

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>

The screenshot shows the Jenkins configuration interface for a job named 'v-profile'. The 'Source Code Management' section is open, showing a 'Git' repository configuration. The 'Repository URL' field is highlighted with a red arrow and contains the value 'https://github.com/betawins/VProfile-1.git'. The 'Credentials' dropdown is set to '- none -'. There are buttons for 'Save' and 'Apply' at the bottom.

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

We are successfully clone the repository.

github.com/betawins/VProfile-1

Gmail YouTube Maps

VProfile-1 Public
forked from naveen-e-git/VProfile-1

Code Pull requests Actions Projects Security Insights

Watch 0 Fork 64 Star 0

This branch is 3 commits ahead of naveen-e-git:VProfile-1:master.

betawins Update README.md

afc1371 · last year 69 Commits

- .github/workflows action commit 6 years ago
- ansible update 6 years ago
- src removing db.sql dump file 8 years ago
- Dockerfile Dockerfile 6 years ago
- README.md Update README.md last year
- azure-pipelines.yml Set up CI with Azure Pipelines 7 years ago
- pom.xml Update pom.xml last year

About

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.

Readme Activity 0 stars 0 watching 64 forks Report repository

Releases 2 tags

Packages

```
[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
hudson.model.UpdateCenter.xml jenkins.install.UpgradeWizard.state logs secret.key
hudson.plugins.git.GitTool.xml jenkins.model.JenkinsLocationConfiguration.xml nodeMonitors.xml secret.key.not-so-secret
identity.key.enc jenkins.telemetry.Correlator.xml plugins secrets
[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 storage as 20 gb.

aws Search [Alt+S]

Account ID: 2355-5102-84 muja

EC2 > Instances > Launch an instance

default sg-0880f310383562b69 X

VPC: vpc-06cf45eab13624fe

Security groups that you add or remove here will be added to or removed from all your network interfaces.

Advanced network configuration

Configure storage Info Advanced

1x 20 GB gp3 Root volume, 3000 IOPS, Not encrypted

Add new volume

Click refresh to view backup information

The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

File systems Edit

Advanced details Info

Summary Number of instances 1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.9.2...read more ami-0341d95f75f311023

Virtual server type (instance type)

c7i-flex.large

Firewall (security group)

default

Storage (volumes)

1 volume(s) - 20 GiB

Cancel Launch instance Preview code

- `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      Repository  Size
Installing:
mysql80-community-release    noarch       el9-1        @commandline 10 k
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.

```
[root@ip-172-31-45-79 ~]# sudo grep 'temporary password' /var/log/mysqld.log
2023-10-22T12:52:32.919953Z 6 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: iq?jXh?sg9WY
[root@ip-172-31-45-79 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.44

Copyright (c) 2000, 2025, 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> ]
```

You change the password for mysqld by using the command

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

```
[root@ip-172-31-45-79 ~]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.44

Copyright (c) 2000, 2025, 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 'Mujaheed2#';
Query OK, 0 rows affected (0.01 sec)

mysql> ]
```

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

```
creating: sonarqube-7.6/lib/jdbc/mysql/
inflating: sonarqube-7.6/lib/jdbc/mysql/mysql-connector-java-5.1.46.jar
creating: sonarqube-7.6/lib/jdbc/postgresql/
inflating: sonarqube-7.6/lib/jdbc/postgresql/postgresql-42.2.5.jar
creating: sonarqube-7.6/lib/jdbc/h2/
inflating: sonarqube-7.6/lib/jdbc/h2/h2-1.3.176.jar
creating: sonarqube-7.6/elasticsearch/plugins/
root@ip-172-31-45-79 opt]# ls
ws  sonarqube-7.6  sonarqube-7.6.zip
root@ip-172-31-45-79 opt]# mv /opt/sonarqube-7.6 /opt/sonar
root@ip-172-31-45-79 opt]# ls
ws  sonar  sonarqube-7.6.zip
root@ip-172-31-45-79 opt]# [REDACTED]
```

Switch to mysql data and create a database

- CREATE DATABASE sonar CHARACTER SET utf8 COLLATE utf8_general_ci;

```
[ec2-user@ip-172-31-45-79 ~]$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.44 MySQL Community Server - GPL

Copyright (c) 2000, 2025, 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, 2 warnings (0.00 sec)
```

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
```

Continuous Code Quality

[Log In](#) [Read documentation](#)

Projects Analyzed: 0

0 Bugs
0 Vulnerabilities
0 Code Smells

Multi-Language

20+ programming languages are supported by SonarQube thanks to our in-house code analyzers, including:

Java	C/C++	C#	COBOL	ABAP	HTML	RPG	JavaScript	Objective C	XML
VB.NET	PL/SQL	Flex	Python	Groovy	PHP	Swift	Visual Basic	PL/I	

Quality Model

Bugs track code that is demonstrably wrong or highly likely to yield unexpected behavior.

