

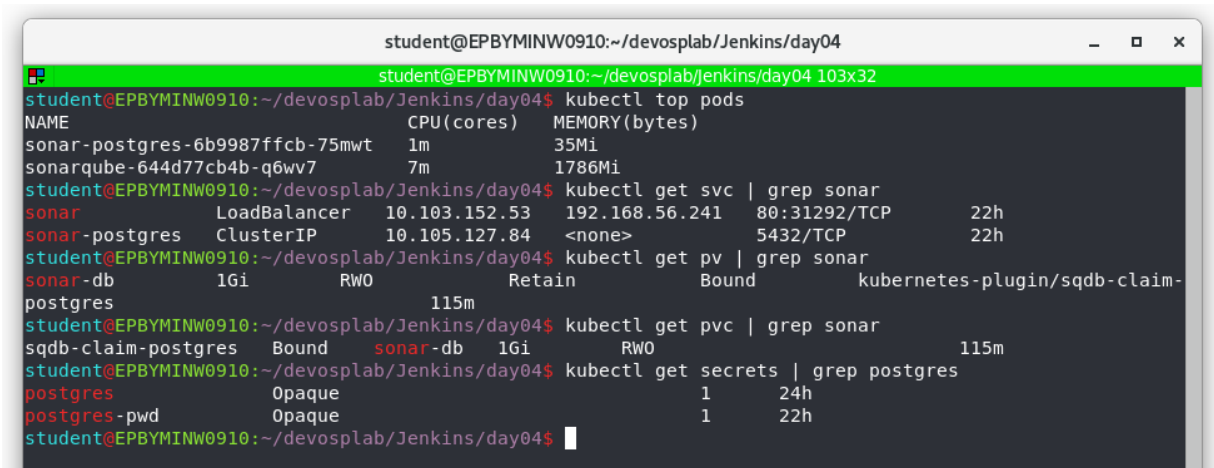
Task 1. Automated Code Analysis

Review

SonarQube and Postgres will be installed (via ansible). Public Java-based project will be setup in Jenkins job with Sonar analysis

Task

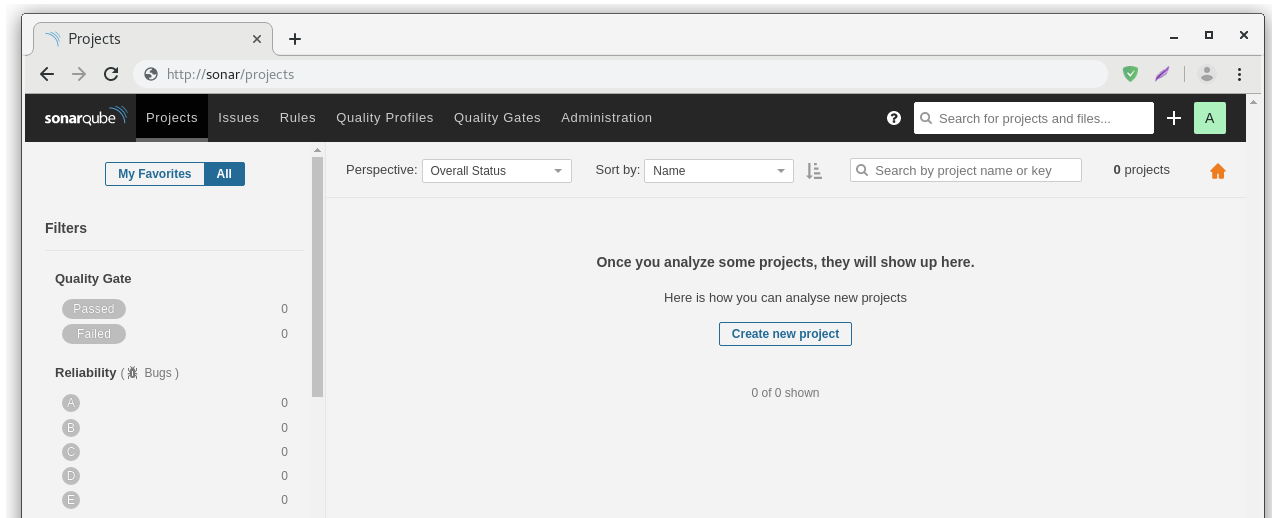
1. Develop kubernetes definitions for the deployments:
 - a) PostgreSQL
 - b) Sonar server
 - c) Service/Ingress



```
student@EPBYMINW0910:~/devosplab/Jenkins/day04
student@EPBYMINW0910:~/devosplab/Jenkins/day04 103x32
student@EPBYMINW0910:~/devosplab/Jenkins/day04$ kubectl top pods
NAME                                CPU(cores)   MEMORY(bytes)
sonar-postgres-6b9987ffcb-75mwt     1m           35Mi
sonarqube-644d77cb4b-q6wv7          7m           1786Mi
student@EPBYMINW0910:~/devosplab/Jenkins/day04$ kubectl get svc | grep sonar
sonar                               LoadBalancer  10.103.152.53   192.168.56.241   80:31292/TCP    22h
sonar-postgres                      ClusterIP      10.105.127.84   <none>           5432/TCP        22h
student@EPBYMINW0910:~/devosplab/Jenkins/day04$ kubectl get pv | grep sonar
sonar-db                            1Gi          RWO            Retain          Bound          kubernetes-plugin/sqdb-claim-
postgres                            115m
student@EPBYMINW0910:~/devosplab/Jenkins/day04$ kubectl get pvc | grep sonar
sqdb-claim-postgres                Bound        sonar-db        1Gi            RWO            115m
student@EPBYMINW0910:~/devosplab/Jenkins/day04$ kubectl get secrets | grep postgres
postgres                           Opaque       1              24h
postgres-pwd                       Opaque       1              22h
student@EPBYMINW0910:~/devosplab/Jenkins/day04$
```

