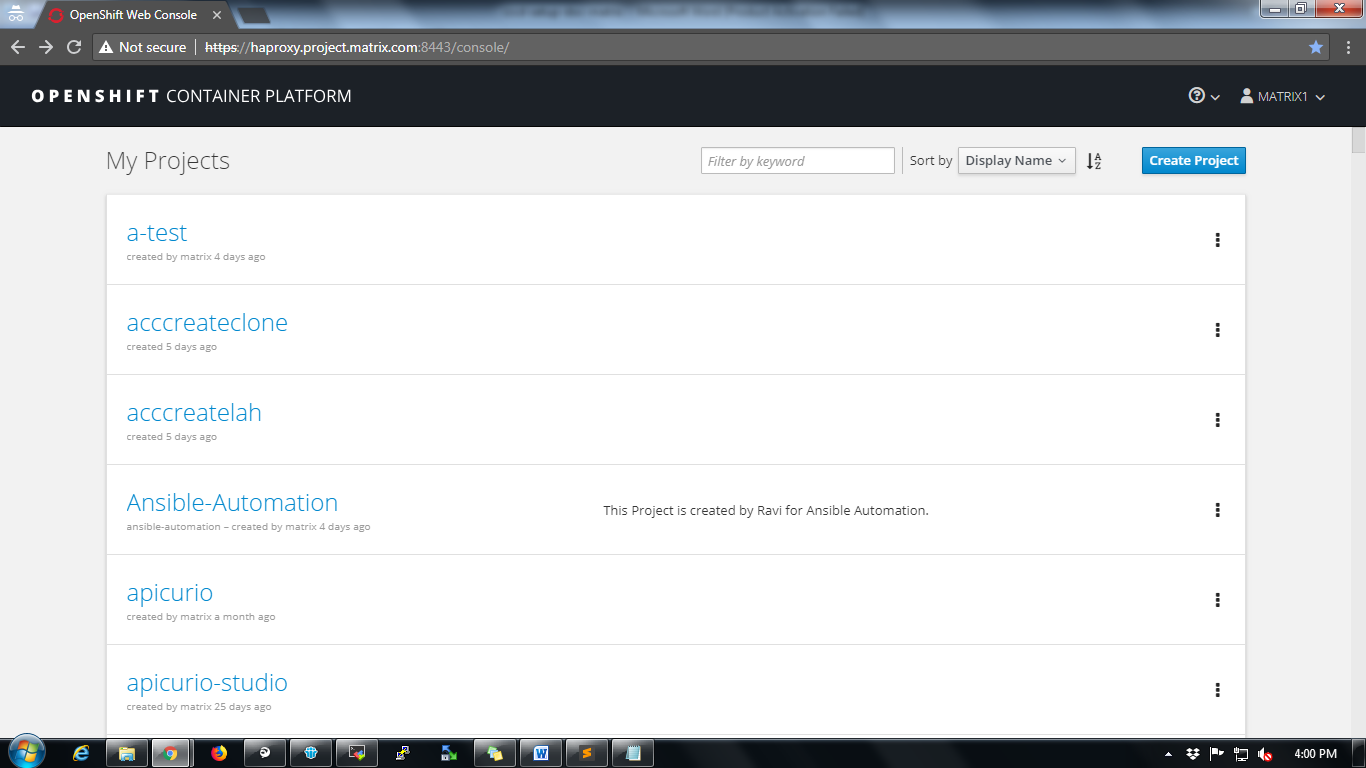
**Setup Sonarqube v7.1 in OCP v3.6**

Author:

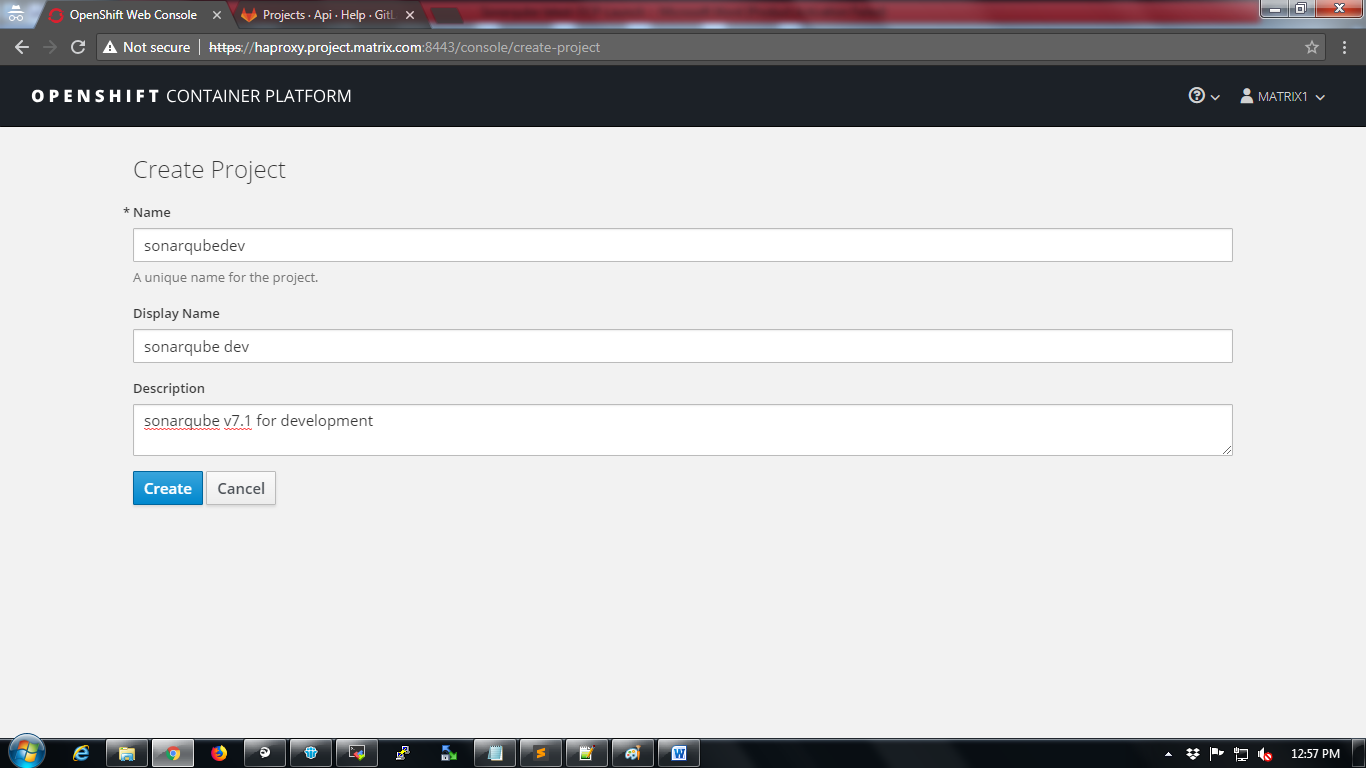
Raghuveer Muthoju

Login into OpenShift UI console.

