JENKINS-TASK03

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

stage1: Git clone to download the source code.

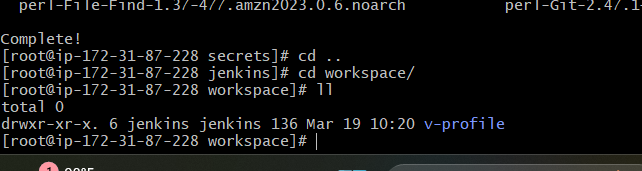
stage2: Sonarqube Integration to check the quality of code

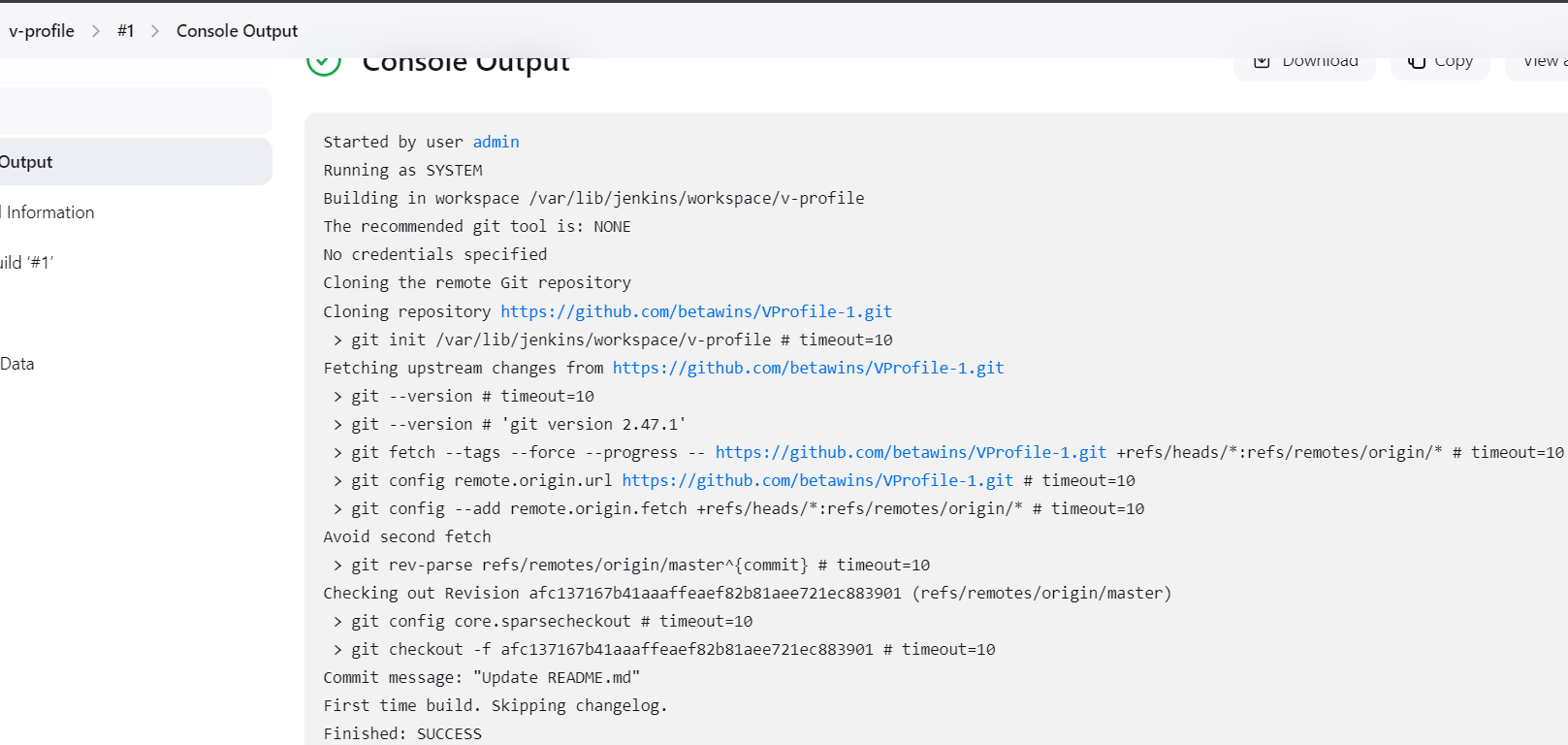
stage3: Slack Integration to send the alerts to slack.