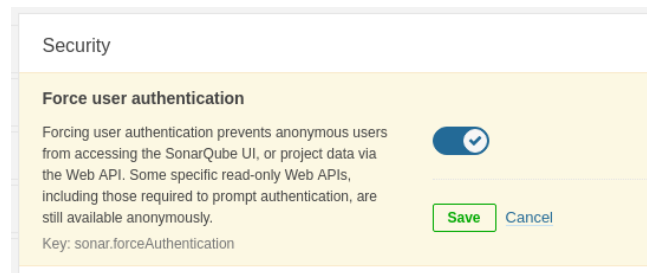
Picture 1.1

Configure Sonar with Nginx, so that it accessible via <http://sonar>



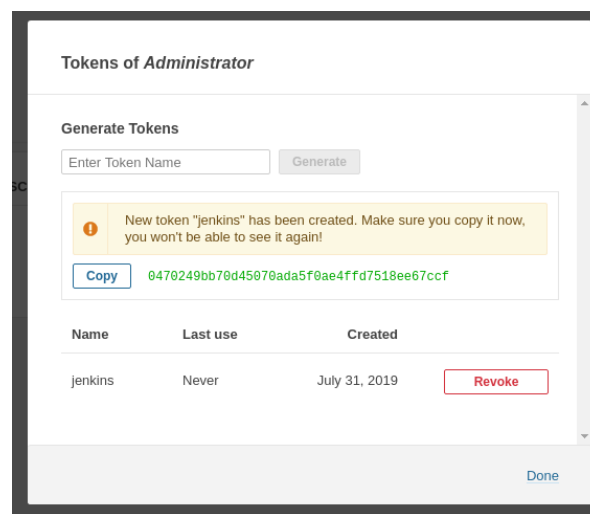
Picture 1.2

2. Configure Sonar security – limit anonymous access to instance

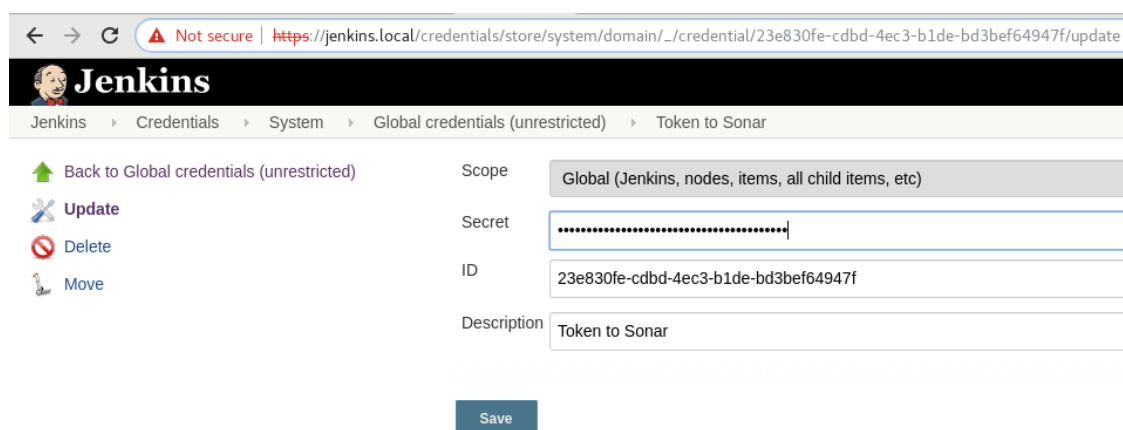


Picture 2

3. Configure Jenkins for Sonar analysis



Picture 3.1 – Generate Token on SonarQube



Picture 3.2 – Add Sonar Token to Jenkins

configuration

SonarQube servers

Environment variables ☒ Enable injection of SonarQube server configuration as build environment variables
If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

Server URL
Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

List of SonarQube installations

Picture 3.2 – Jenkins Configure System

Global Tool Configuration

SonarQube Scanner

List of SonarScanner for MSBuild installations on this system

SonarQube Scanner installations

☒ Install automatically

Picture 3.3 – Jenkins Global Tools Configuration

4. Create Sonar analysis profile and add/modify/remove a couple of rules. Set it as the default profile.

The screenshot shows the 'Quality Profiles / Java' configuration page for a profile named 'JenkinsTest'. At the top right, it indicates 'Updated: 1 minute ago' and 'Used: Never', with buttons for 'Changelog' and a settings icon. The main content is divided into three sections: 'Rules', 'Inheritance', and 'Projects'. The 'Rules' section contains a table with columns 'Rules', 'Active', and 'Inactive'. Below the table is a yellow box stating 'Sonar way rules not included' with a count of 381 and an 'Activate More' button. The 'Inheritance' section shows 'JenkinsTest' as the parent profile with '0 active rules' and '0 overridden rules', and a 'Change Parent' button. The 'Projects' section lists 'JenkinsTest' as the only project with a 'Change Projects' button. A 'Permissions' section at the bottom shows that the 'Administrator' user has the 'Manage Quality Profile' permission, with a 'Grant permissions to more users' button.