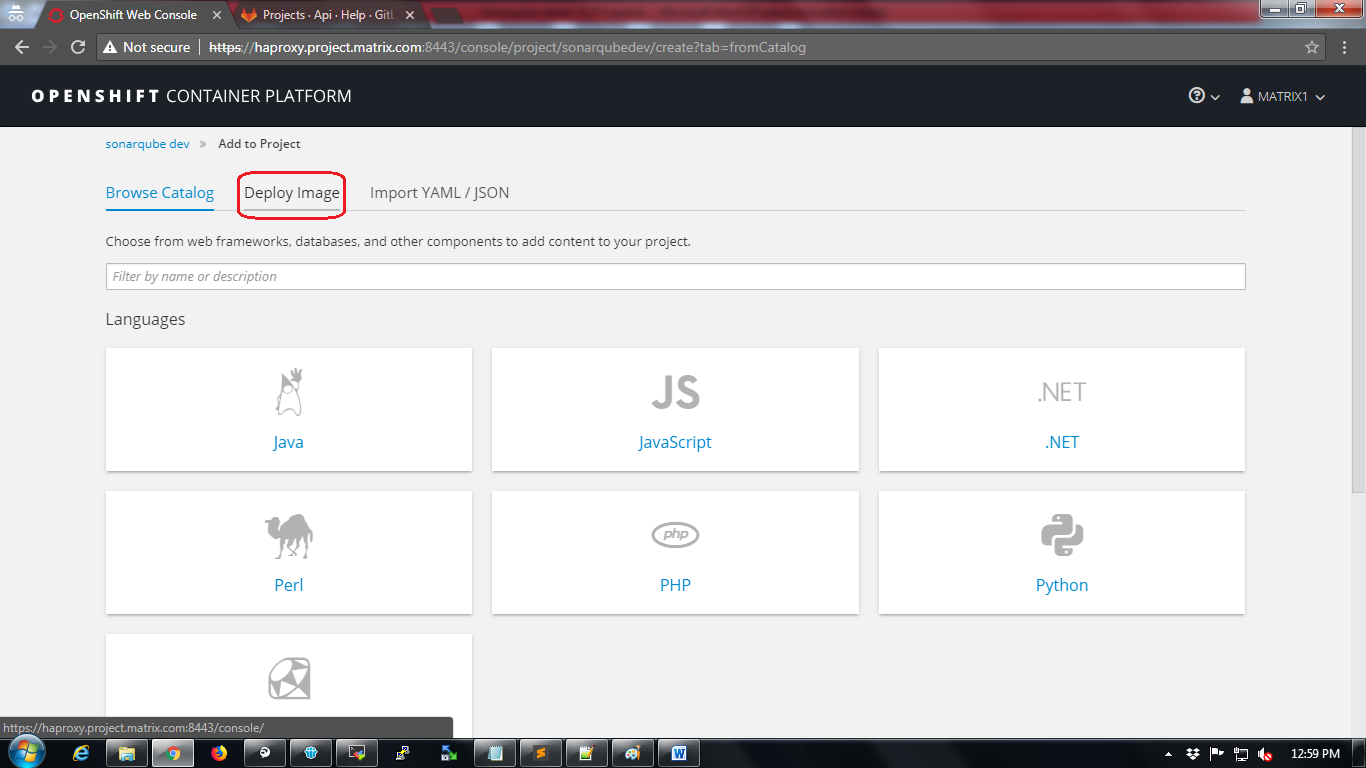
Click on Create Project button.



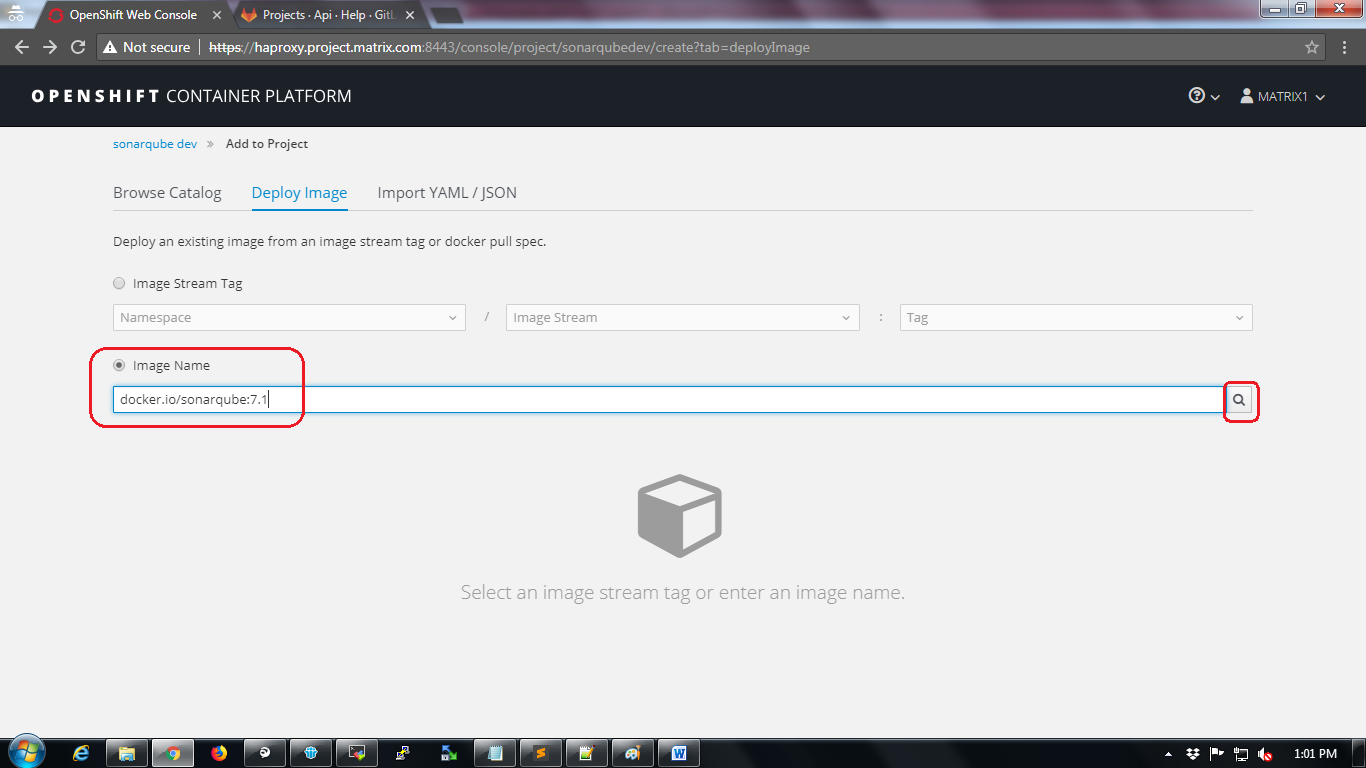
Fill up project details and click on Create button.



Click on “Deploy Image” tab.

