stopped first

sudo rm -rf /opt/sonar/data/es6

sudo service sonar restart

**Monitor Logs for Startup Progress and Errors:**

sudo tail -f /opt/sonar/logs/sonar.log

sudo tail -f /opt/sonar/logs/es.log

sudo tail -f /opt/sonar/logs/web.log

**Check Running Processes:**

ps aux | grep -i sonar

ps aux | grep -i elasticsearch

**in the master jenkins:**

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

**Build:**

**- `Invoke top-level Maven targets`**

**`Goals : clean install`**

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

**```sh**

* **Create a job**
* **attach the git repo https url**
* **go to build steps>>choose executive sonar scanner>>in that paste the bellow code in analysis properties**

**till now integration steps done**

* in the sonar server>>go to administrator>>go to my account>>go security>>generate a token

Go to the manage jenkins>>go to credentials>>create a credentials>>with the generated sonar token

in manage jenkins>>select systems>>in that go to SonarQube servers >>give the sonar server ip url>>attach the create credential before>>click on save

Go to manage jenkins>>select tools>>go to option SonarQube Scanner installations>>in that give the sonar scanner path that we have done in jenkins master last step of integration steps >>save it

for notification install slack notification plugins

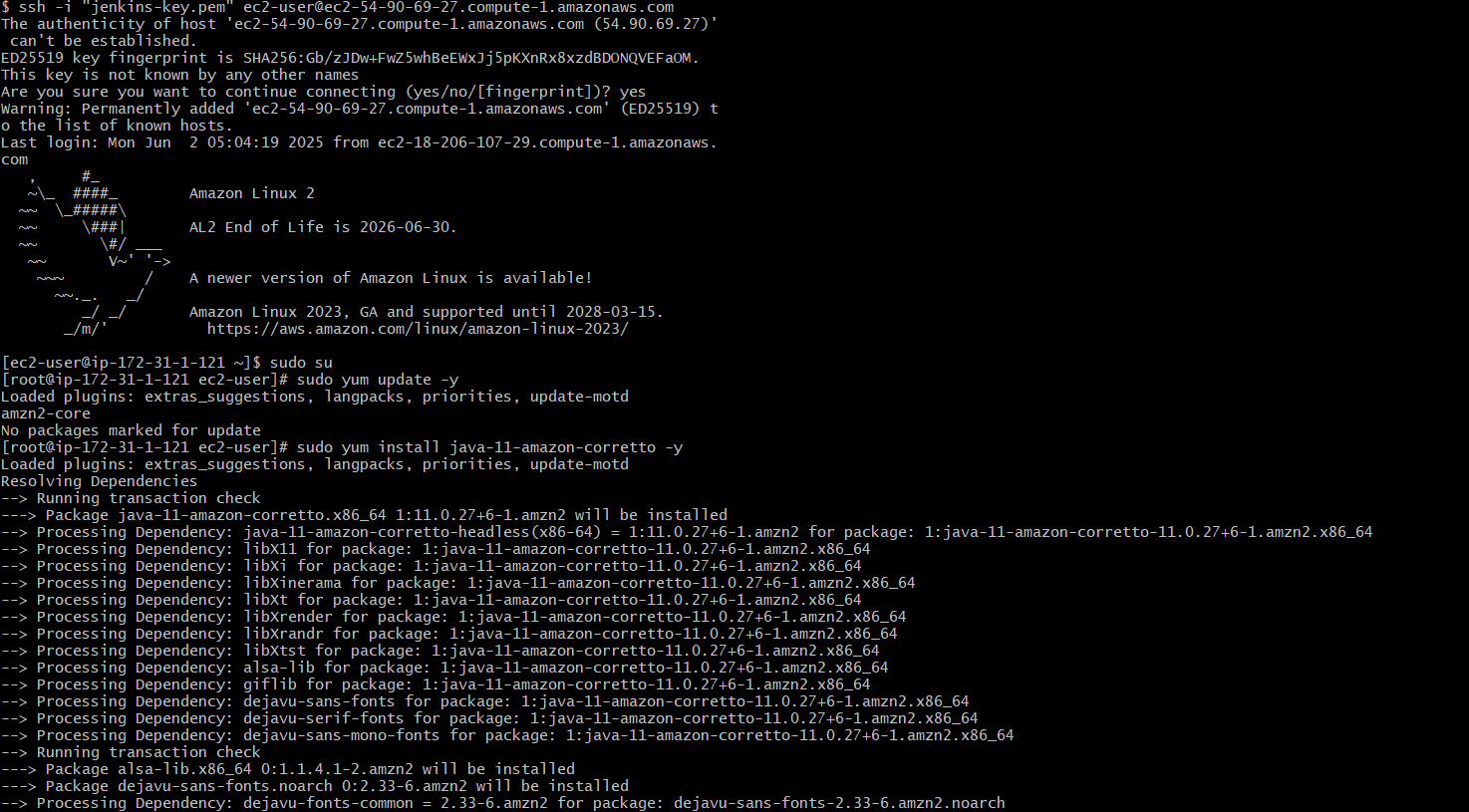
* for creation slack webhooks go to

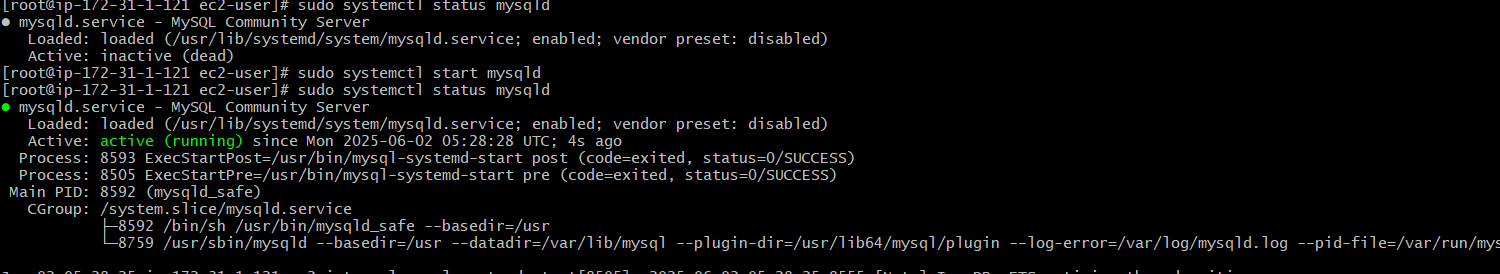
<https://api.slack.com/apps/A08UQP2Q0CV/incoming-webhooks>