Rules	Active	Inactive
Total	0	537
Bugs	0	121
Vulnerabilities	0	44
Code Smells	0	340
Security Hotspots	0	32

Sonar way rules not included 381

Permissions

Users with the global "Manage Quality Profile" permission can manage this quality profile.

A Administrator
admin

Grant permissions to more users

Picture 4

5. Choose some open-source project for building with Jenkins and create a regular build job (for example...<https://github.com/MNT-Lab/build-tools/tree/master/helloworld-project/helloworld-ws>).

The screenshot shows the 'Build' configuration page in Jenkins. It has two input fields: 'Root POM' with the value 'helloworld-ws/pom.xml' and 'Goals and options' with the value 'build'. Both fields have a help icon to their right. At the bottom right, there is an 'Advanced...' button.

Build

Root POM

Goals and options

Advanced...

Picture 5

6. Add Sonar analysis in Jenkins job configuration.

The screenshot shows the 'Pre Steps' configuration in Jenkins. The 'Execute SonarQube Scanner' step is selected. The configuration includes the following fields:

- Task to run:** A text input field.
- JDK:** A dropdown menu set to '(Inherit From Job)'. Below it, a note says 'JDK to be used for this SonarQube analysis'.
- Path to project properties:** A text input field.
- Analysis properties:** A text area containing the following properties:

```
sonar.projectKey=MNT-project
sonar.projectName=MNT project
sonar.projectVersion=1.0
sonar.login=admin
sonar.password=admin
sonar.sources=/var/jenkins_home/workspace/TestSonar/helloworld-ws/src/main/java/
```
- Additional arguments:** A text input field with a dropdown arrow.
- JVM Options:** A text input field with a dropdown arrow.