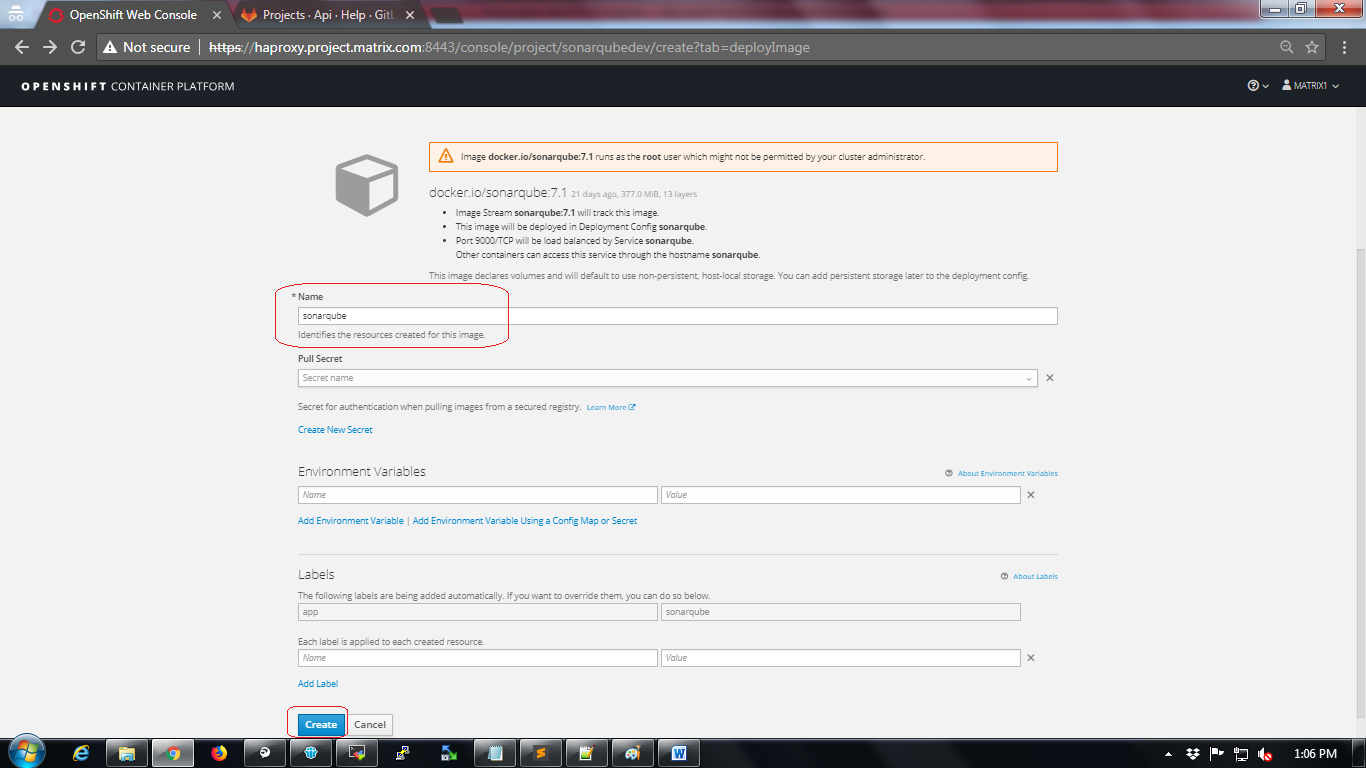


Select “Image Name” radio button and type “docker.io/sonarqube:7.1” in the search box and click on search button as shown in the below screenshot

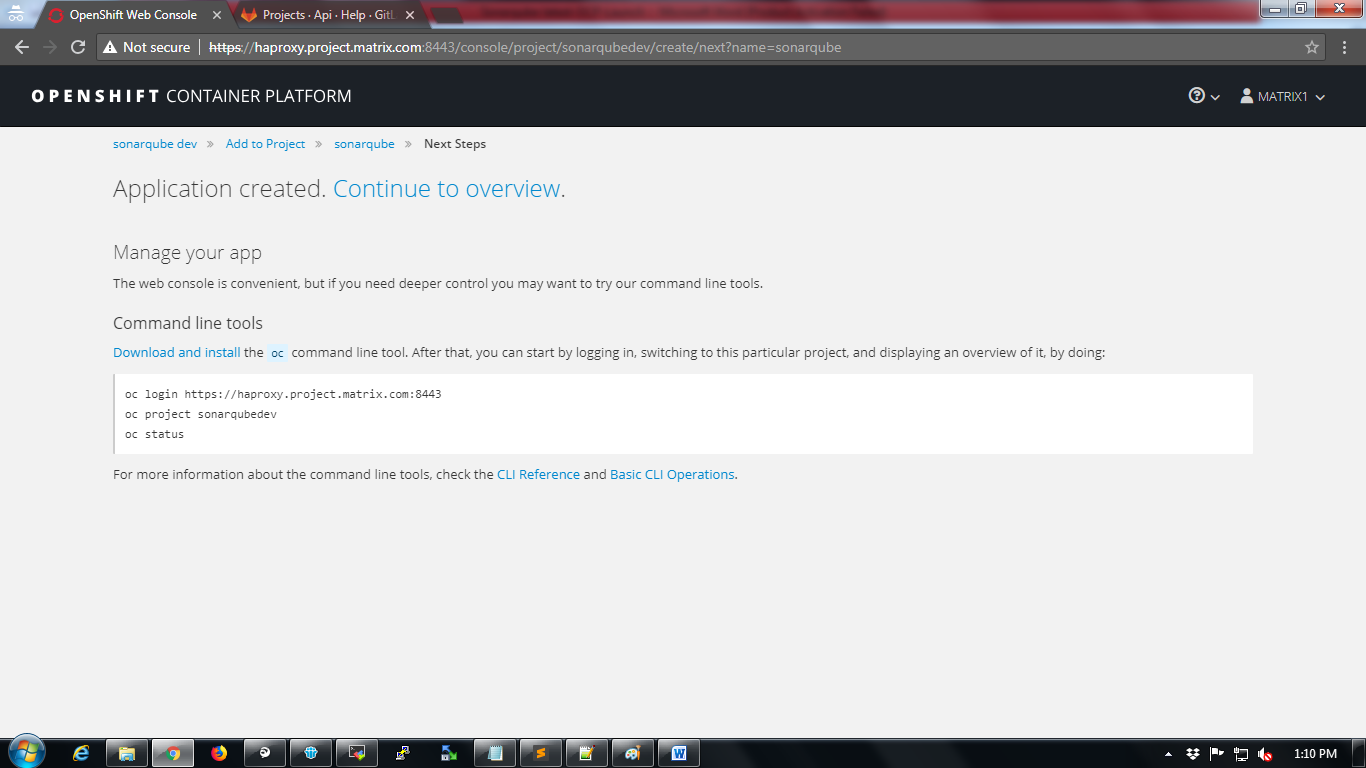


After clicking on search button sonarqube resources will be created as per the image as shown in the below screenshot.

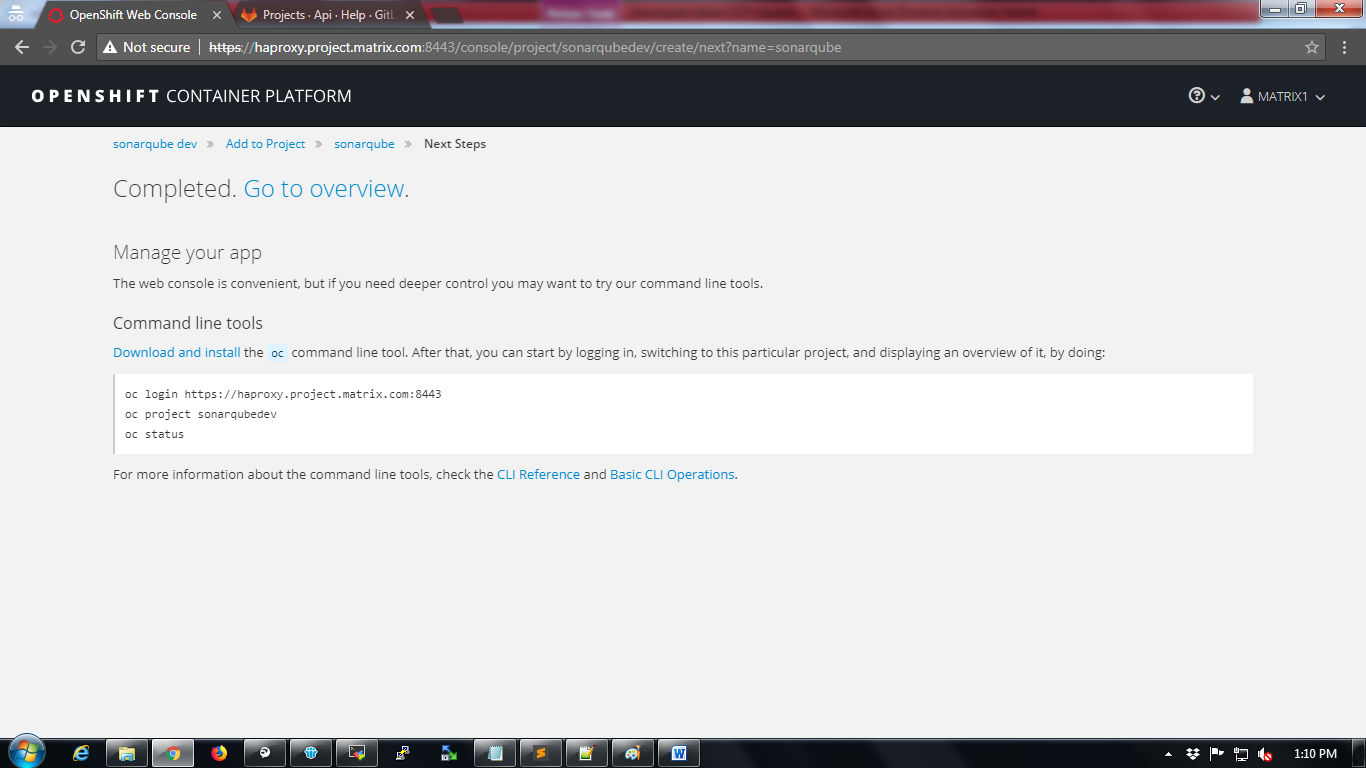
Click on “Create” button.