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

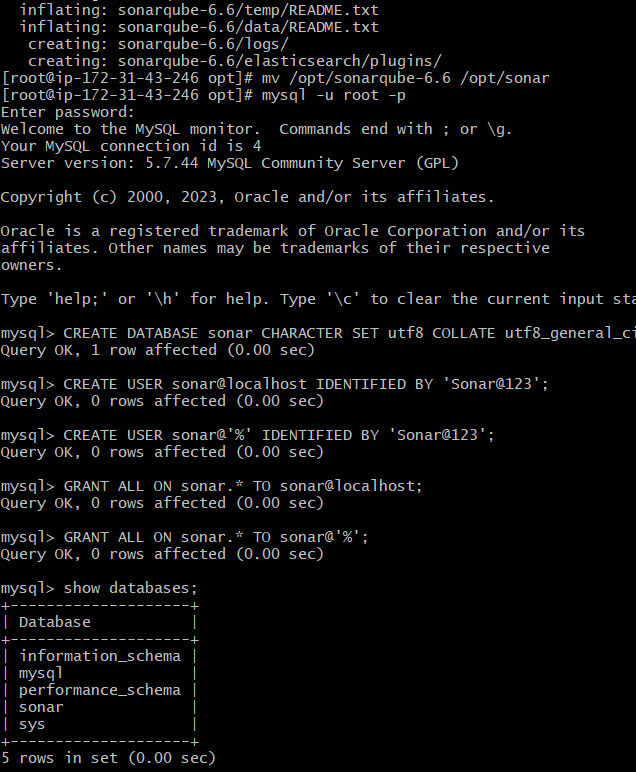
stage1:

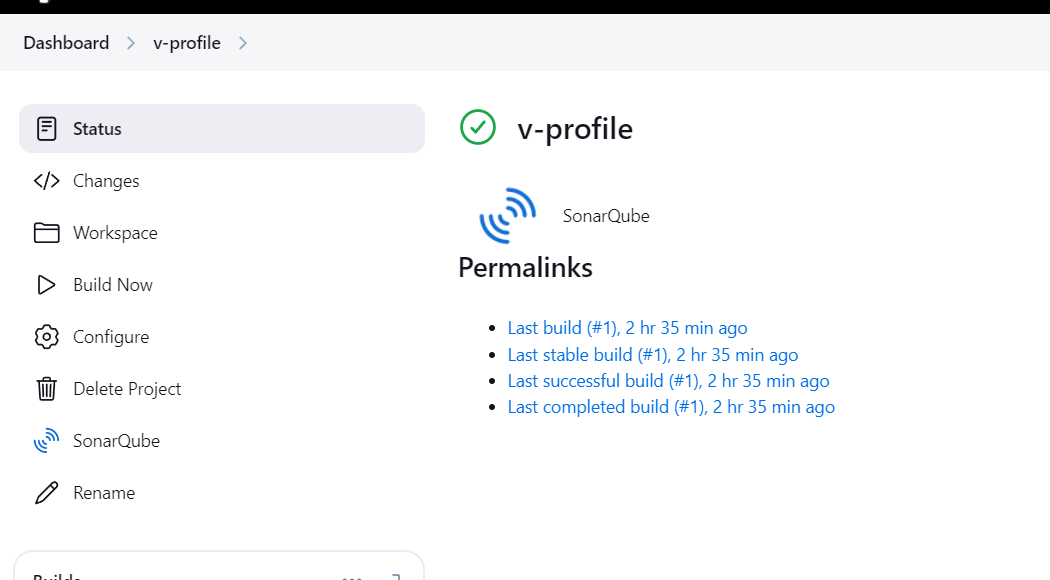
I have launched one server and installed jenkins in it and successfully clone the repo in my jenkins