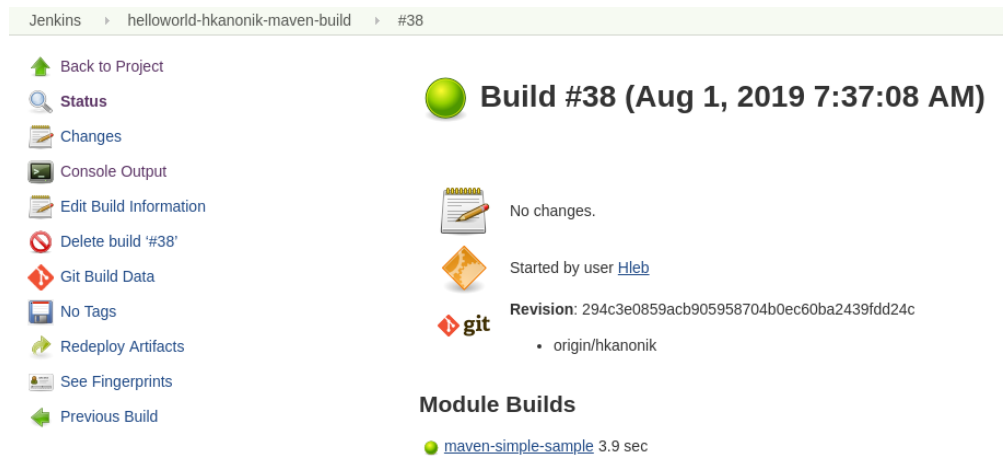
At the bottom, there is a button labeled 'Add pre-build step' with a dropdown arrow.

Picture 6

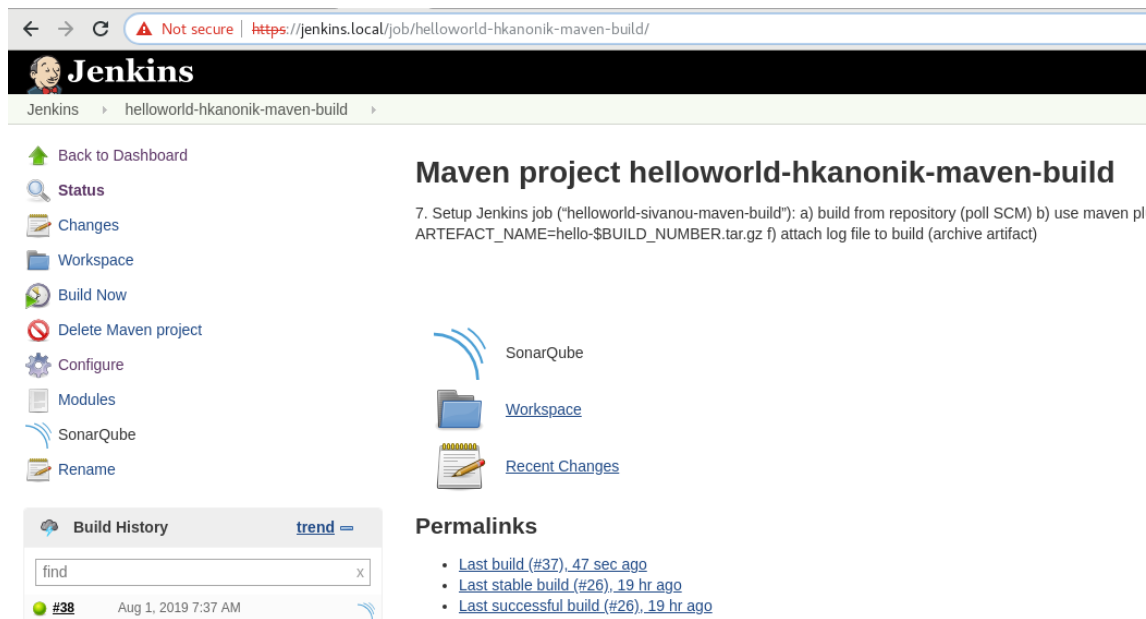
7. Add some lines in the code and Run configured job several times to see the difference.
8. Add some lines with bug (for example infinite loop (endless loop)) and check the results.

```
INFO: Project base dir: /var/jenkins_home/workspace/helloworld-hkanonik-maven-build
INFO: ----- Scan MNT project
INFO: Load server rules
INFO: Load server rules (done) | time=60ms
INFO: Base dir: /var/jenkins_home/workspace/helloworld-hkanonik-maven-build
INFO: Working dir: /var/jenkins_home/workspace/helloworld-hkanonik-maven-build/.scannerwork
INFO: Source paths: /var/jenkins_home/workspace/helloworld-hkanonik-gradle-build/home-task/src/main/java/com/test
INFO: Source encoding: UTF-8, default locale: en
INFO: Index files
WARN: File '/var/jenkins_home/workspace/helloworld-hkanonik-gradle-build/home-task/src/main/java/com/test/Project.java' is ignored. It is not located
in module basedir '/var/jenkins_home/workspace/helloworld-hkanonik-maven-build'.
INFO: 0 files indexed
INFO: Sensor SonarJavaXmlFileSensor [java]
INFO: Sensor SonarJavaXmlFileSensor [java] (done) | time=0ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=0ms
INFO: Sensor CPD Block Indexer
INFO: Sensor CPD Block Indexer (done) | time=0ms
INFO: Calculating CPD for 0 files
INFO: CPD calculation finished
INFO: Analysis report generated in 139ms, dir size=23 KB
INFO: Analysis reports compressed in 7ms, zip size=5 KB
INFO: Analysis report uploaded in 81ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://192.168.56.241/sonar/dashboard/index/MNT-project
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://192.168.56.241/sonar/api/ce/task?id=AWxMHIWa5Y9jqPxjS15r
INFO: Task total time: 2.837 s
```