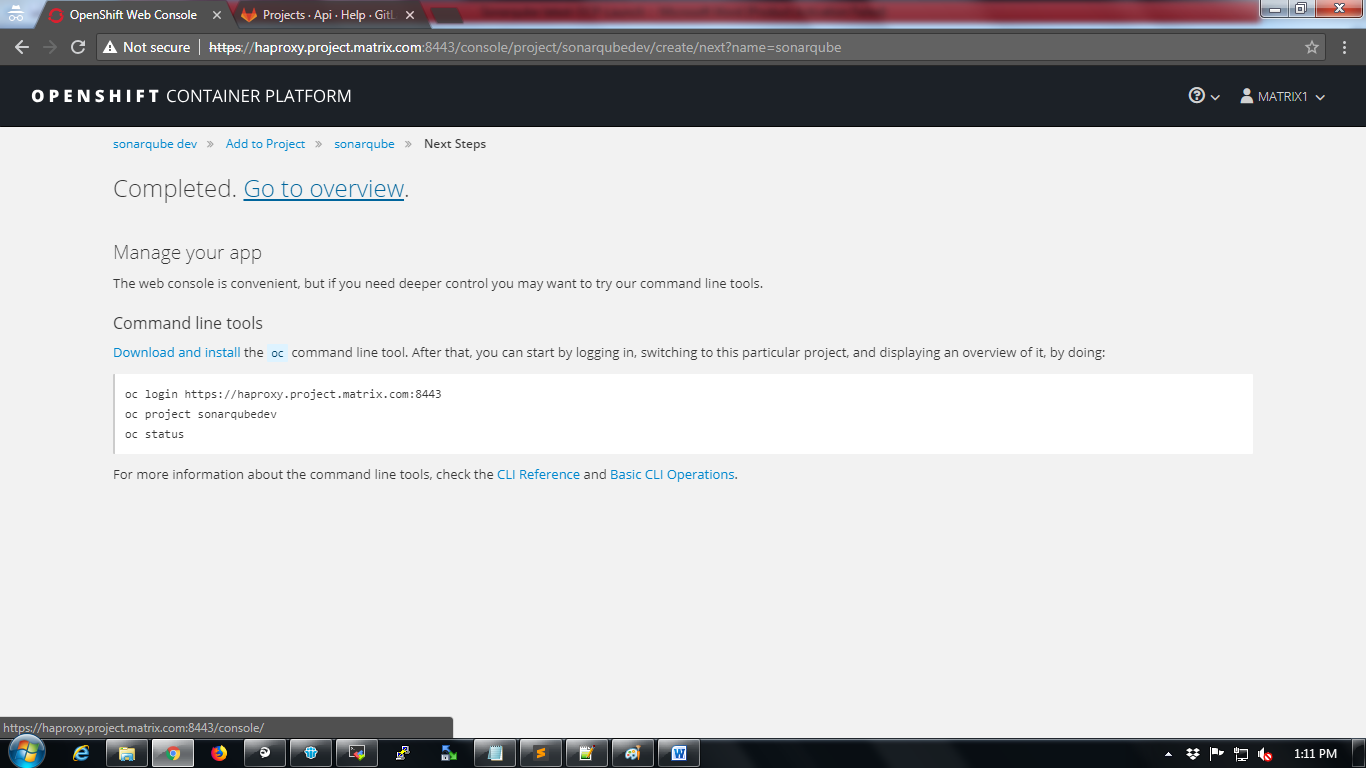
The next page will show status that “Application created”.

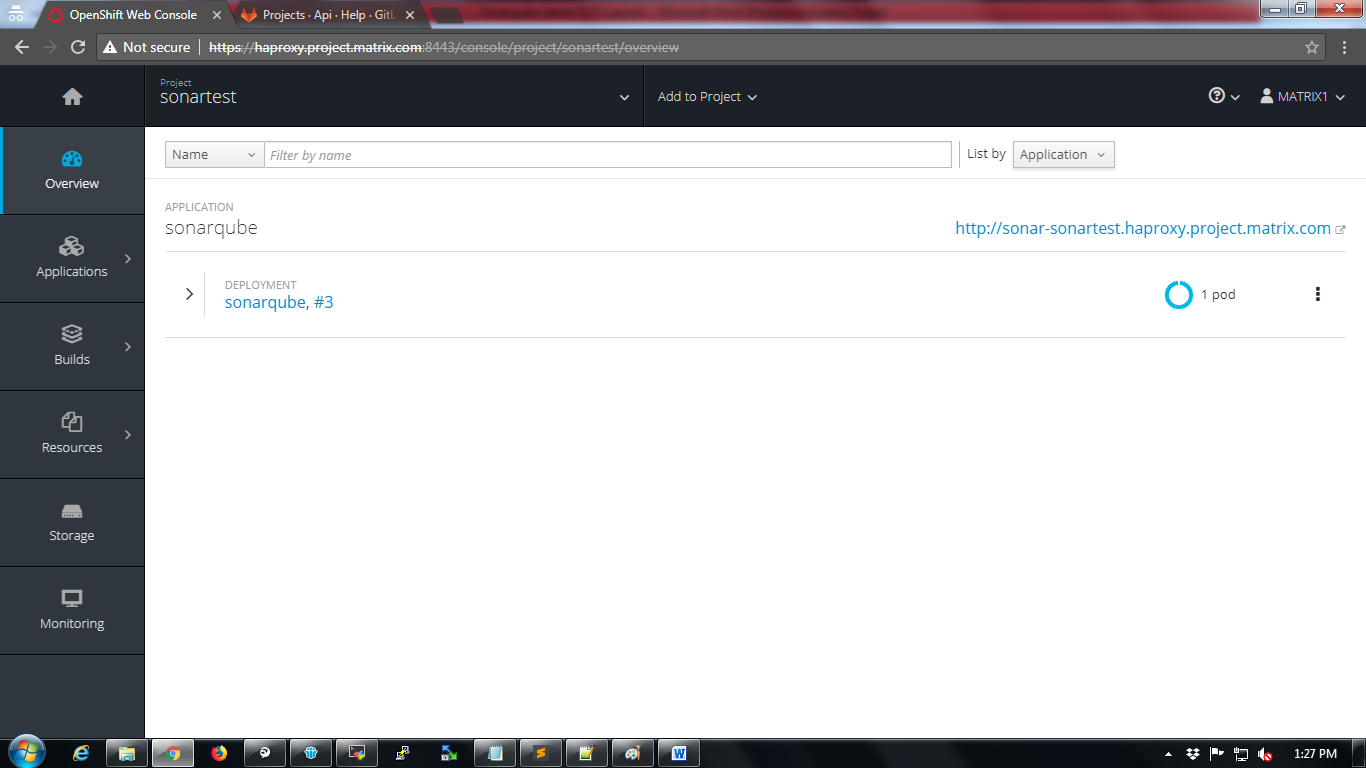


Wait for a while till the status shows “Completed” as shown below



Click on “Go to overview” tab to verify the project home.