Vulnerabilities are raised on code that is potentially vulnerable to exploitation by hackers.

Code Smells will confuse maintainers or give them pause. They are measured primarily in terms of the time they will take to fix.

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

The screenshot shows the Jenkins plugin manager interface. On the left, there's a sidebar with options like 'Updates', 'Available plugins' (which is selected), 'Installed plugins', 'Advanced settings', and 'Download progress'. The main area has a search bar at the top with the query 'sonar'. Below it is a table with columns: 'Install', 'Name', 'Released', and 'Health'. There are four rows of results:

Install	Name	Released	Health
<input type="checkbox"/>	Personalization for Blue Ocean 1.27.23	14 days ago	97
<input checked="" type="checkbox"/>	SonarQube Scanner 2.18	9 mo 3 days ago	84
<input type="checkbox"/>	Sonar Quality Gates 352.vdcdb_d7994fb_6	8 mo 7 days ago	96
<input type="checkbox"/>	Quality Gates 2.5	9 yr 5 mo ago	42

A red callout box highlights the 'SonarQube Scanner' row, indicating it is selected for installation. A warning message at the bottom of the table says: 'Warning: This plugin version may not be safe to use. Please review the following security notices.' followed by a link: 'Check details committed to this build'.

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 .

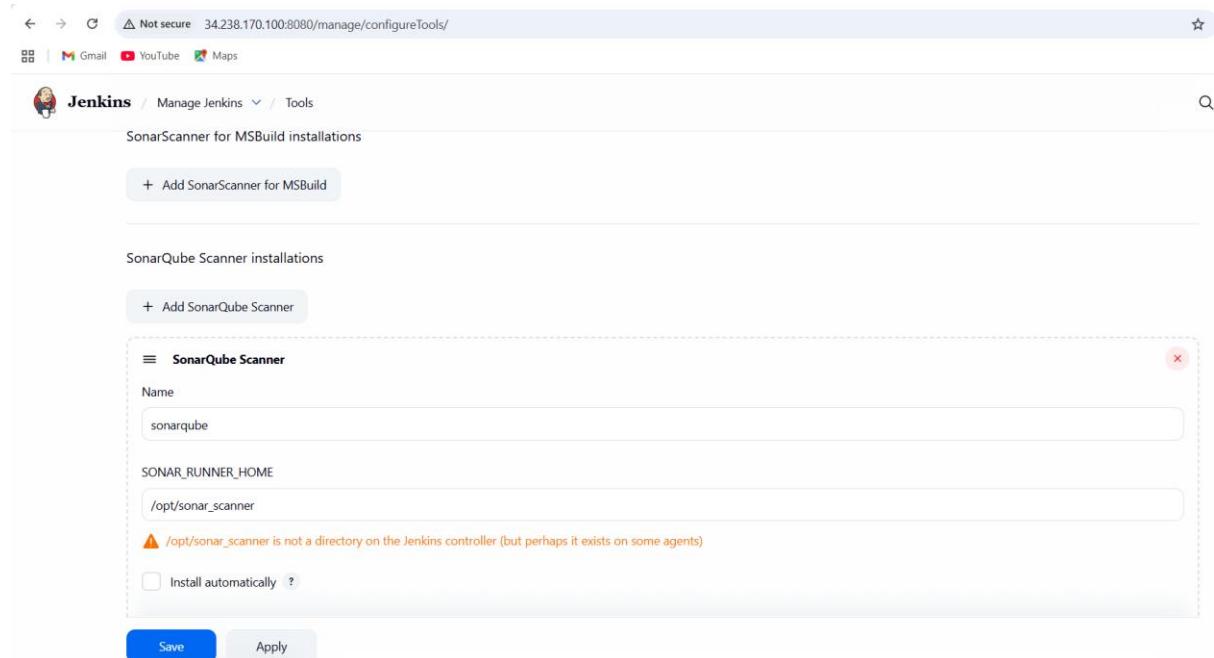
The screenshot shows the Jenkins 'System' configuration page. At the top, there's a section for 'SonarQube servers'. It includes a note: 'If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.' with an unchecked checkbox next to 'Environment variables'. Below this is a 'SonarQube installations' section with a note: 'List of SonarQube installations'. It contains a table with a single row:

Name
sonarqube

Below the table are fields for 'Server URL' (with a default value of `http://localhost:9000`) and 'Server authentication token' (with a placeholder 'secret'). At the bottom are 'Save' and 'Apply' buttons.

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

The screenshot shows the Jenkins job configuration interface. On the left, a sidebar lists configuration sections: General, Source Code Management, Triggers, Environment (which is selected), Build Steps, and Post-build Actions. A red arrow points from the text above to the 'Execute SonarQube Scanner' option in a context menu that has popped up over the 'Build Steps' section. The menu also includes options like Execute Windows batch command, Execute shell, Invoke Ant, Invoke Gradle script, Invoke top-level Maven targets, Run with timeout, Set build status to "pending" on GitHub commit, SonarScanner for MSBuild - Begin Analysis, and SonarScanner for MSBuild - End Analysis. Below the menu, there is a note about SonarScanner for MSBuild being used for code compilation, testing, and deployment. At the bottom of the menu is a '+ Add build step' button.

In analysis properties give this code and save it.

sonar.projectKey=Sabear

sonar.projectName=Sabear

sonar.projectVersion=1.0

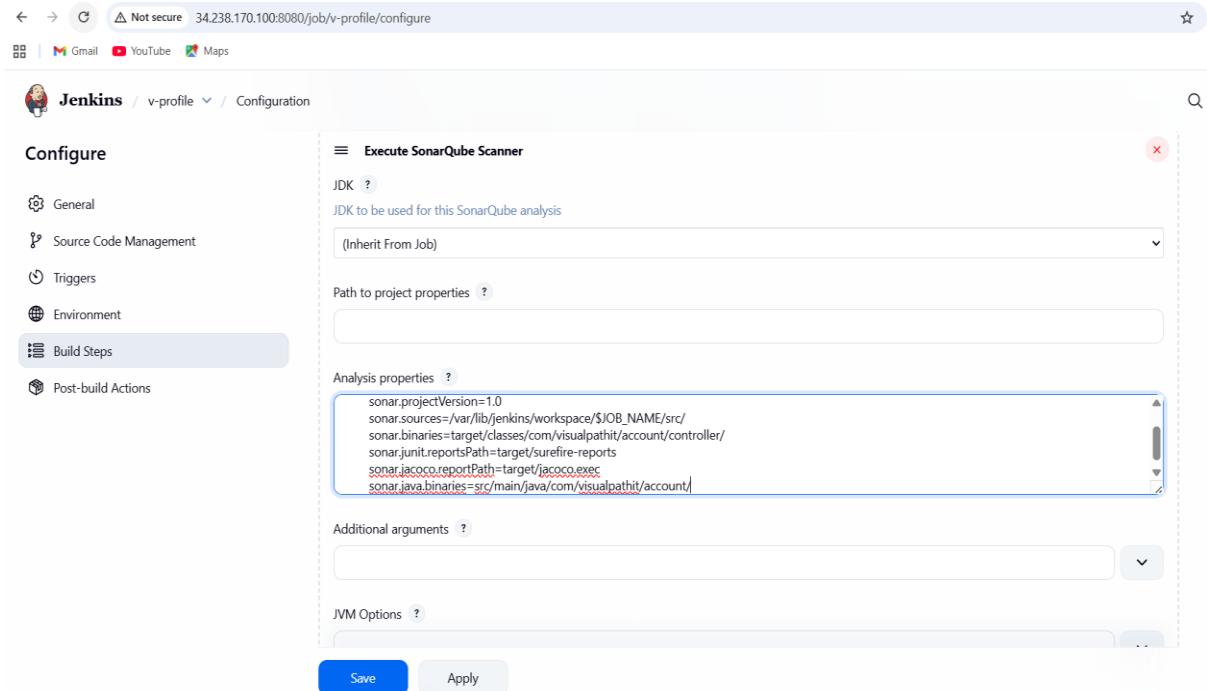
sonar.sources=/var/lib/jenkins/workspace/\$JOB_NAME/src
/