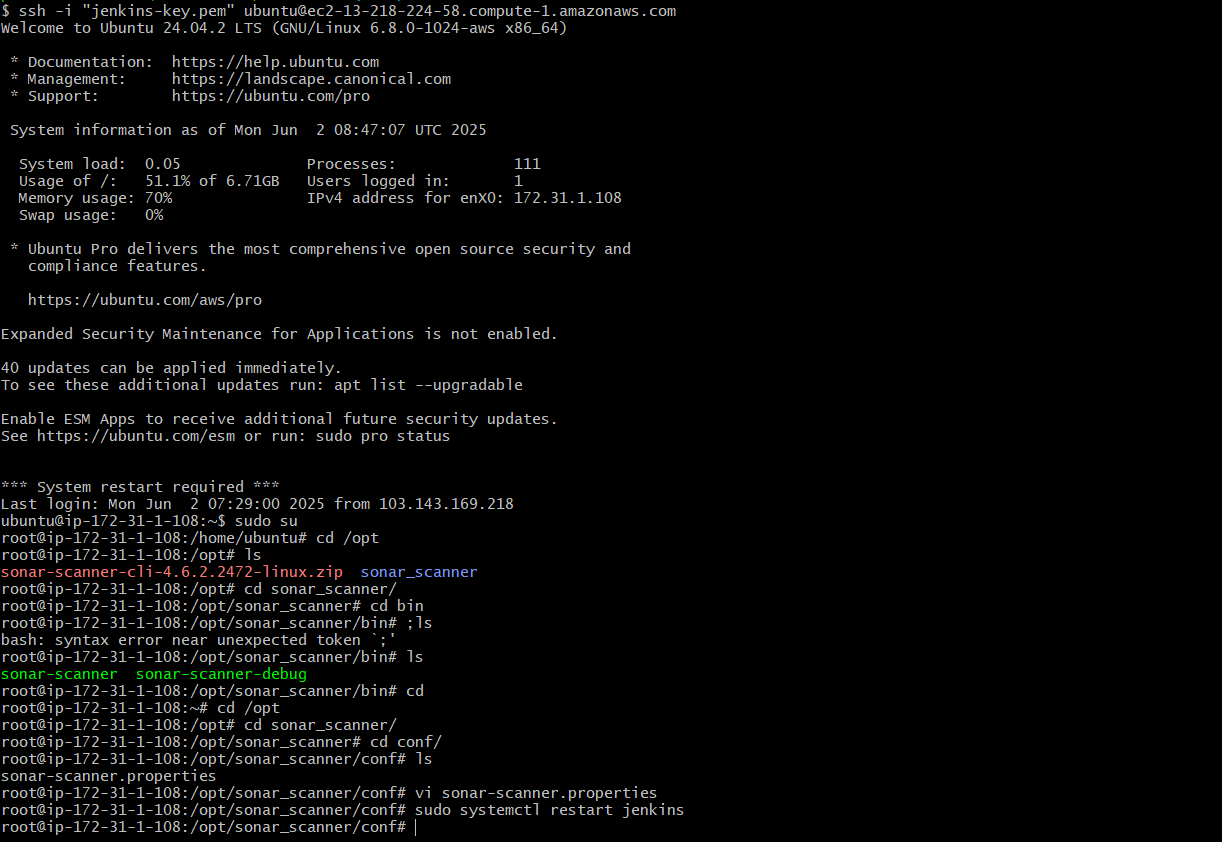
create the channel and go to that channel and in the settings you will see the incomming webhooks >>generate the webhooks

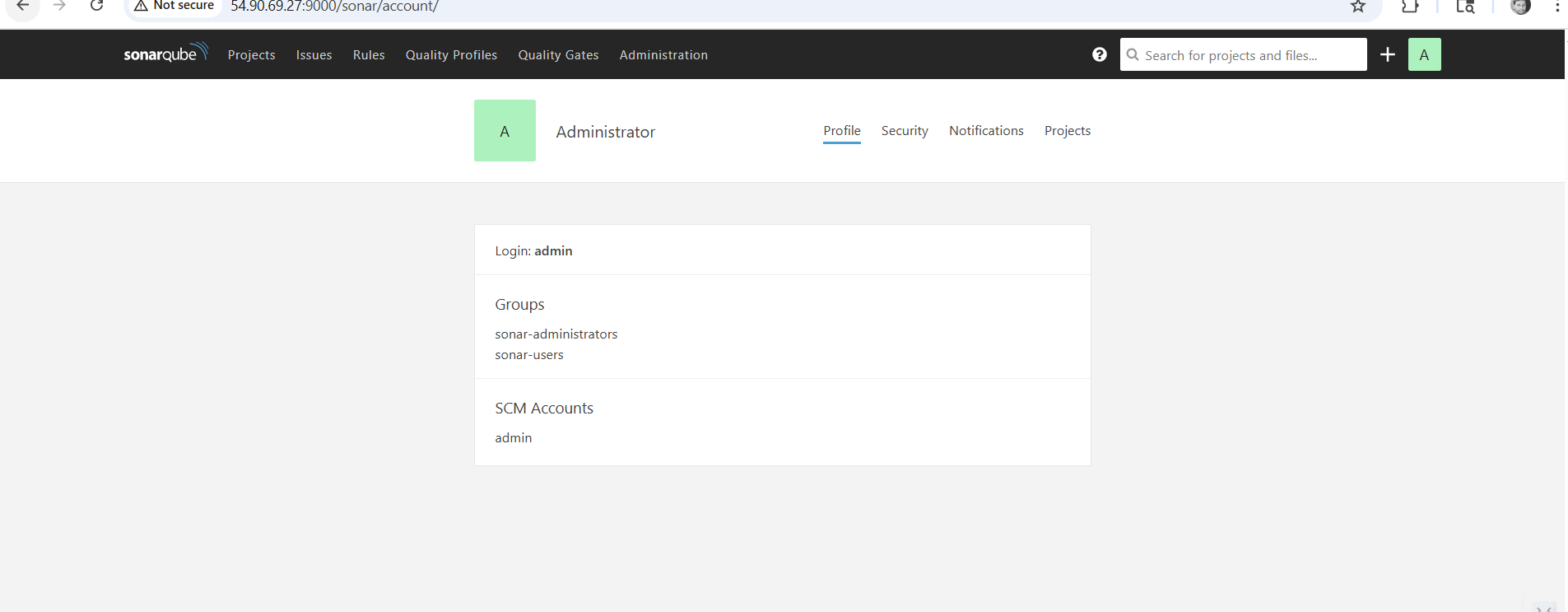
* go to the manage jenkins >>slect system>>in the bottom you will see slack >>give the credentials what we have created

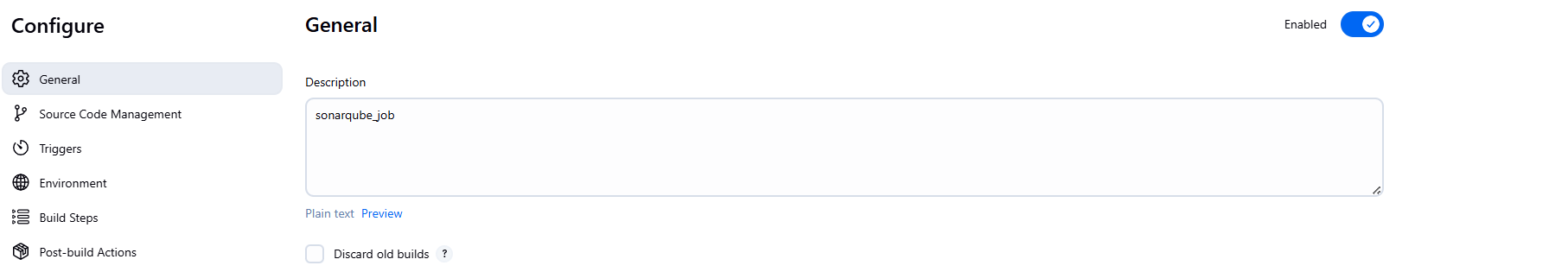
 in the jobs configure >>go to bottom >>slect post build ACTIONS>>slect slack notification>>then selct all>>in th advance >>give the slack credentials >>save it