**Adding Root permissions to OCP service**

OCP services may require root privileges for applications. The root privileges can be implemented using “userroot” service in OCP.

To add “useroot” privileges

1. Login into any master node VM of OCP
2. Select the project

# oc project sonartest

1. Create userroot service account

# oc create serviceaccountuseroot

1. Add useroot policy

# ocadm policy add-scc-to-user anyuid -z useroot

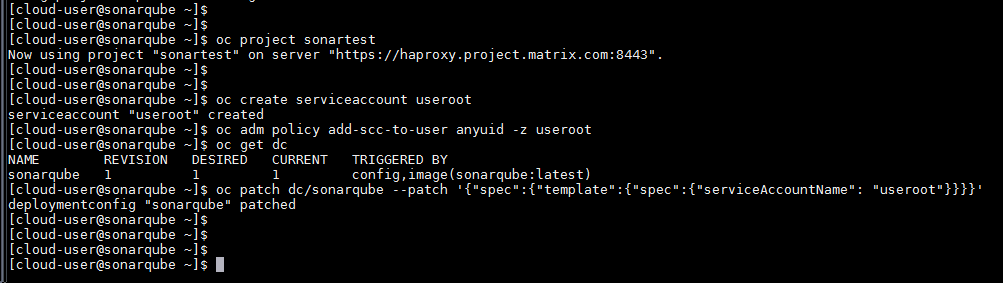
1. Get the dc name

# oc get dc

1. Patch the useroot account using dc name

# oc patch dc/<dcname> --patch '{"spec":{"template":{"spec":{"serviceAccountName": "useroot"}}}}'

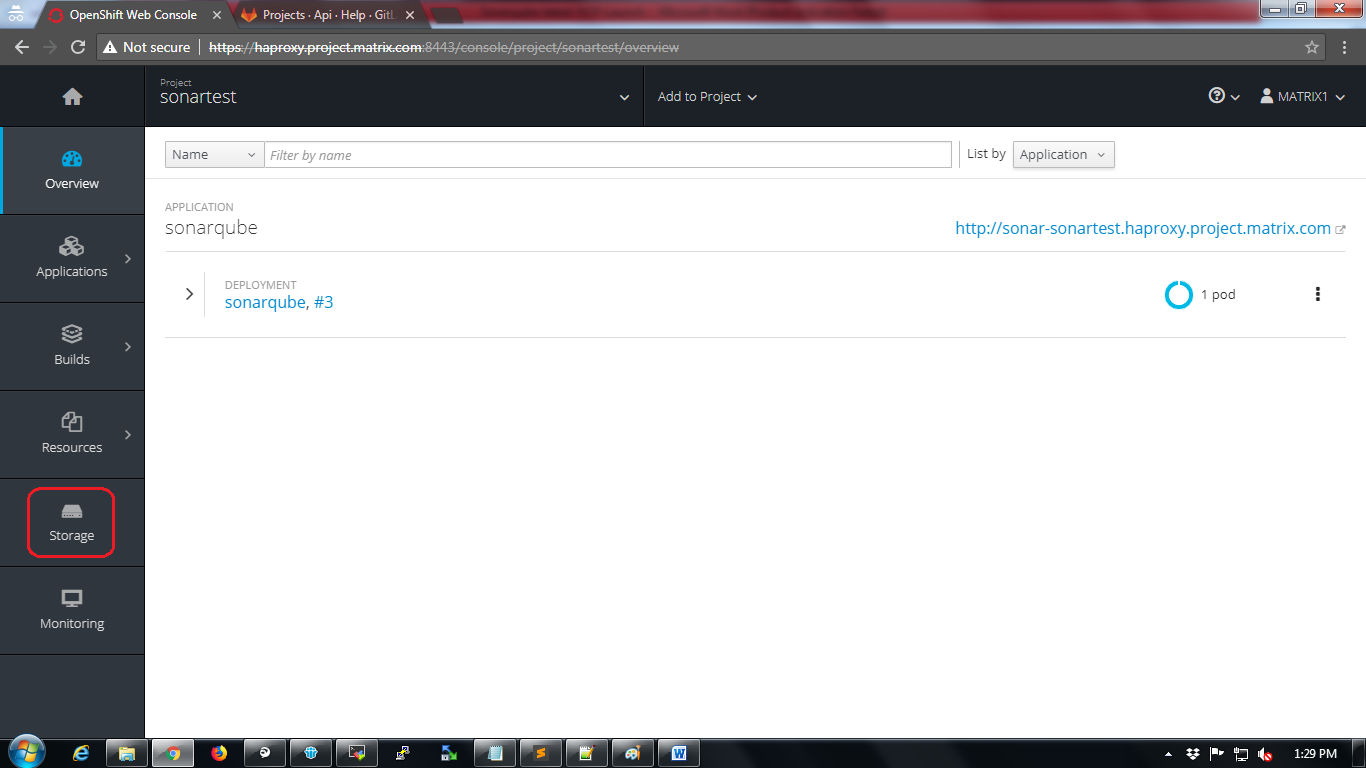
1. Refer screenshot below



**Adding persistent storage to sonarqube**

Persistent storage will store the application data even if pod in OCP restarts/crashes