sonar.binaries=target/classes/com/visualpathit/account/con
troller/

sonar.junit.reportsPath=target/surefire-reports

sonar.jacoco.reportPath=target/jacoco.exec

sonar.java.binaries=src/main/java/com/visualpathit/accoun
t/



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-scanner-4.6.2.2472-linux.zip
--2025-10-28 14:25:15-- https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-4.6.2.2472-linux.zip
Resolving binaries.sonarsource.com (binaries.sonarsource.com)... 99.84.188.45, 99.84.188.21, 99.84.188.1
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-linux.zip 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.

The screenshot shows a Jenkins job named "first_job". The top navigation bar includes links for Gmail, YouTube, and Maps. The main content area displays the SonarQube Quality Gate status, which is "Passed" (green). Below this, there is a "Permalinks" section with a bulleted list of recent builds. At the bottom, a "Builds" table lists two entries: build #7 (Passed, 9:23 AM) and build #6 (Failed, 9:18 AM).

Build	Status	Time
#7	Passed	9:23 AM
#6	Failed	9:18 AM

- **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.

The screenshot shows the Jenkins job configuration page for 'first_job'. The left sidebar lists 'General', 'Source Code Management', 'Triggers', 'Environment', 'Build Steps', and 'Post-build Actions'. A red arrow points from the 'Post-build Actions' link to the expanded dropdown menu. The menu includes options like 'Aggregate downstream test results', 'Archive the artifacts', 'Build other projects', etc., and 'Slack Notifications' is highlighted.

The screenshot shows the expanded 'Post-build Actions' section. It displays a list of notification types: 'Notify Build Start' (checked), 'Notify Success' (checked), 'Notify Aborted' (checked), 'Notify Not Built' (unchecked), 'Notify Unstable' (unchecked), 'Notify Regression' (unchecked), 'Notify Every Failure' (unchecked), and 'Notify Back To Normal' (unchecked). An 'Advanced' button is visible at the bottom.

Task 2

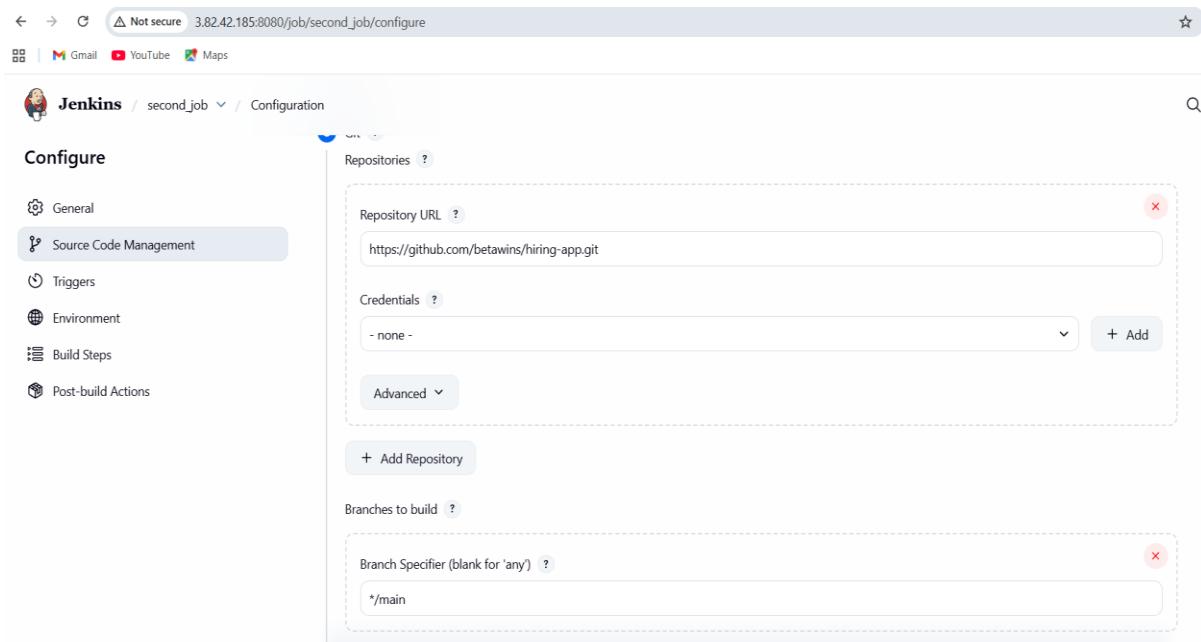
Create one Jenkins job using the below code and create three stages:

- **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.