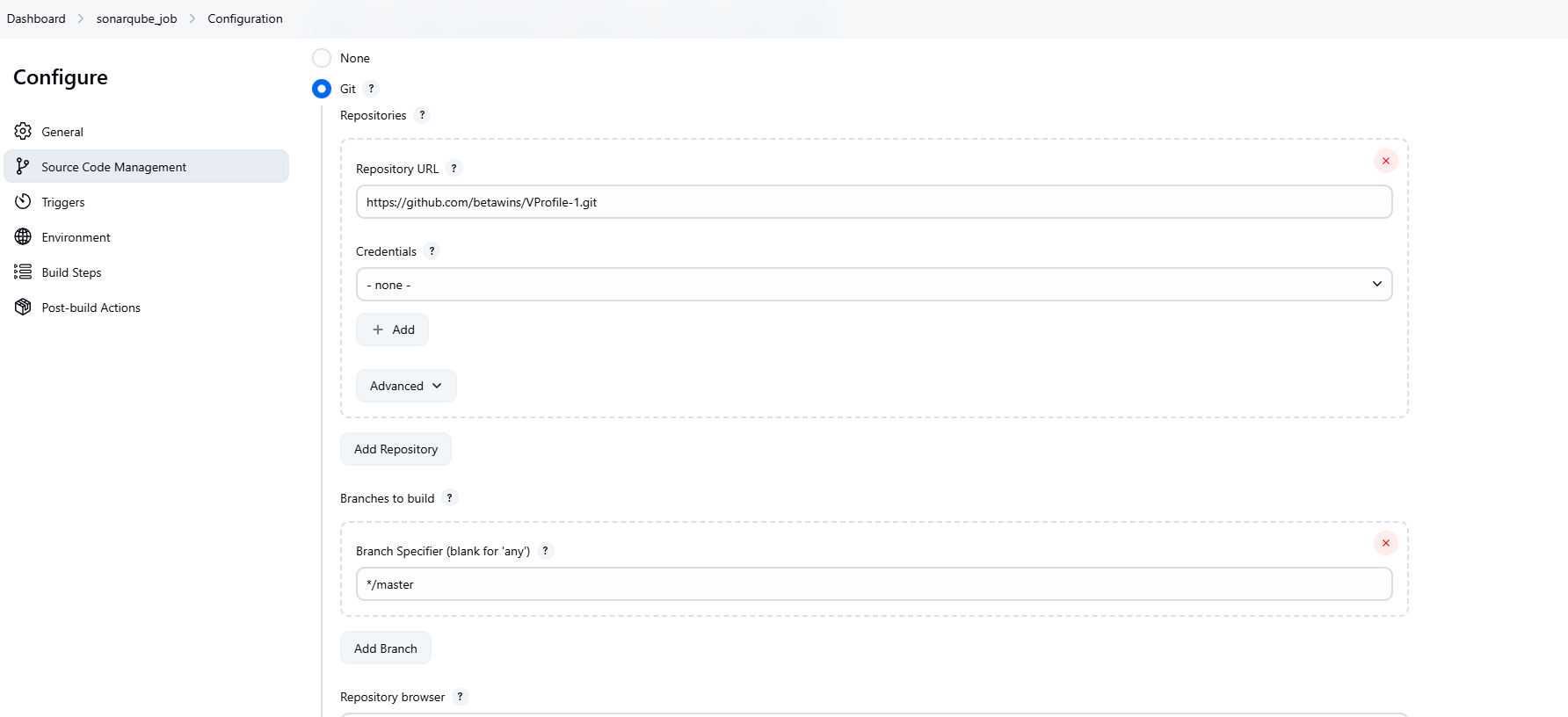


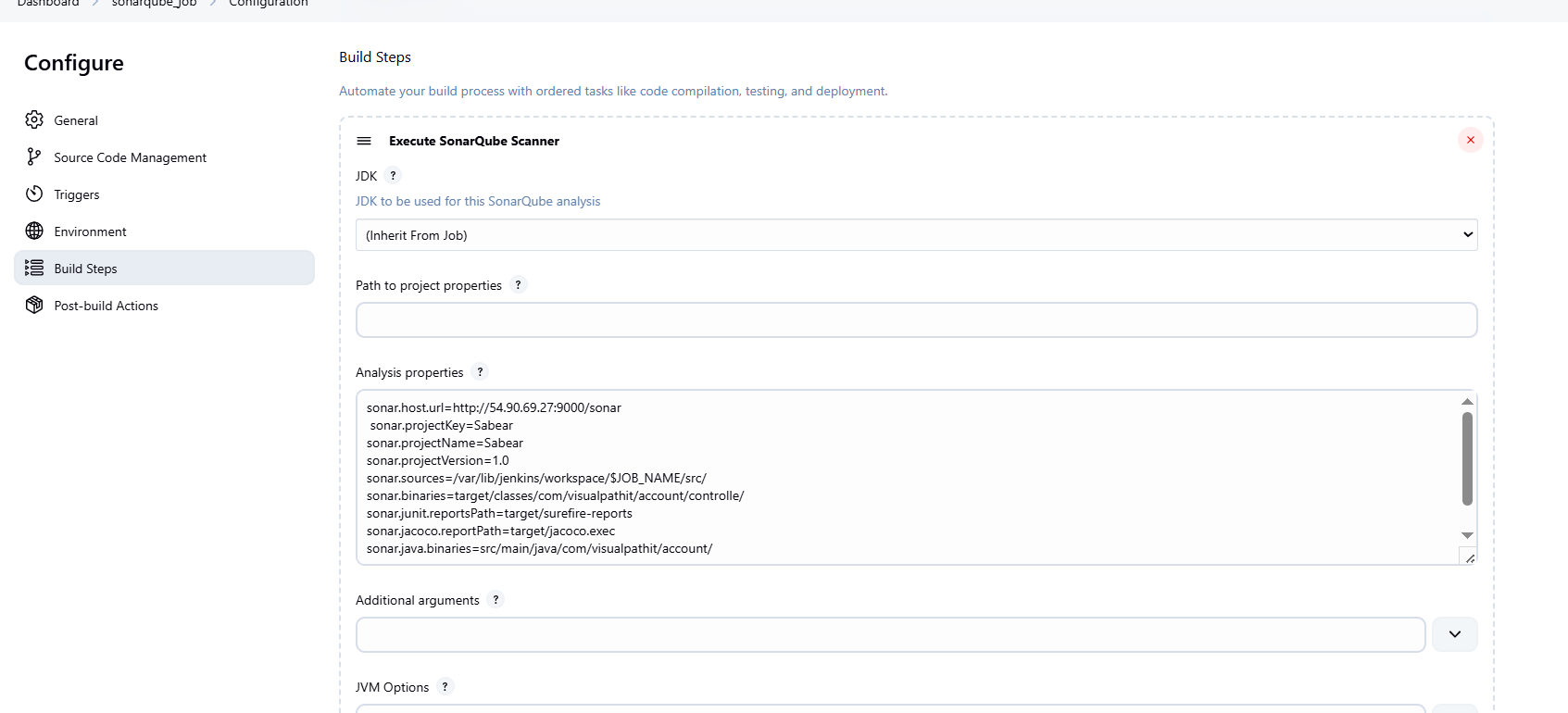


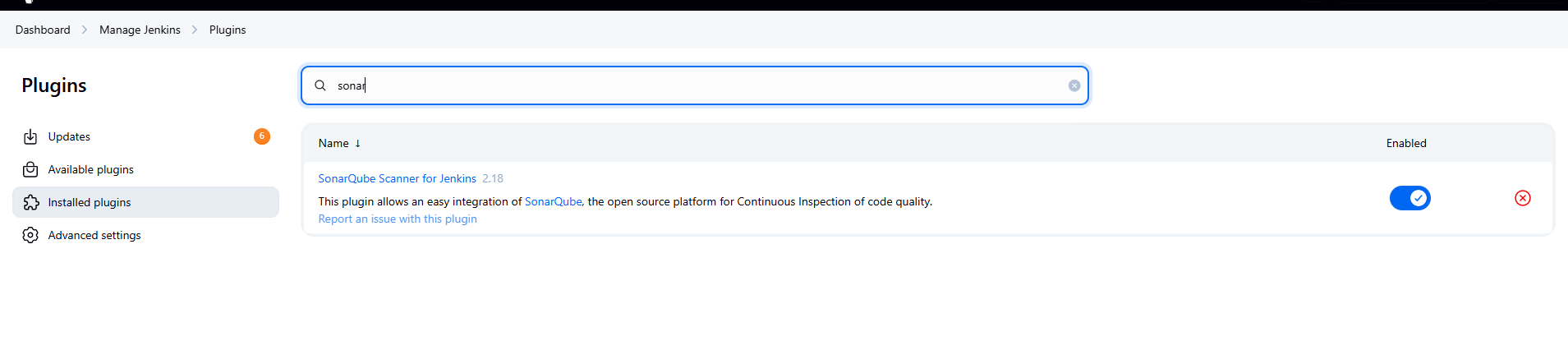