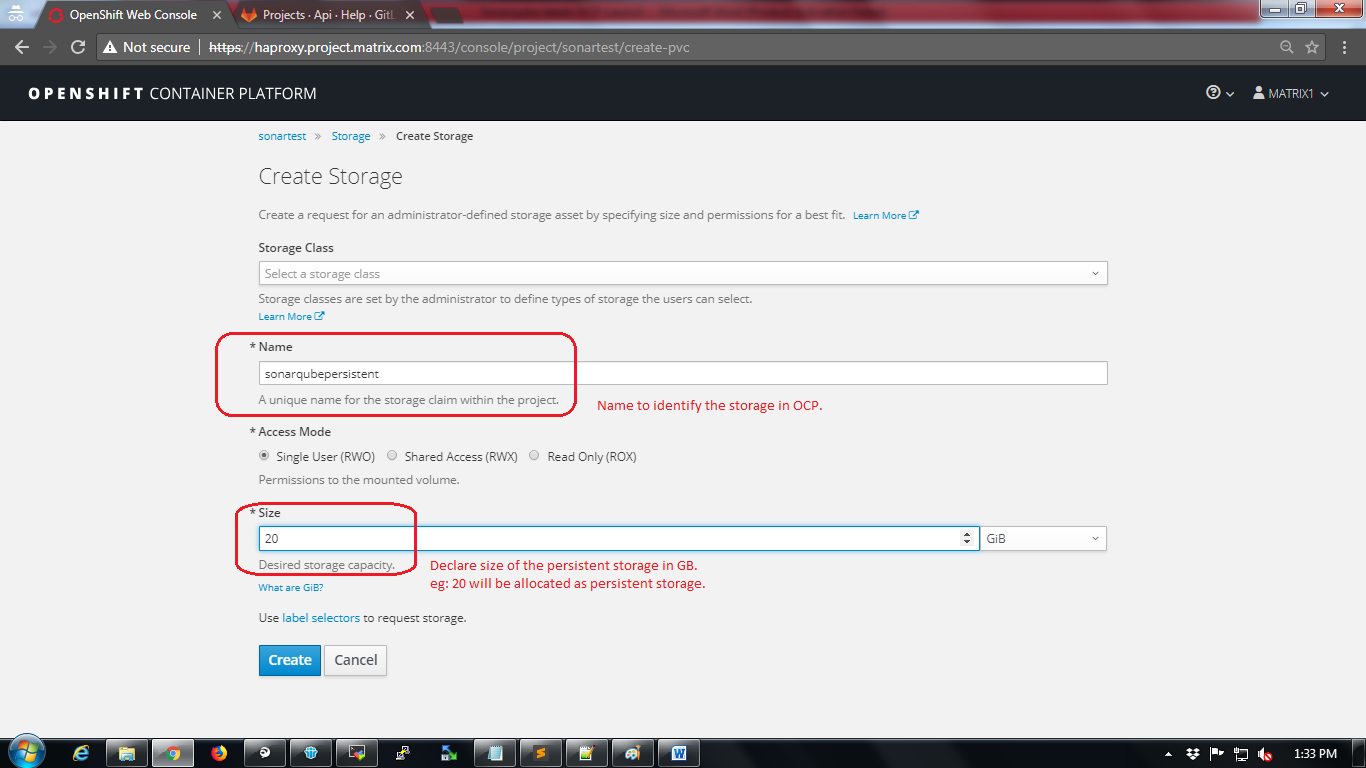
Click on “Storage” option to create persistent storage

****

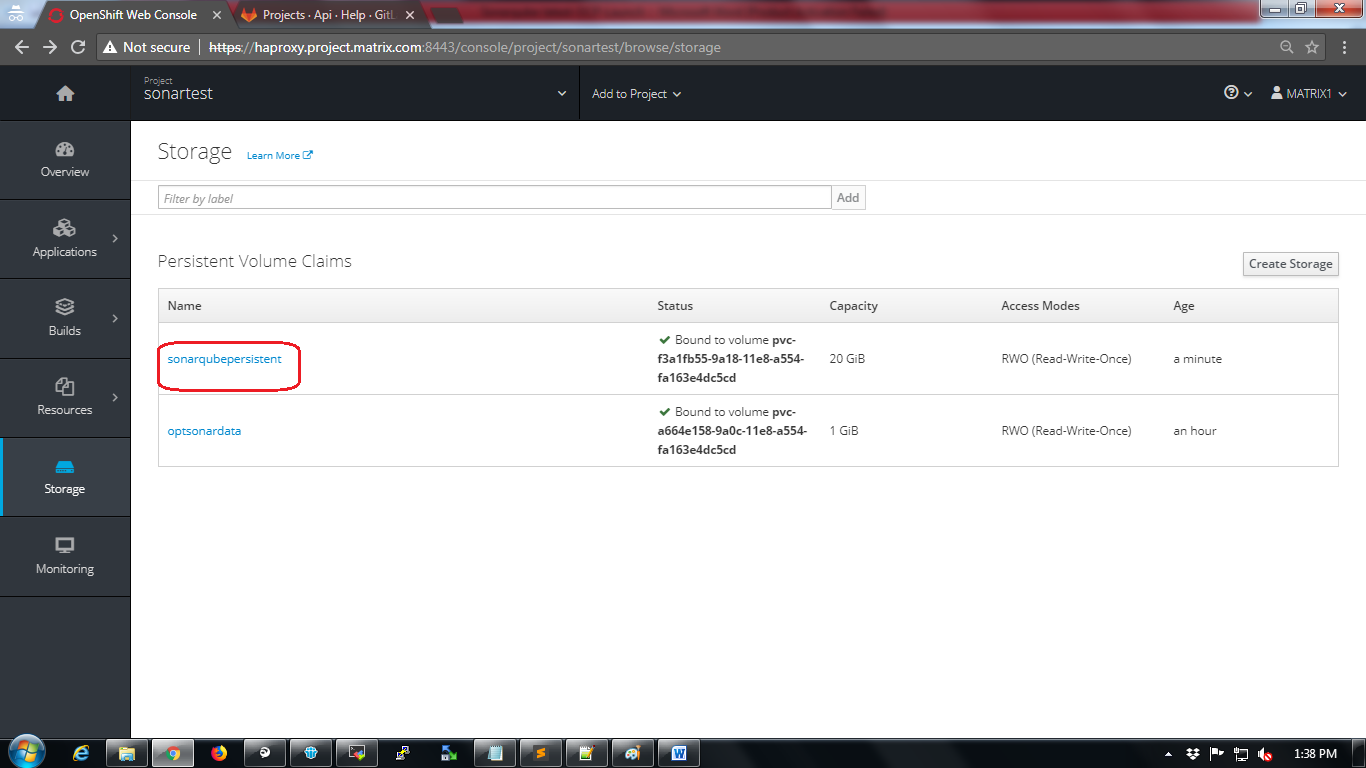
Click on “Create Storage” button.

Provide mandatory storage configuration detailsand click on “Create” button.

Refer screenshot below for sample configuration.