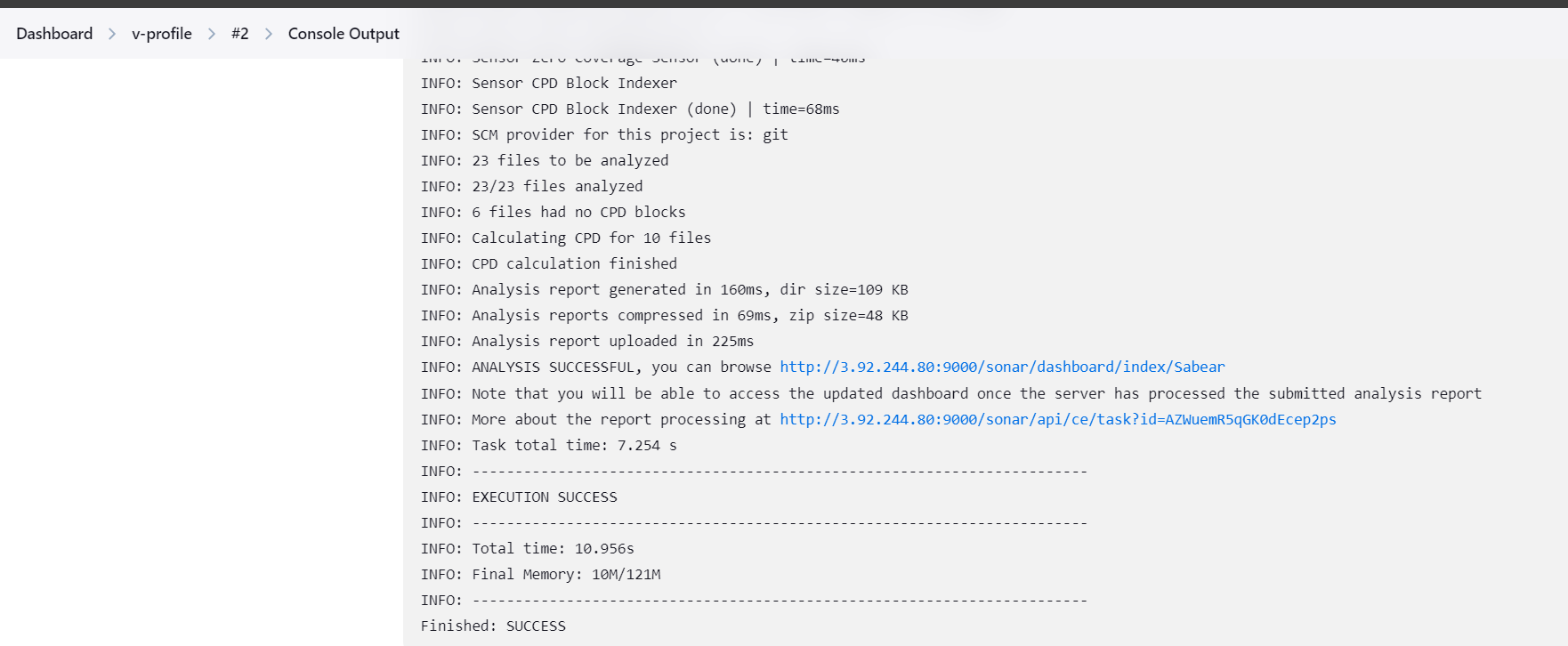


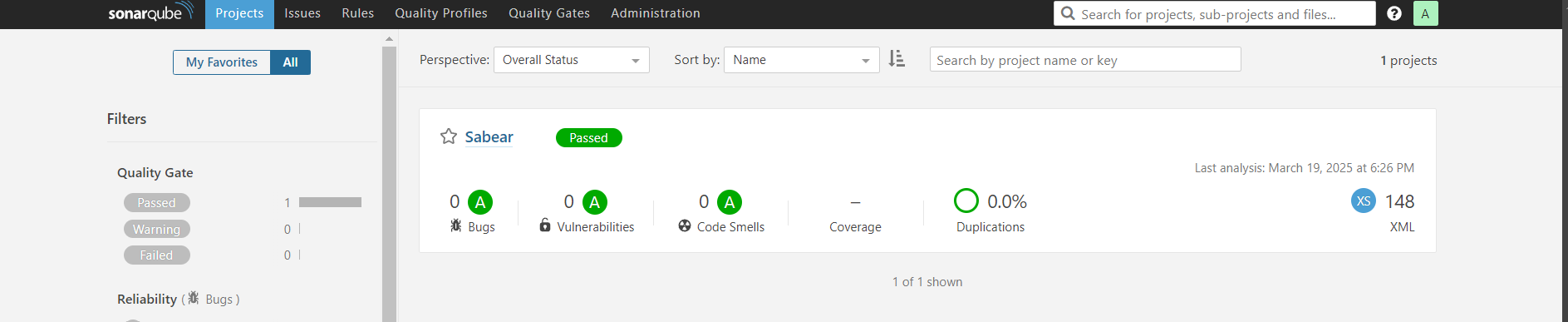


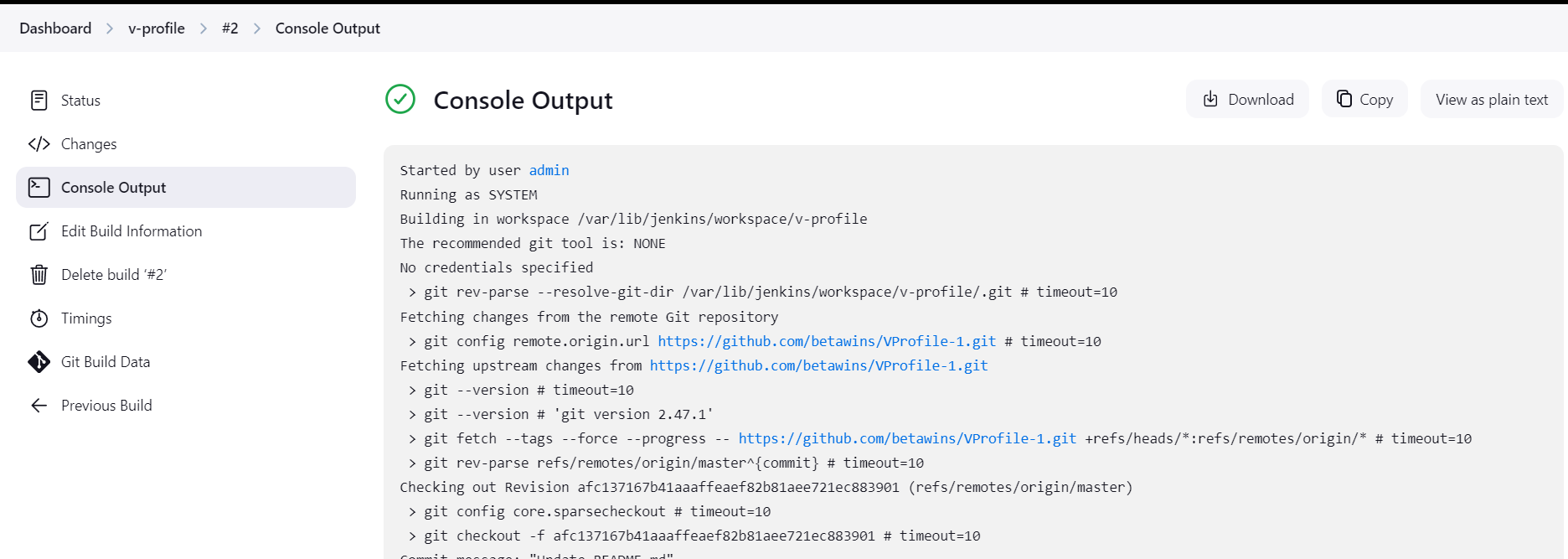