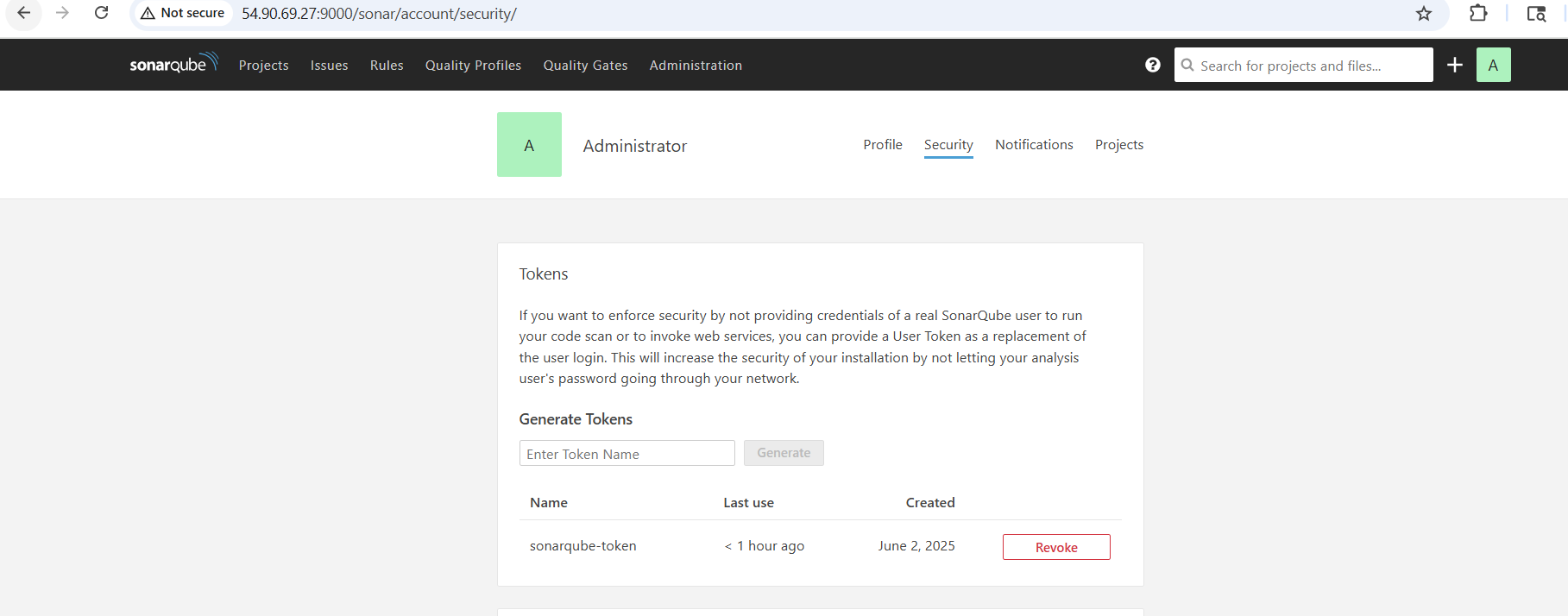


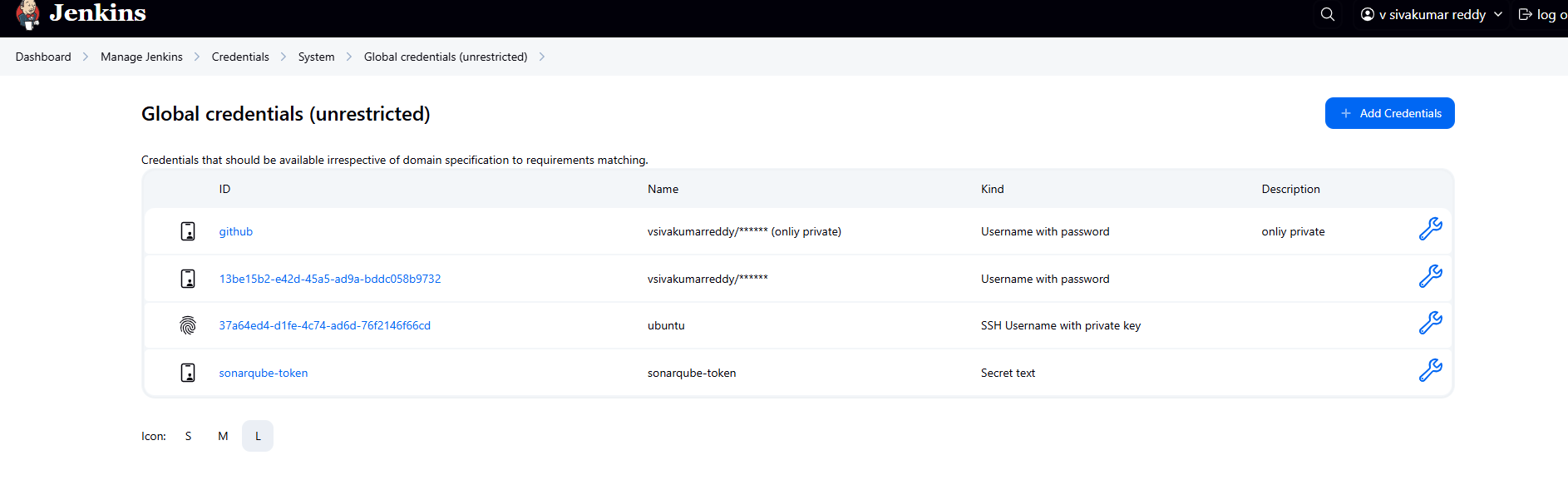


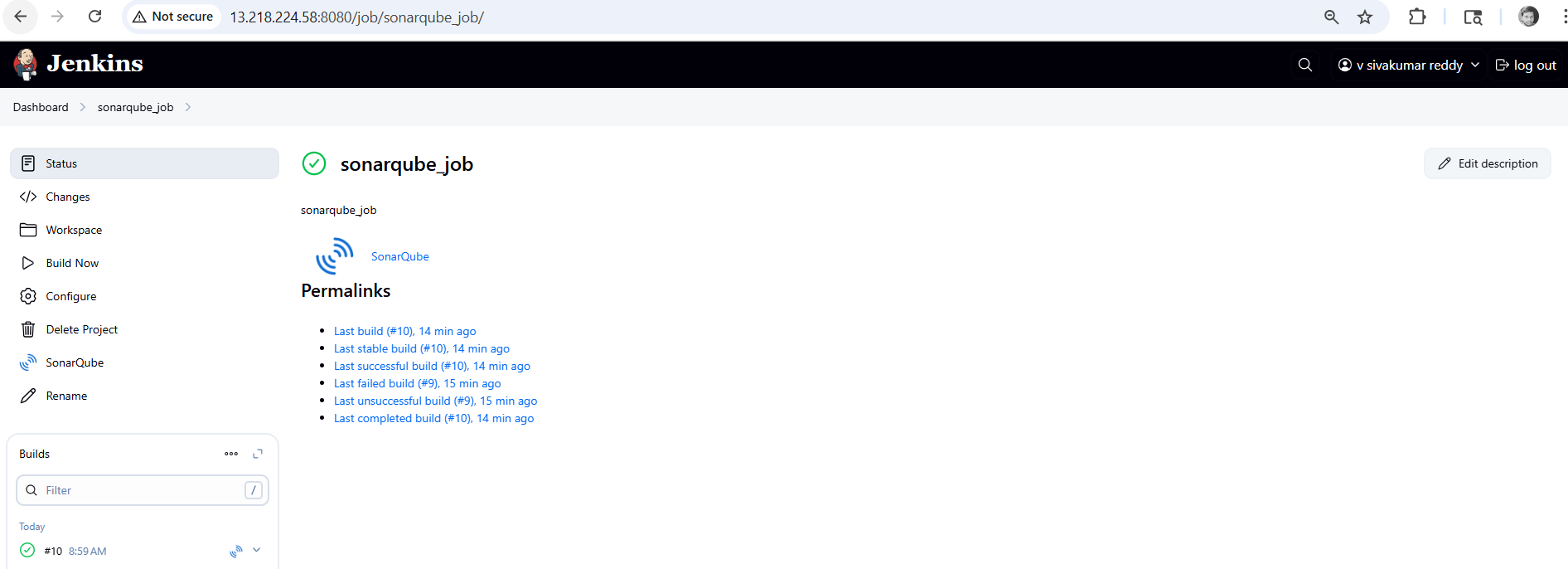


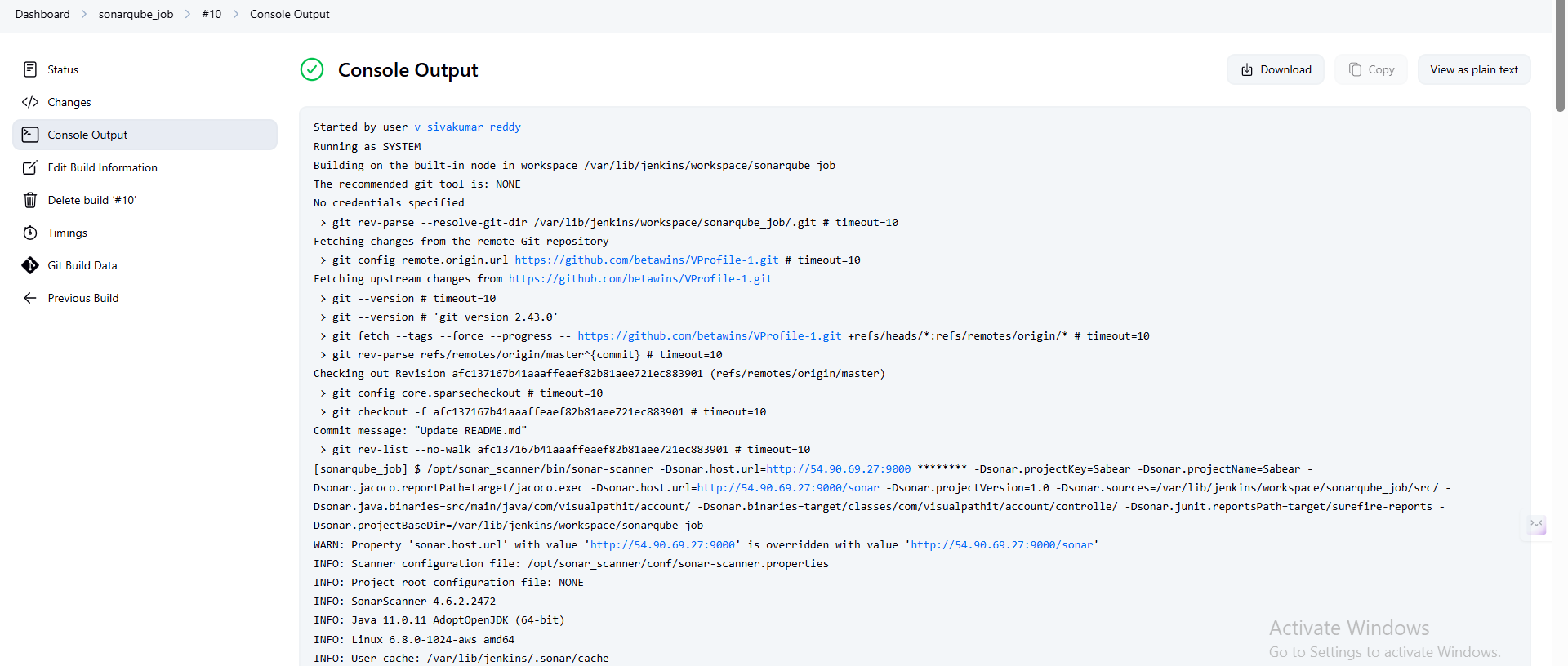