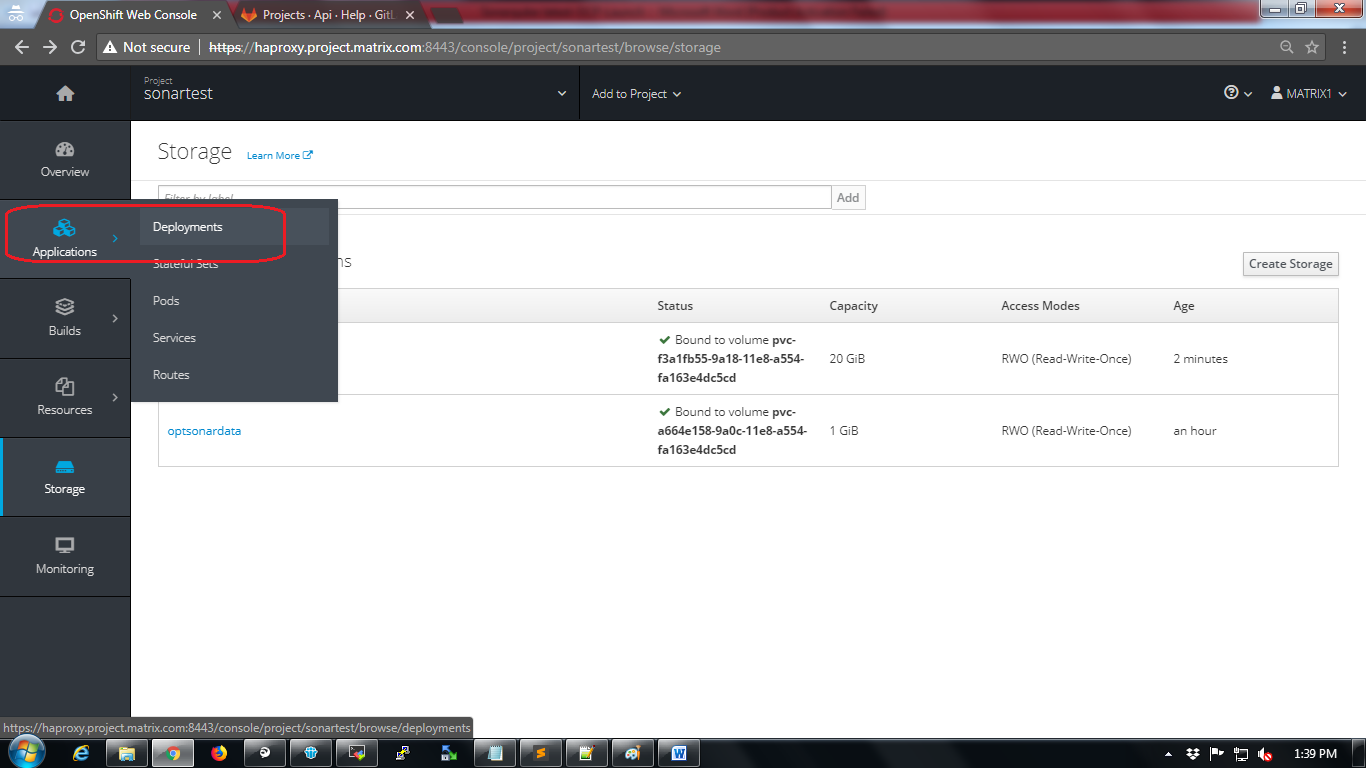


Storage will be displayed after creation.

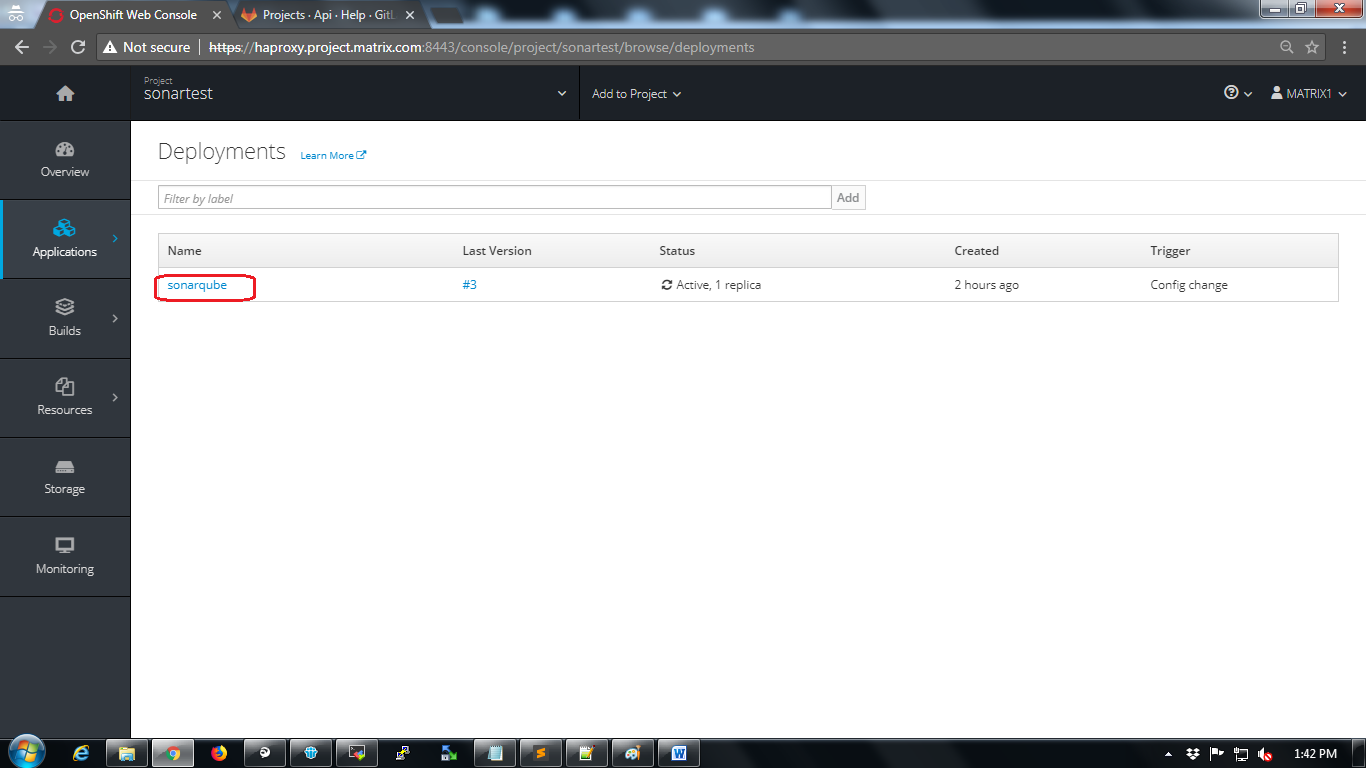


Now the storage should be added to the deployed image.