Step 2 : Sonarqube Integration

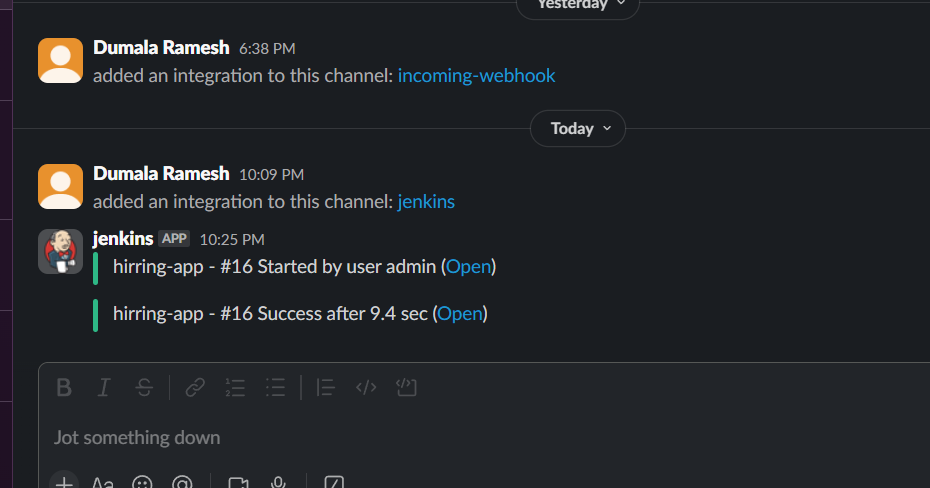












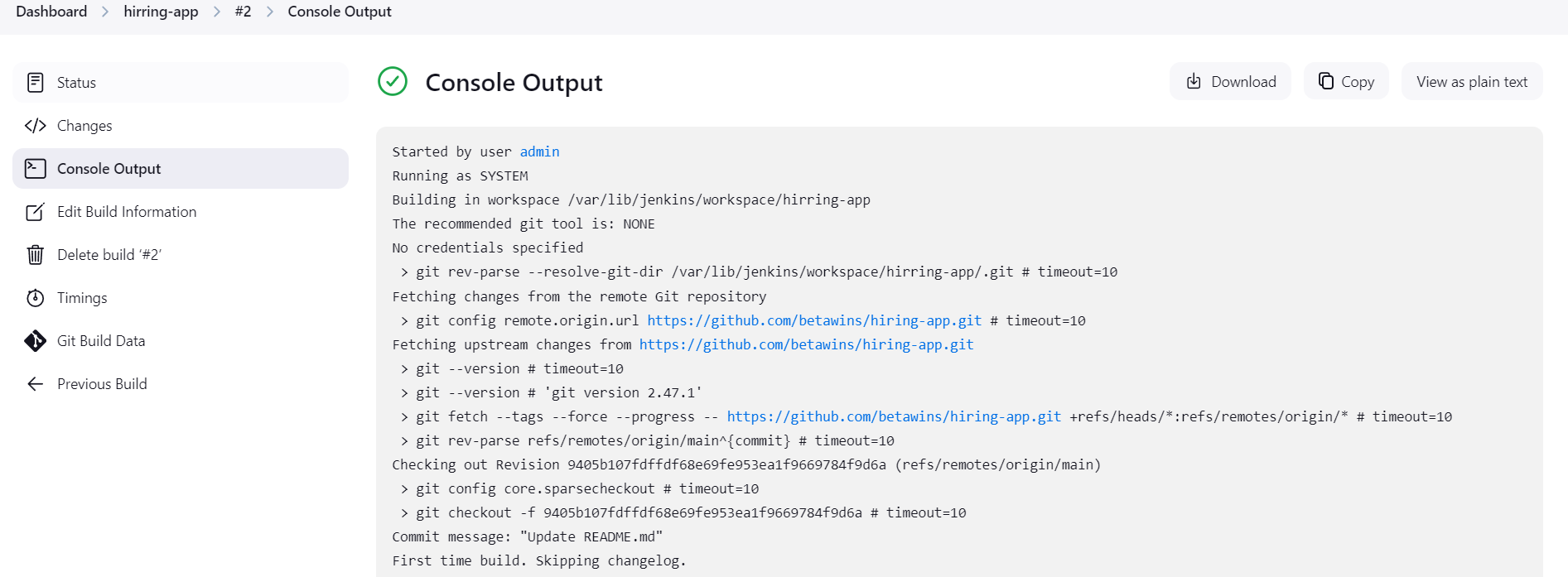
2) 1) Create one jenkins