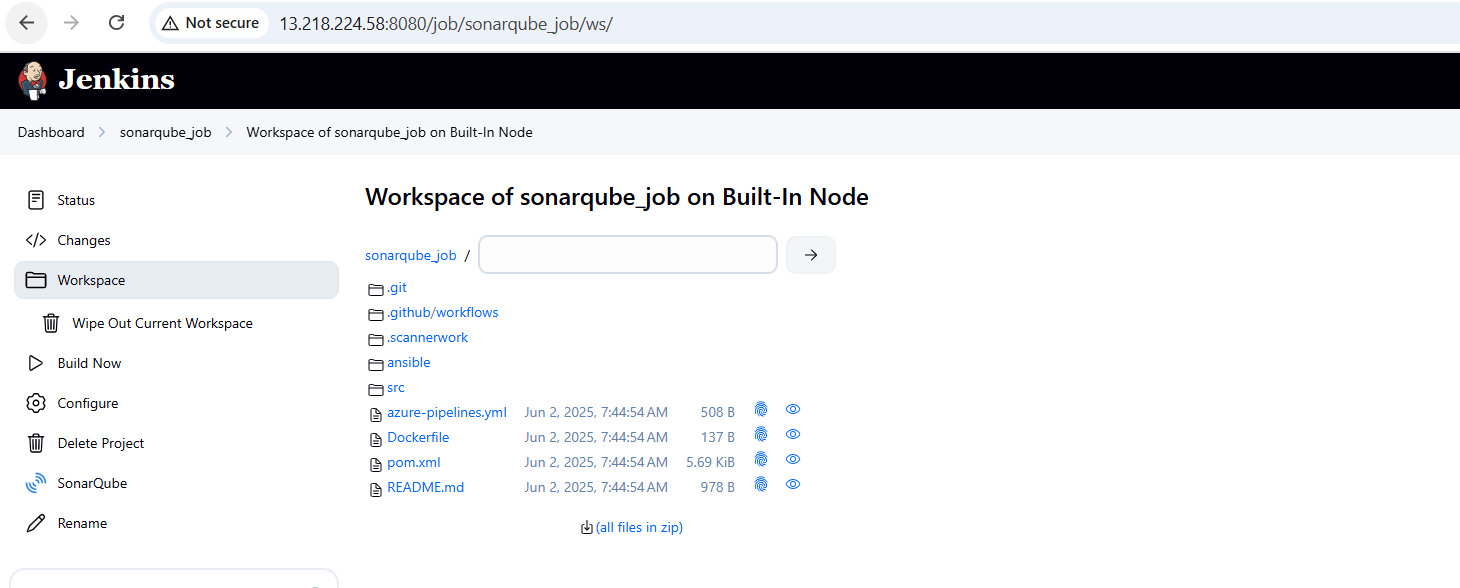


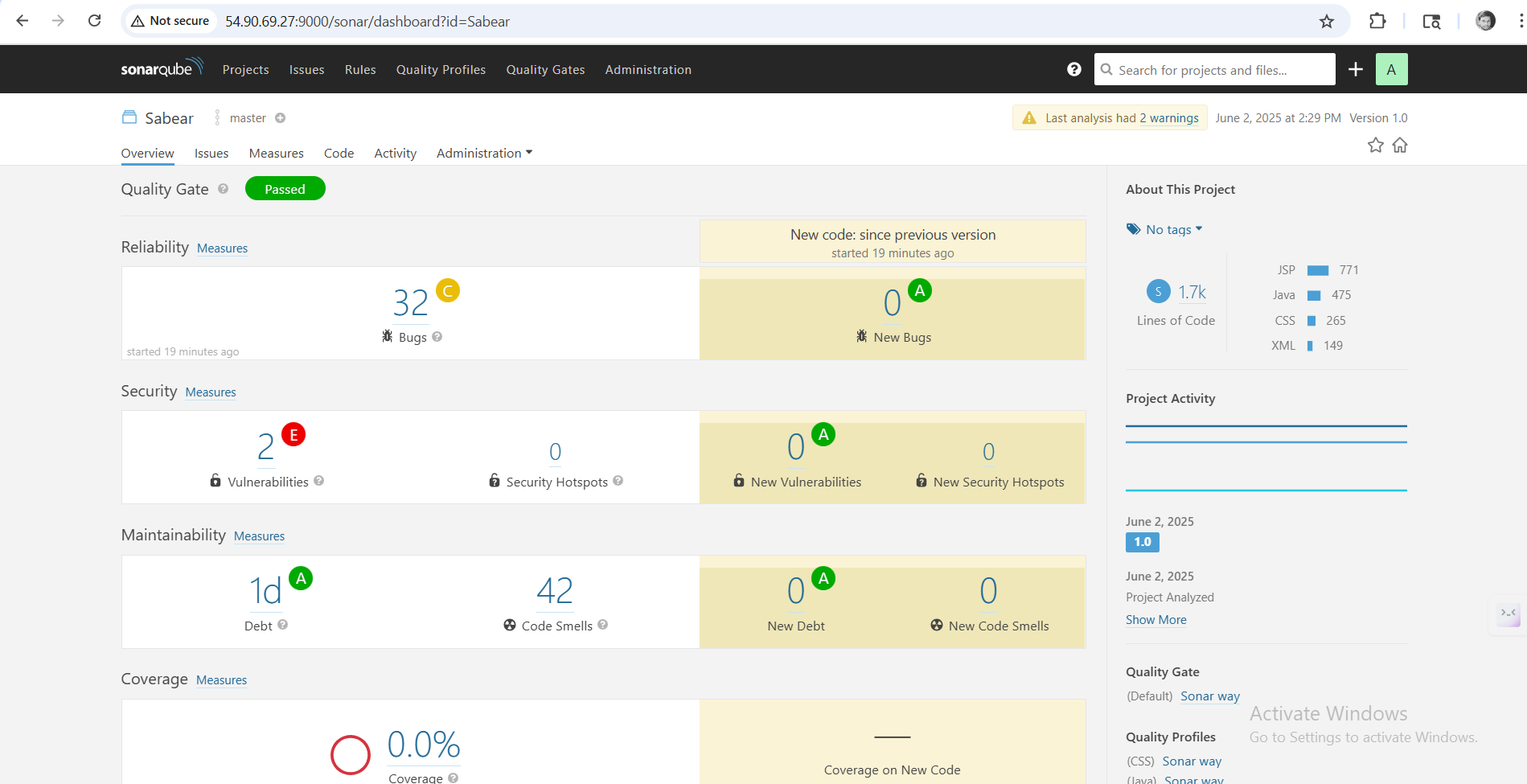


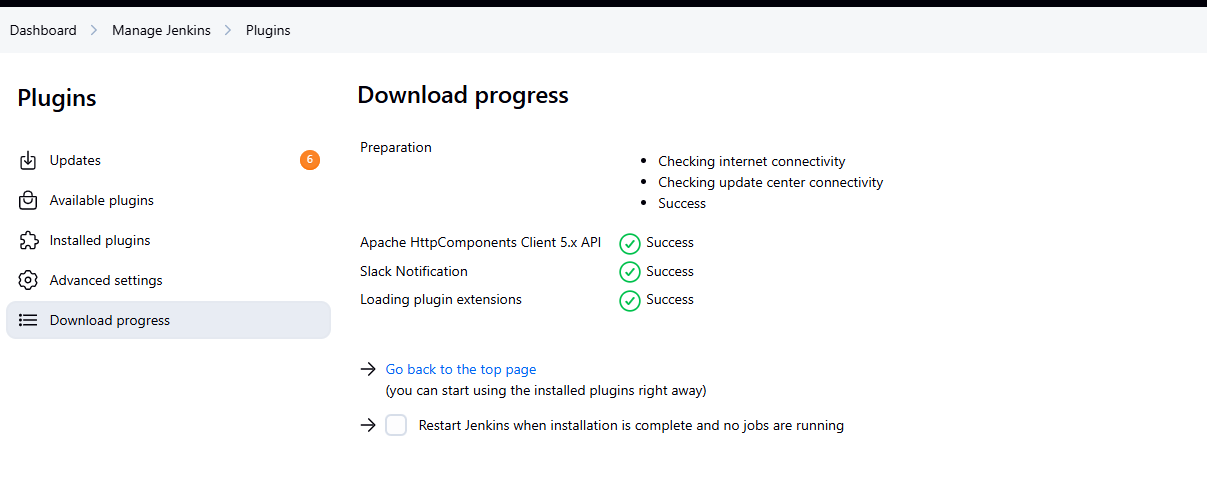