The screenshot shows the AWS EC2 'Launch an instance' wizard. The first step, 'Name and tags', has 'sonarqube' entered in the Name field. The second step, 'Application and OS Images (Amazon Machine Image)', shows a search bar and a list of AMIs including Amazon Linux 2023.9.2. The third step, 'Configure storage', shows a 20 GiB gp3 root volume. The fourth step, 'Summary', displays the selected configuration: 1 instance, AMI (Amazon Linux 2023.9.2), instance type (t2-flexible), security group (default), and storage (1 volume - 20 GiB). A 'Launch instance' button is at the bottom right.

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%[=====] 234 MB/s   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 specifies which 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

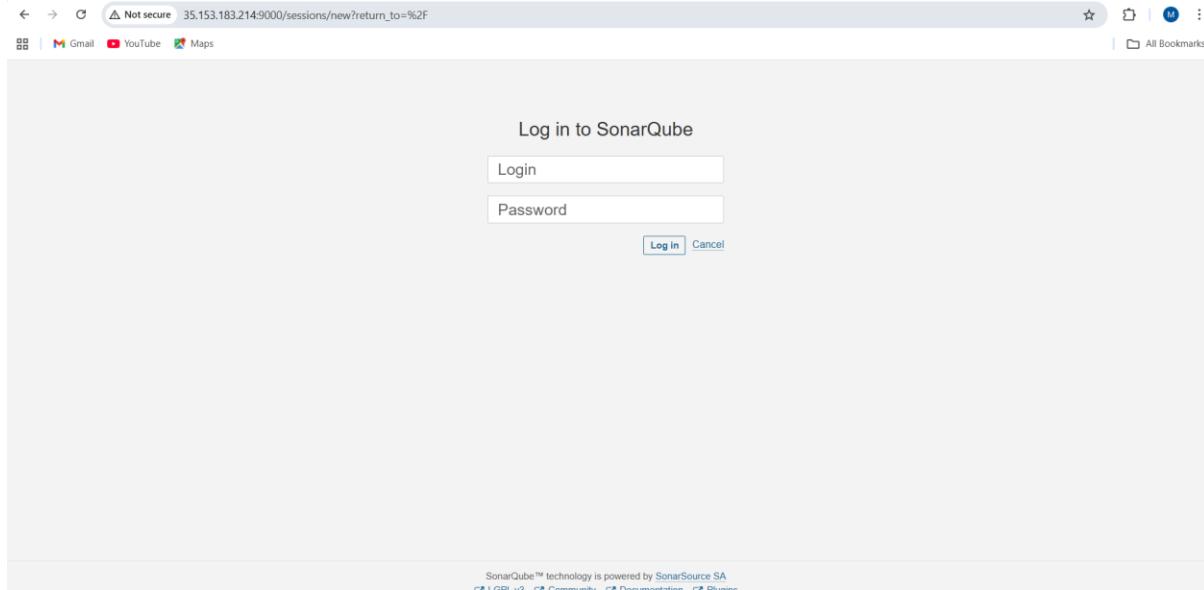
- cd /opt
- sudo chown -R ec2-user:ec2-user sonar

```
[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

You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. [Learn More](#)

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

Administrator

My Account

Log out

Sonar way BUILT-IN

This quality gate complies with Clean as You Code

This quality gate complies with the [Clean as You Code](#) methodology, so that you benefit from the most efficient approach to delivering Clean Code. It ensures that:

- No new bugs are introduced
- No new vulnerabilities are introduced
- All new security hotspots are reviewed
- New code has limited technical debt
- New code has limited duplication
- New code is properly covered by tests

Conditions

Conditions on New Code

Generate a token in the security field.

You're running a version of SonarQube that is no longer active. Please upgrade to an active version immediately. [Learn More](#)

sonarcube Projects Issues Rules Quality Profiles Quality Gates Administration

Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Expires in			
Enter Token Name	Select Token Type	30 days			
New token "sonarqube-container" has been created. Make sure you copy it now, you won't be able to see it again!					
Copy sqa_df3edd309f4cd9883c33e8ebbc96b84f3872abd					
Name	Type	Project	Last use	Created	Expiration
sonarqube-contai...	Global		Never	October 27, 2025	November 26, 2025
Revoke					

Install sonarqube scanner in plugins.

The screenshot shows the Jenkins plugin manager interface. The left sidebar has links for Plugins, Updates, Available plugins, Installed plugins, Advanced settings, and Download progress. The main area lists installed plugins with their status: Git (Success), EDDSA API (Success), Trilead API (Success), SSH Build Agents (Success), Matrix Authorization Strategy (Success), LDAP (Success), jsoup API (Success), Email Extension (Success), Mailer (Success), Theme Manager (Success), Dark Theme (Success), Loading plugin extensions (Success), SonarQube Scanner (Success), and Loading plugin extensions (Success). Below the list are two buttons: 'Go back to the top page' and 'Restart Jenkins when installation is complete and no jobs are running'.