job using the below code and create three stages.

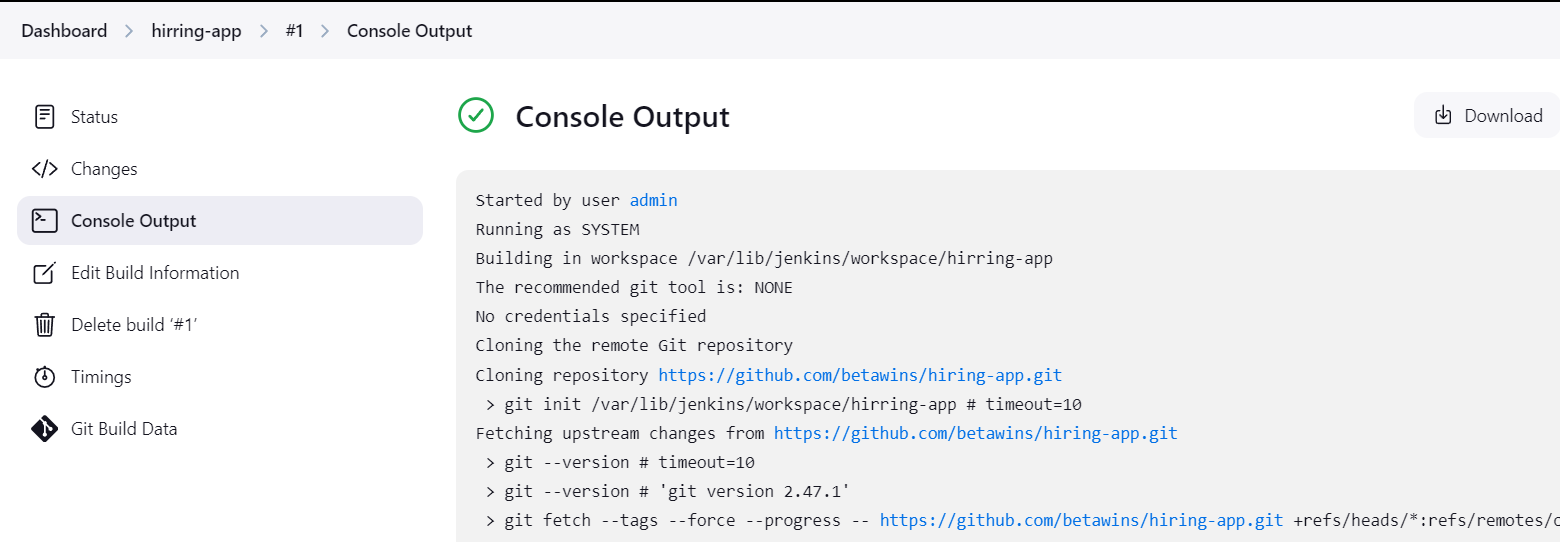
stage1: Git clone to download the source code.