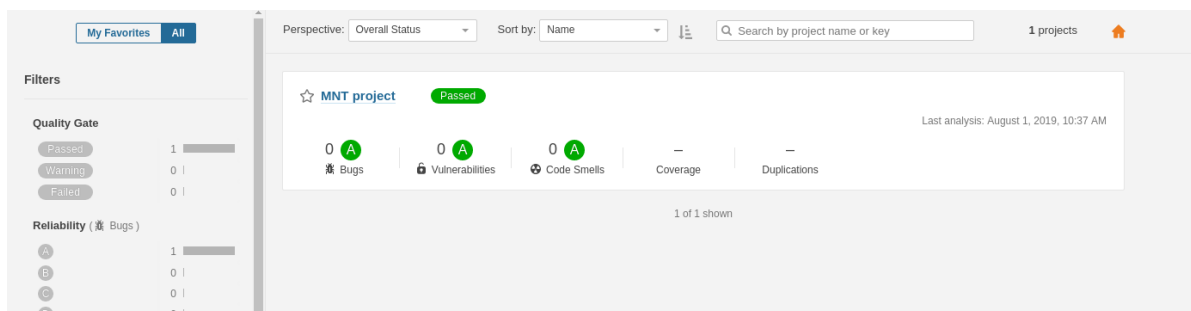
Picture 7-8.1



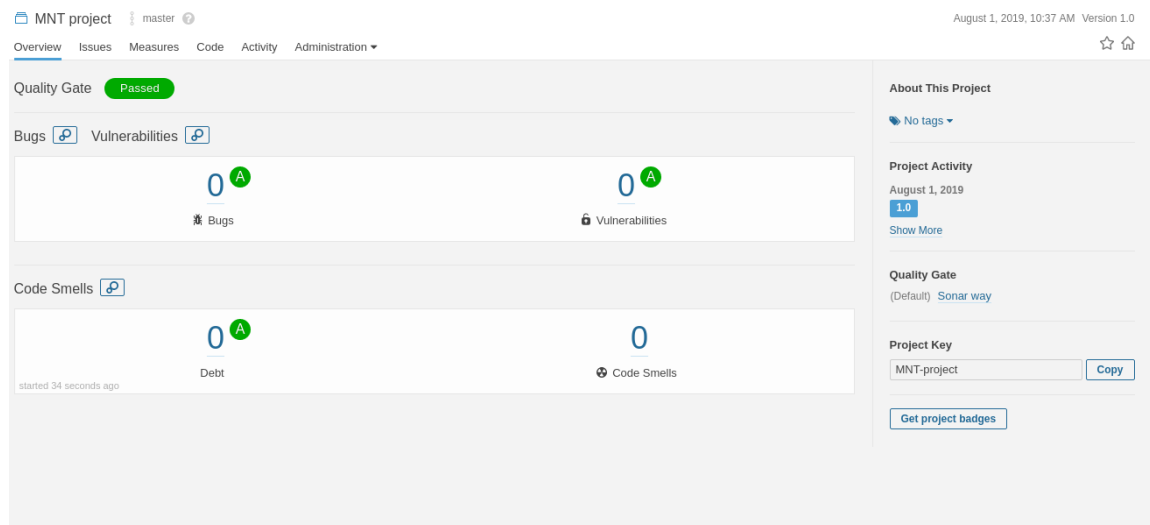
Picture 7-8.2



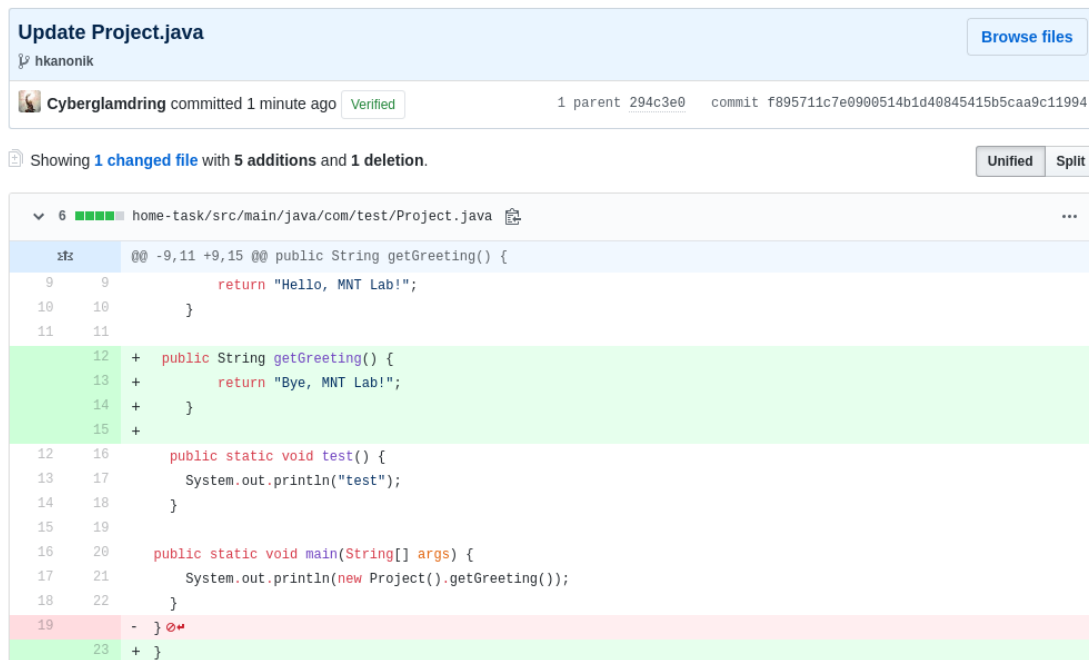
Picture 7-8.3



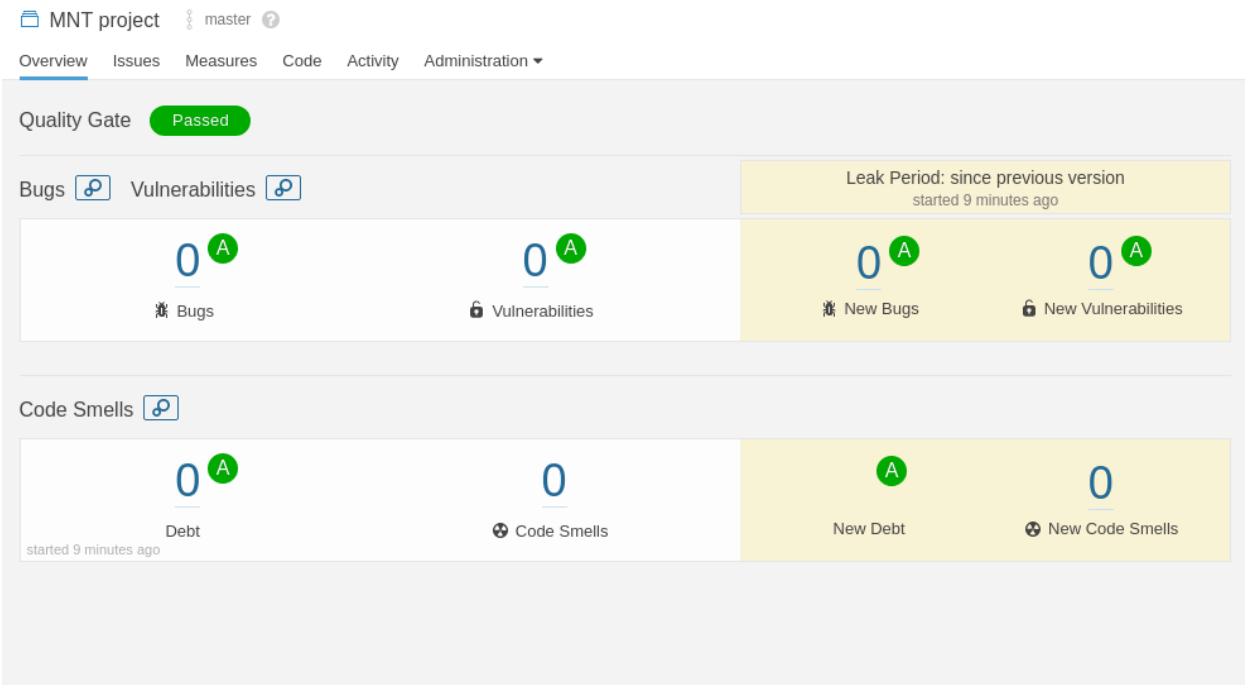