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 .

The screenshot shows the Jenkins system configuration page under 'SonarQube installations'. It includes fields for 'Name' (set to 'secret'), 'Server URL' (set to 'http://54.198.71.141:9000'), and 'Server authentication token' (set to 'secret'). There is also an 'Advanced' dropdown and buttons for 'Save' and 'Apply'.

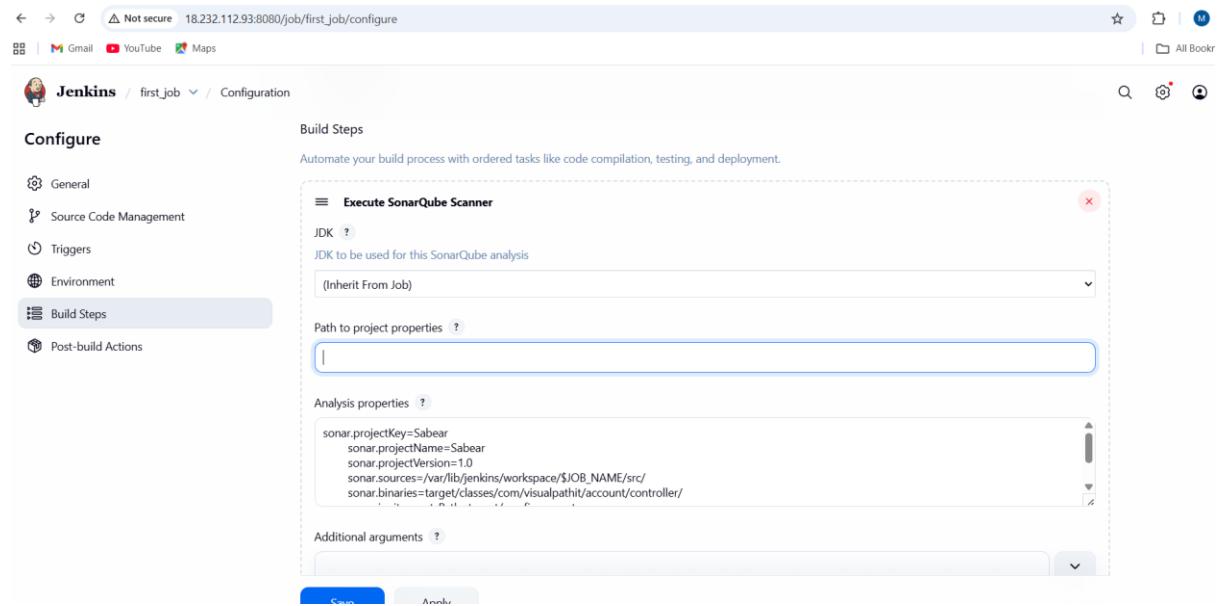
Go to your job and go to build steps click on execute sonarqube scanner and give the analysis properties of your code and give the credentials and save it.

The screenshot shows the Jenkins job configuration page for 'first_job'. On the left, there are tabs for 'Configure', 'General', 'Source Code Management', 'Triggers', 'Environment' (which is selected), 'Build Steps', and 'Post-build Actions'. In the main area, under 'Build Steps', there is a list of available steps: 'Add build step', 'Filter', 'Execute SonarQube Scanner', 'Execute Windows batch command', 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', and 'Run with timeout'. A red arrow points to the 'Execute SonarQube Scanner' option.

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

The screenshot shows the Jenkins 'Manage Jenkins' page under the 'Tools' tab. It displays the 'SonarScanner for MSBuild installations' section. There is a button '+ Add SonarScanner for MSBuild'. Below it, there is a 'SonarQube Scanner installations' section with a button '+ Add SonarQube Scanner'. A modal window titled 'SonarQube Scanner' is open, showing fields for 'Name' (set to 'sonarqube-container') and 'SONAR_RUNNER_HOME' (set to '/opt/sonar_scanner'). A checkbox 'Install automatically' is also present. At the bottom of the modal are 'Save' and 'Apply' buttons. The background shows the Jenkins interface with various icons and a toolbar at the bottom.

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



The screenshot shows the Jenkins job configuration page for 'first_job'. The left sidebar has 'Build Steps' selected. A single step, 'Execute SonarQube Scanner', is listed under 'Build Steps'. It includes fields for 'JDK' (set to '(Inherit From Job)'), 'Path to project properties' (empty), and 'Analysis properties' (containing 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/). At the bottom are 'Save' and 'Apply' buttons.

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.

The screenshot shows a Jenkins job named "second_job". On the left, there's a sidebar with various Jenkins management links like Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, Rename, and Credentials. The main content area displays the SonarQube Quality Gate results for the project "second_job". It shows a green checkmark icon and the text "second_job". Below that is the "SonarQube Quality Gate" section, which includes the project name "Sabear", a "Passed" status badge, and a "Success" message indicating "server-side processing". A "Permalinks" section lists recent builds. At the bottom, there's a "Builds" table with one entry (#1) from today at 10:47 AM.

REST API Jenkins 2.52