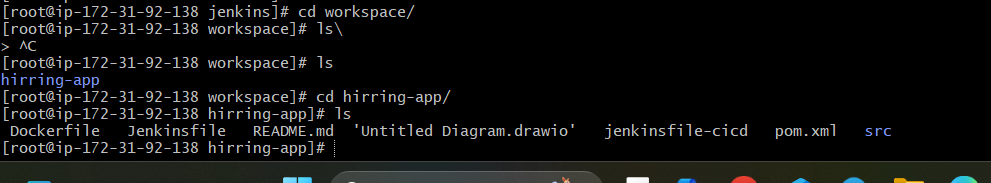
stage2: Sonarqube Integration to check the quality of code

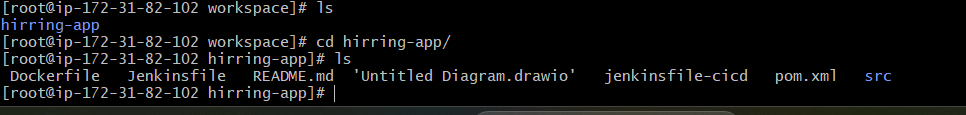
stage3: Slack Integration to send the alerts to slack.

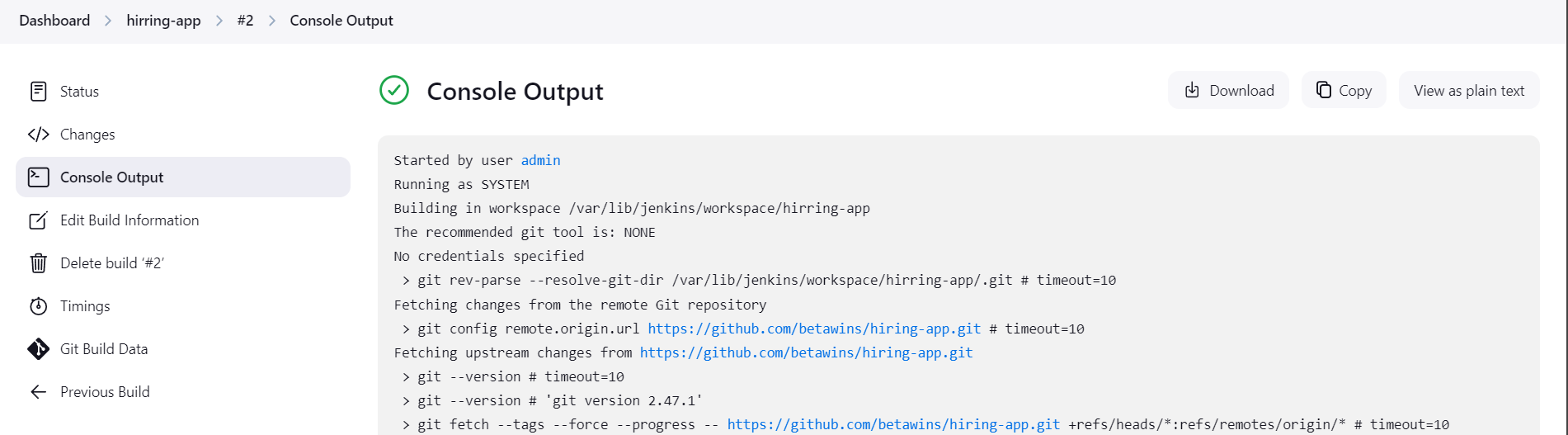
URL: <https://github.com/betawins/hiring-app.git>