Picture 7-8.4



Picture 7-8.5



Picture 7-8.6 – Update code

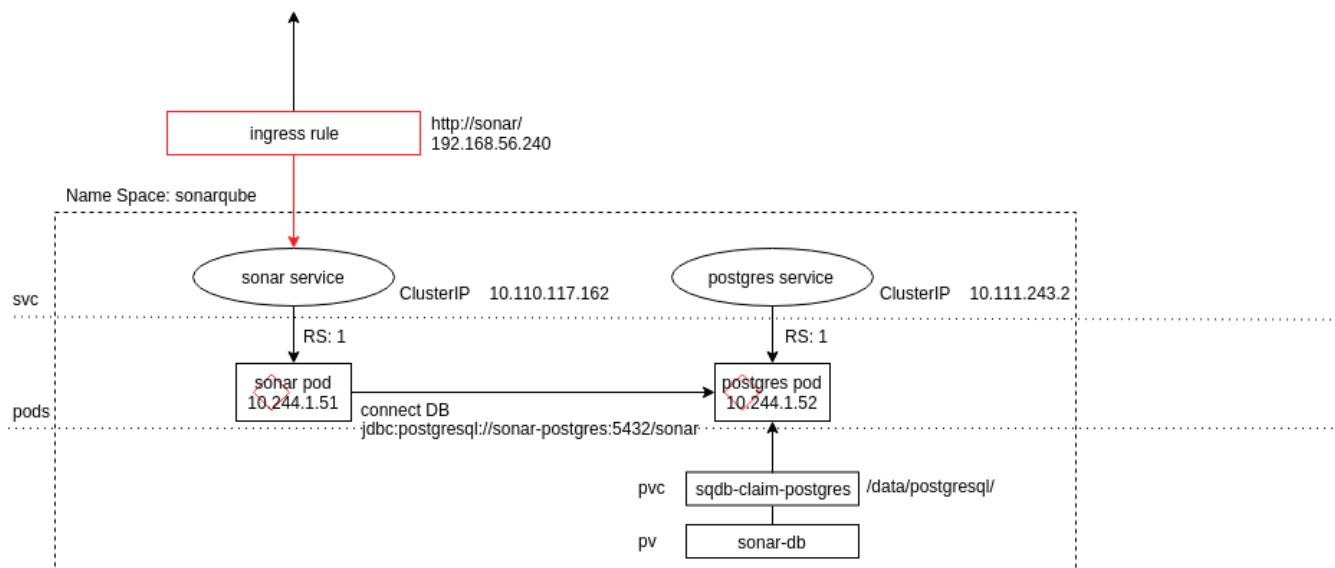


Picture 7-8.7 – Changes in sonar

9. Prove by screenshots and links to GitHub.

Link: <https://github.com/MNT-Lab/build-t00ls/tree/hkanonik>

10. Create a diagram which describes the interaction between the items in Kubernetes (pods: Jenkins, sonar; their services, ingresses, etc)



Picture 9