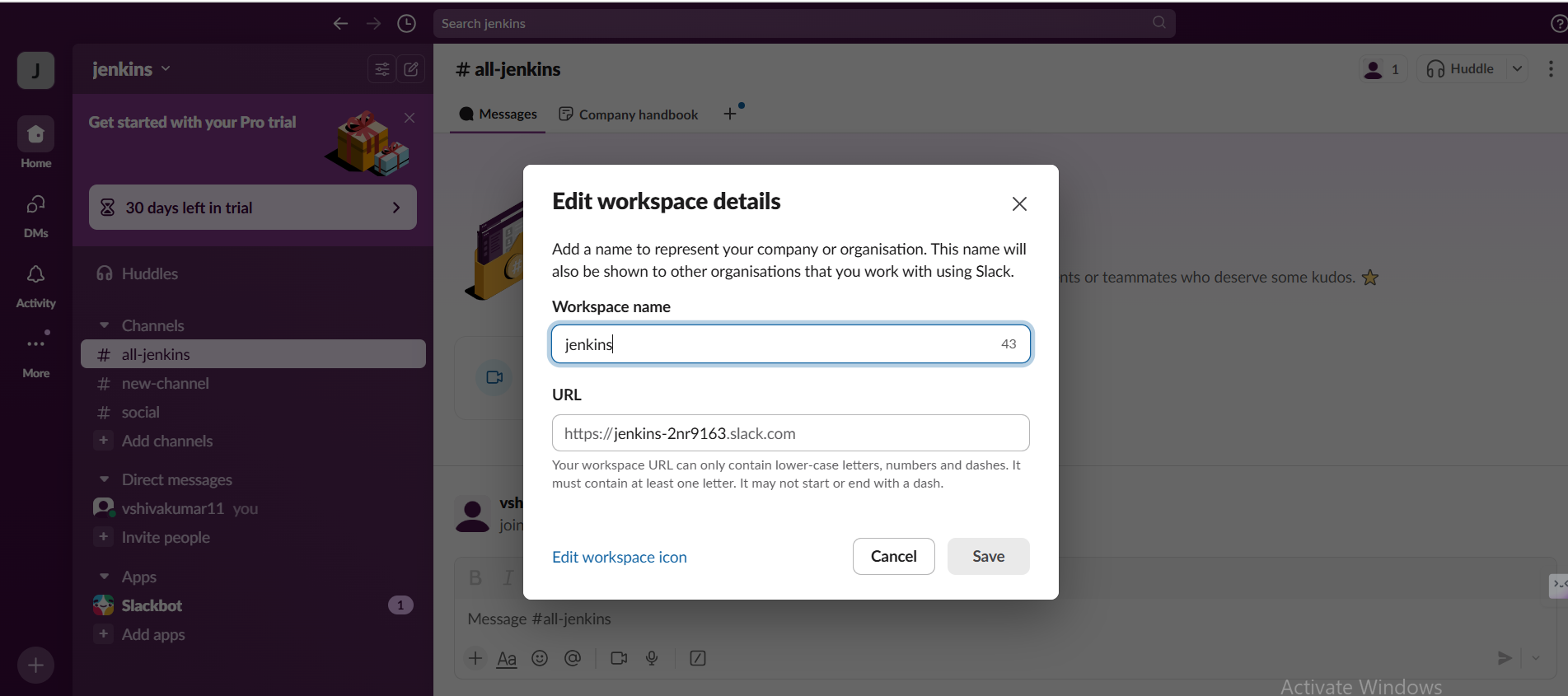


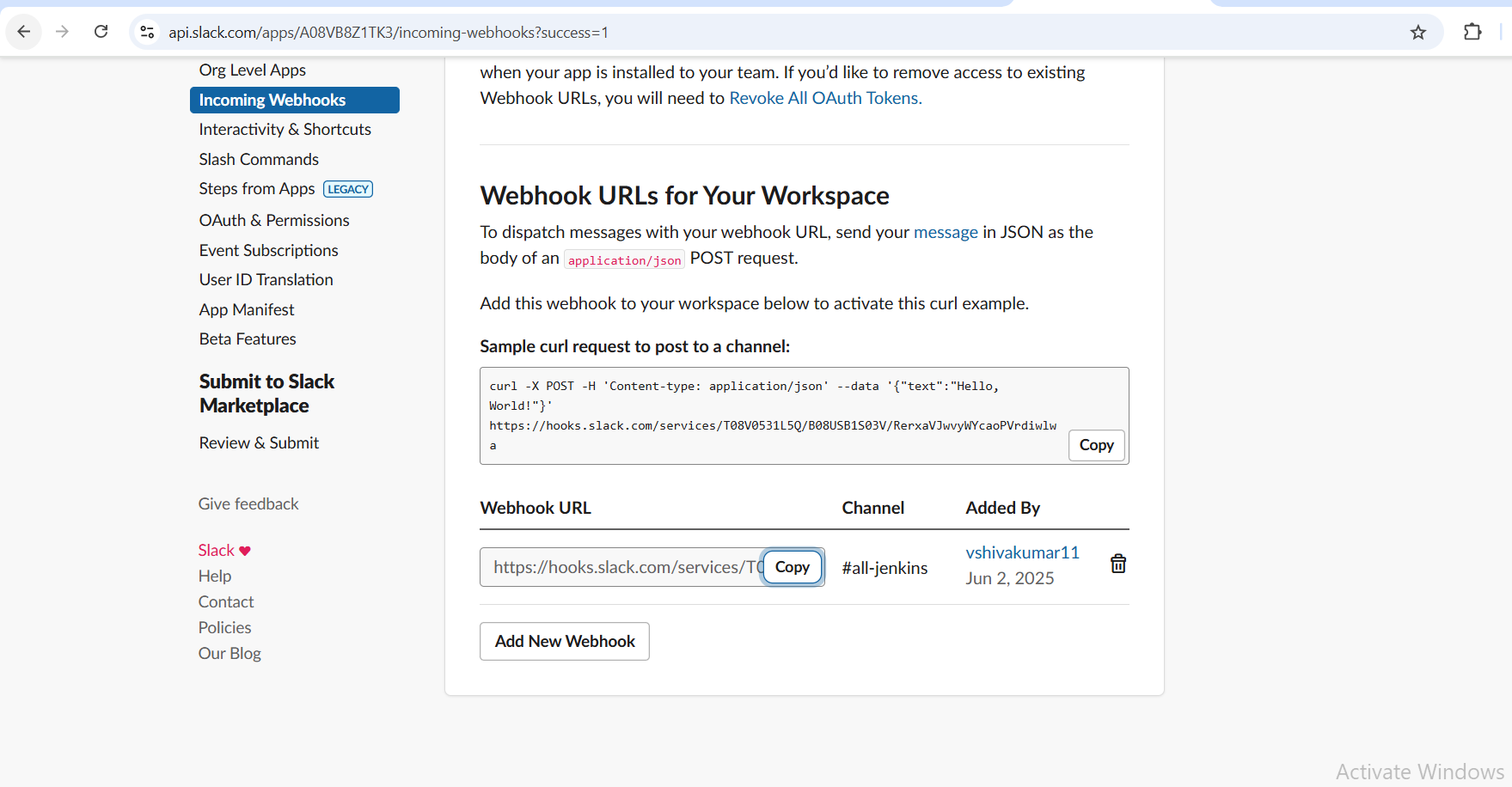


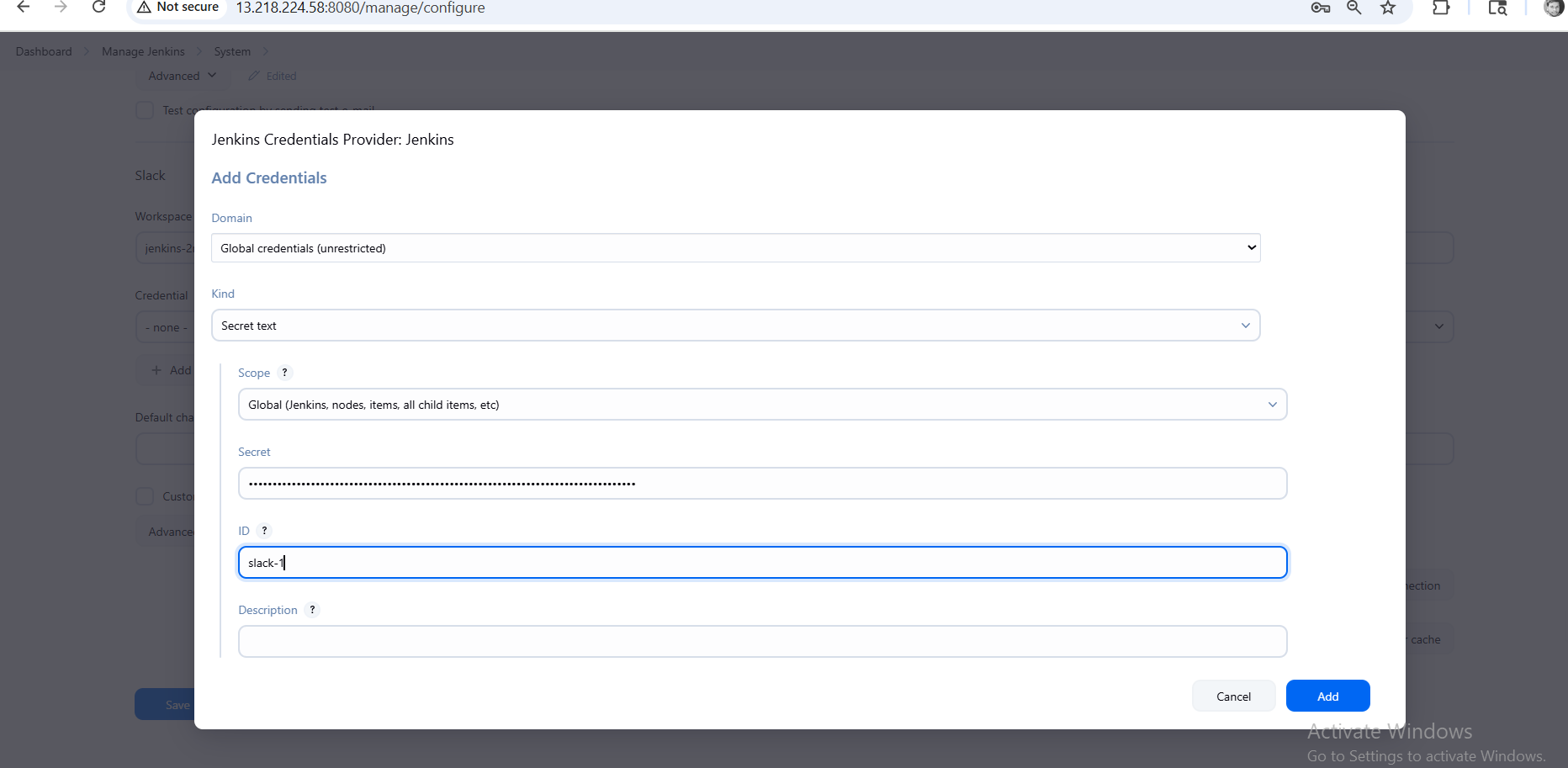