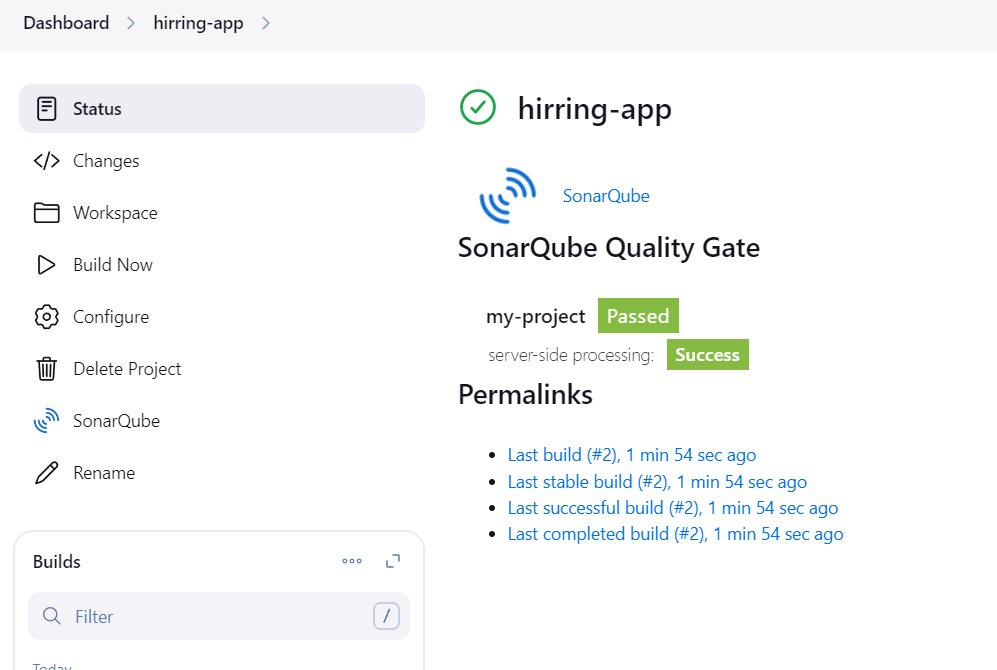


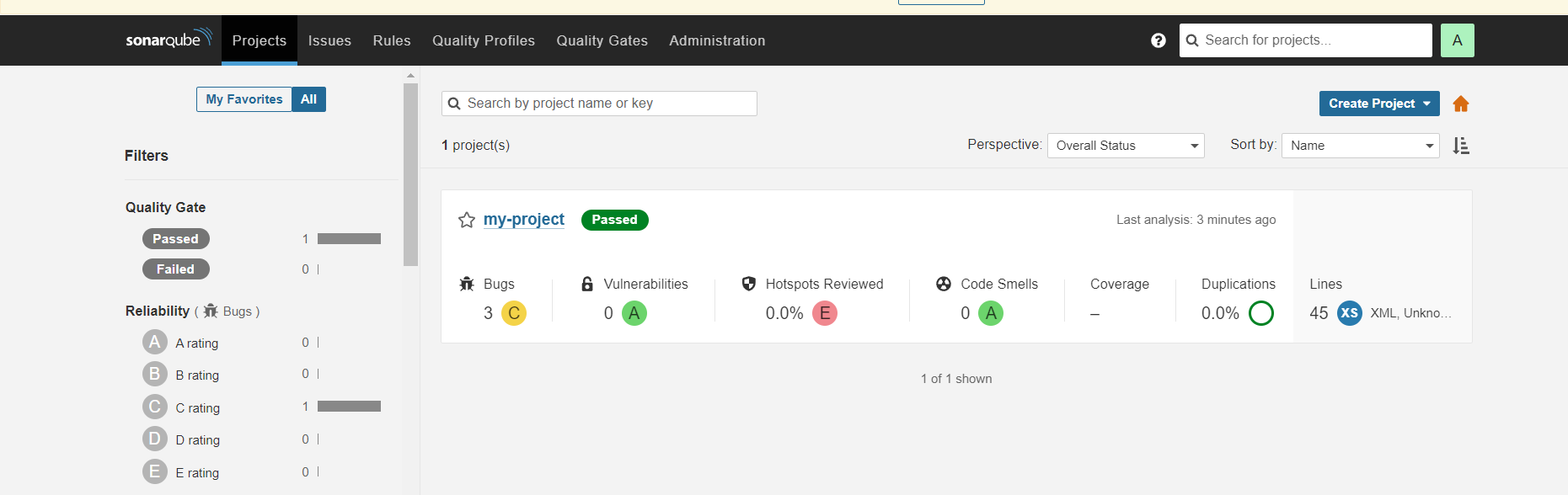


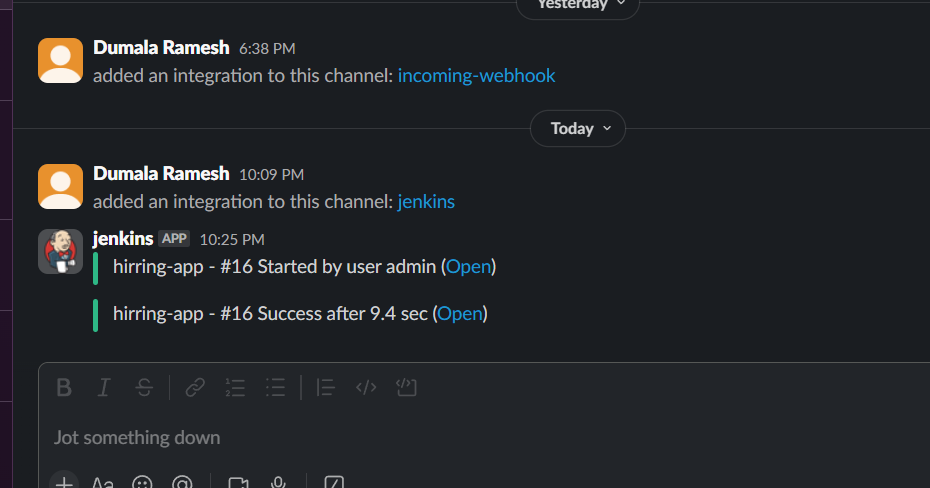












#!/usr/bin/env sh

#

# SonarScanner Startup Script for Unix

#

# Required ENV vars:

# JAVA\_HOME - Location of Java's installation, optional if use\_embedded\_jre is set

#

# Optional ENV vars:

# SONAR\_SCANNER\_OPTS - Parameters passed to the Java VM when running the SonarScanner

# SONAR\_SCANNER\_DEBUG\_OPTS - Extra parameters passed to the Java VM for debugging

real\_path () {

target=$1

(

while true; do

cd "`dirname "$target"`"

target=`basename "$target"`

test -L "$target" || break

target=`readlink "$target"`

done

echo "`pwd -P`/$target"

)

}

script\_path=`real\_path "$0"`

sonar\_scanner\_home=`dirname "$script\_path"`/..

# make it fully qualified

sonar\_scanner\_home=`cd "$sonar\_scanner\_home" && pwd -P`

jar\_file=$sonar\_scanner\_home/lib/sonar-scanner-cli-4.6.2.2472.jar

# check that sonar\_scanner\_home has been correctly set

if [ ! -f "$jar\_file" ] ; then