The screenshot shows the SonarQube interface for the project "Sabear". The top navigation bar includes "Projects", "Issues", "Rules", "Quality Profiles", "Quality Gates", and "Administration". A search bar and a "Create Project" button are also present. The main content area displays the project analysis results for "Sabear", which has passed. It shows various metrics: Bugs (17 C), Vulnerabilities (0 A), Hotspots Reviewed (0.0% E), Code Smells (34 A), Coverage (0.0%), Duplications (12.9%), and Lines (1.7k S). A sidebar on the left contains filters for Quality Gate (Passed 1, Failed 0), Reliability (A rating 0, B rating 0, C rating 1, D rating 0, E rating 0), and Security (A rating 1, B rating 0, C rating 0, D rating 0, E rating 0). A banner at the bottom right encourages upgrading to an active version of SonarQube.

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

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

The screenshot shows the Jenkins plugin manager interface. In the search bar at the top, the text "sl" is typed. Below the search bar, there is a table with columns: "Install", "Name ↓", "Released", and "Health". One row is highlighted with a blue checkmark in the "Install" column, indicating it is selected. The row details the "Slack Notification" plugin, version 795.v4b_9705b_e6d47, released 2 months and 24 days ago, with a green health status of 100. Another row for "Job DSL 1.93" is shown below, with a note that it is up for adoption and has a health status of 80.

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

The screenshot shows the Jenkins configuration page for a job named "first_job". On the left, a sidebar lists "Configure" sections: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. The "Post-build Actions" section is currently selected and highlighted with a grey background. A red arrow points from the text "select slack notifications" in the previous step to the "Slack Notifications" option in the dropdown menu. The dropdown menu also includes other actions like Aggregate downstream test results, Archive the artifacts, and Build other projects. At the bottom of the configuration page, there are "Save" and "Apply" buttons.

The screenshot shows the Jenkins 'Configuration' page for a job named 'first_job'. On the left, a sidebar lists various configuration sections: General, Source Code Management, Triggers, Environment, Build Steps, and Post-build Actions. 'Post-build Actions' is currently selected and highlighted with a grey background. On the right, under the heading 'Define what happens after a build completes, like sending notifications, archiving artifacts, or triggering other jobs.', there is a section titled 'Slack Notifications' with the following options:

- Notify Build Start
- Notify Success
- Notify Aborted
- Notify Not Built
- Notify Unstable
- Notify Regression
- Notify Every Failure
- Notify Back To Normal

A small 'Advanced' dropdown menu is visible at the bottom right of the configuration area.

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.