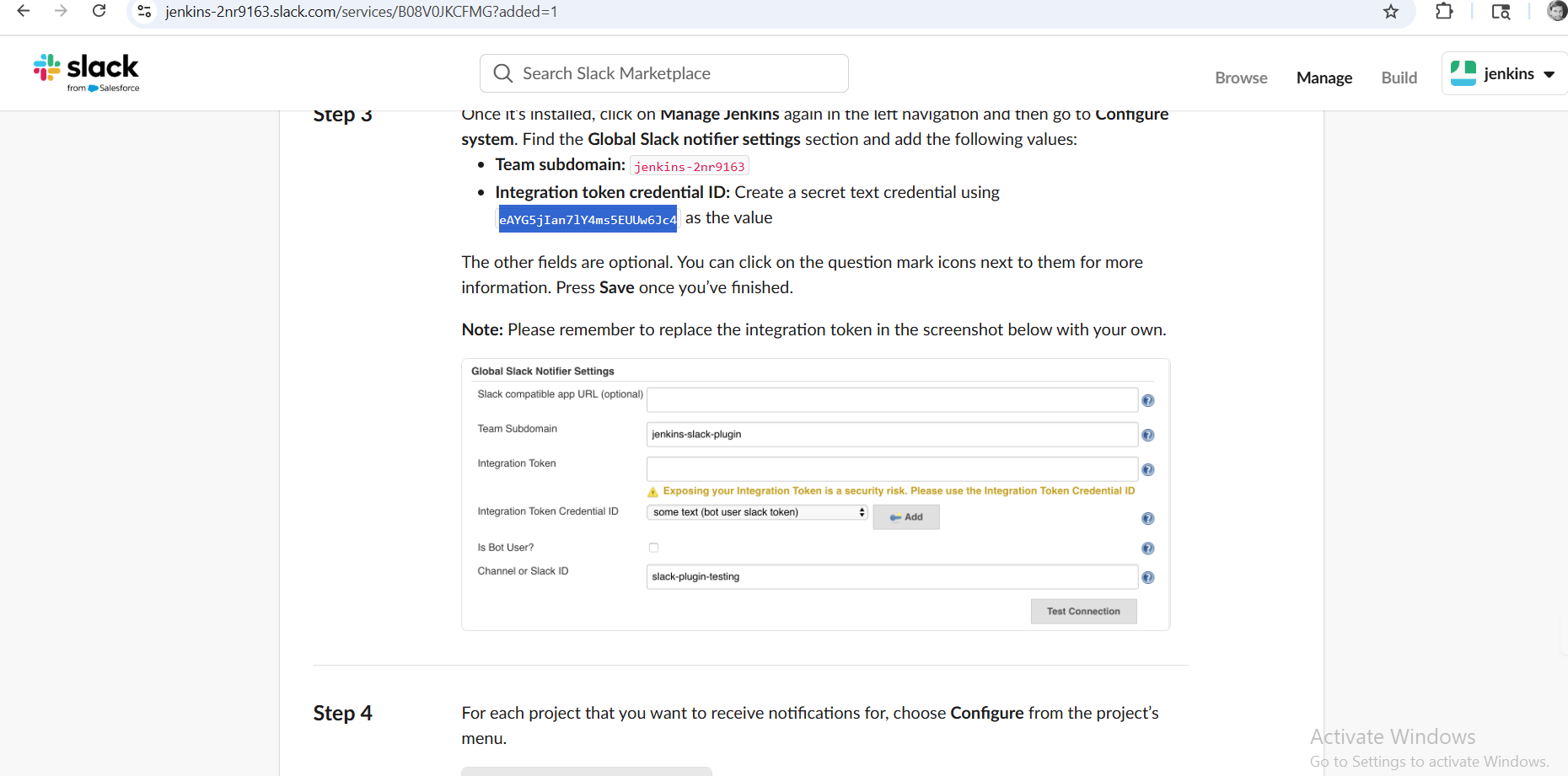


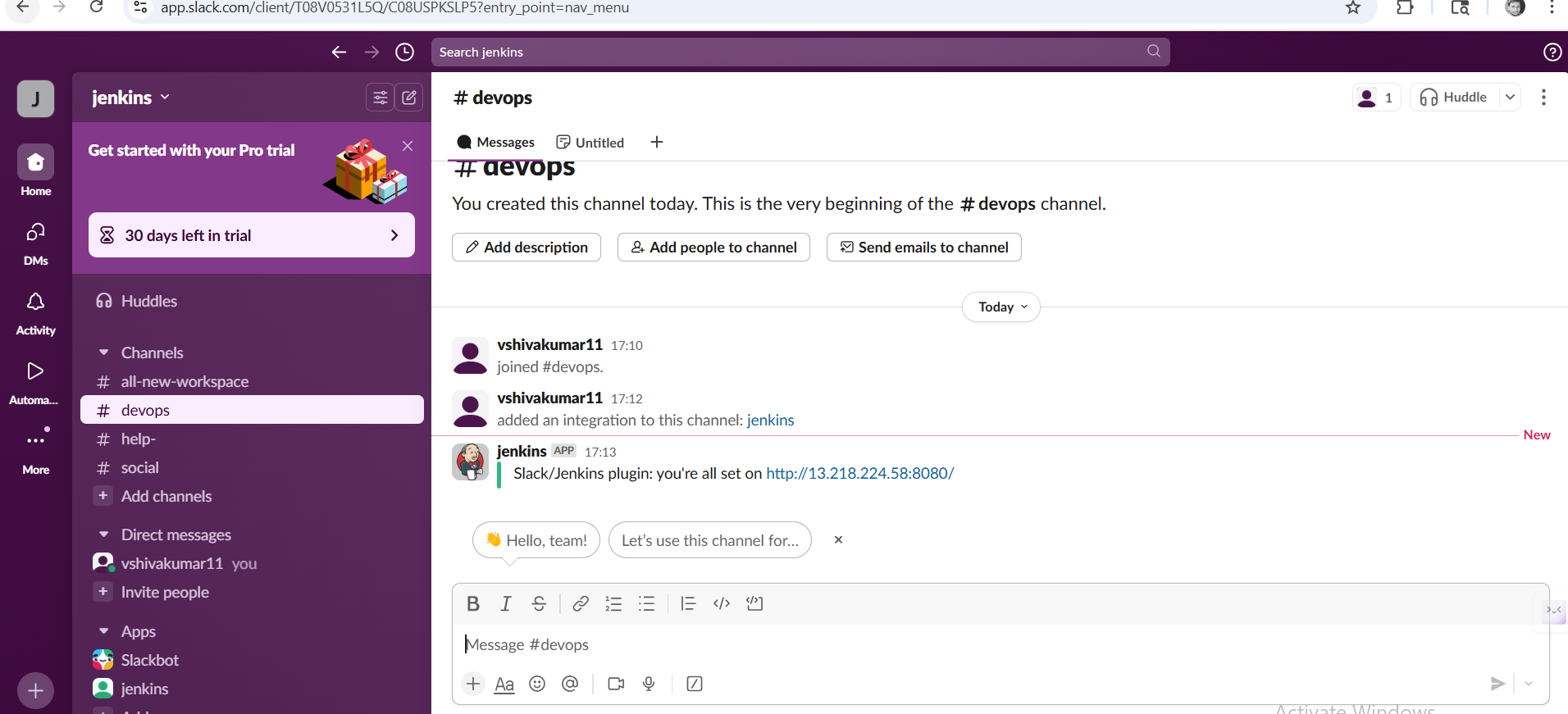












1. 1) Create one jenkins job using the below code and create three stages.

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>

STEPS:

For this question also repeat the first question steps