echo "File does not exist: $jar\_file"

echo "'$sonar\_scanner\_home' does not point to a valid installation directory: $sonar\_scanner\_home"

exit 1

fi

use\_embedded\_jre=true

if [ "$use\_embedded\_jre" = true ]; then

export JAVA\_HOME="$sonar\_scanner\_home/jre"

fi

if [ -n "$JAVA\_HOME" ]

then

java\_cmd="$JAVA\_HOME/bin/java"

else

java\_cmd="`which java`"

fi

if [ -z "$java\_cmd" -o ! -x "$java\_cmd" ] ; then

echo "Could not find 'java' executable in JAVA\_HOME or PATH."

exit 1

fi

project\_home=`pwd`

#echo "Info: Using sonar-scanner at $sonar\_scanner\_home"

#echo "Info: Using java at $java\_cmd"

#echo "Info: Using classpath $jar\_file"

#echo "Info: Using project $project\_home"

exec "$java\_cmd" \

-Djava.awt.headless=true \

$SONAR\_SCANNER\_OPTS \

$SONAR\_SCANNER\_DEBUG\_OPTS \

-classpath "$jar\_file" \

-Dscanner.home="$sonar\_scanner\_home" \

-Dproject.home="$project\_home" \

org.sonarsource.scanner.cli.Main "$@"