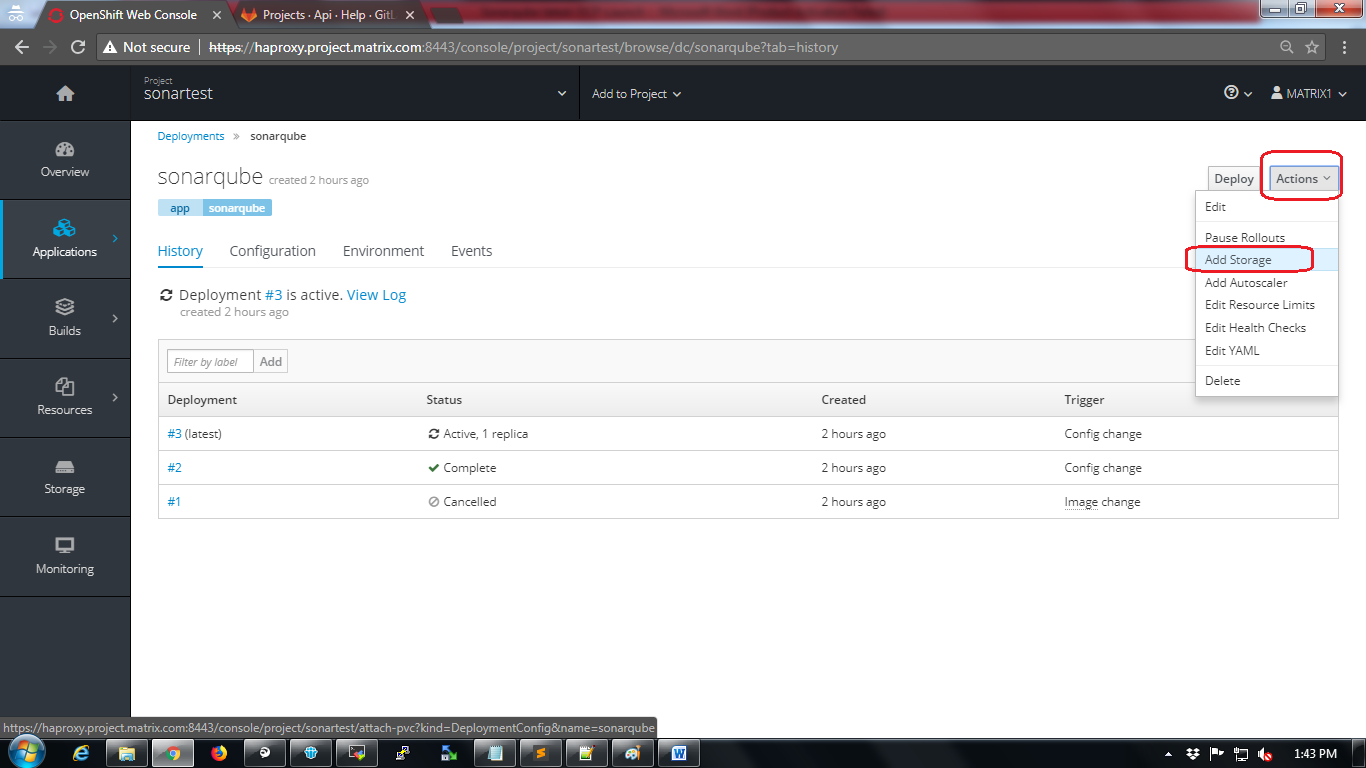
Click on Applications🡪 Deployments



Click on name of the deployment as shown in below screenshot



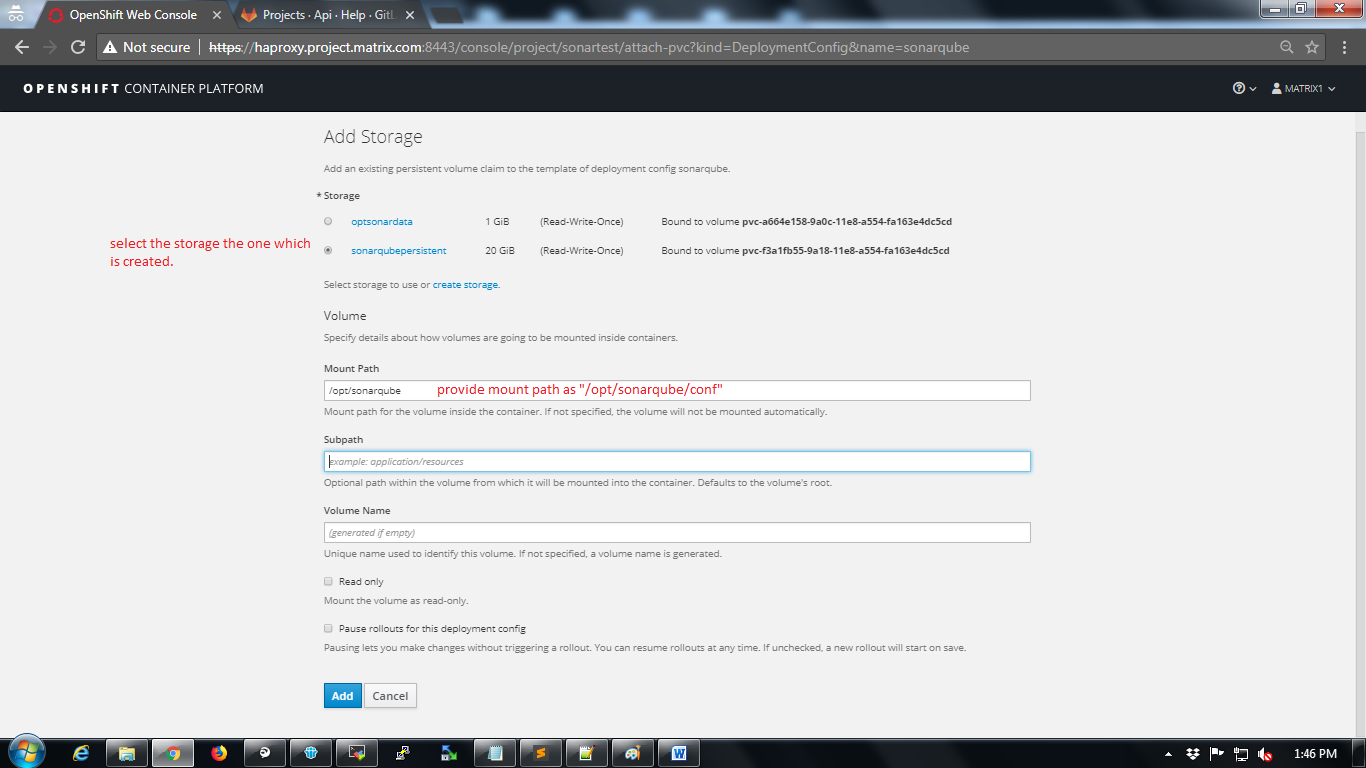
Click on “Actions” dropdown and select “Add Storage” option.



Select the storage the one which was created in previous step.

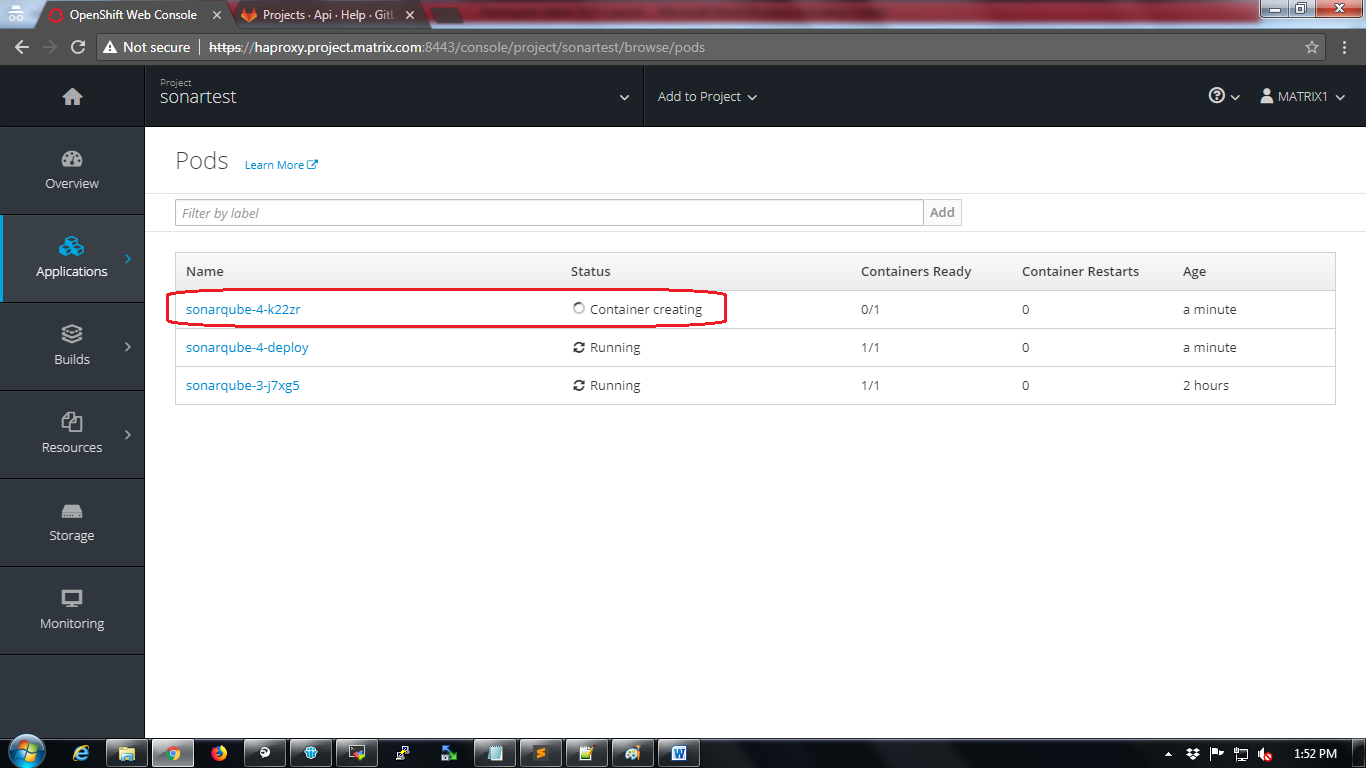
Provide mount path details and make sure to add “Mount Path” as “/opt/sonarqube/conf” as mentioned in the screenshot below.

Click on “Add” button.



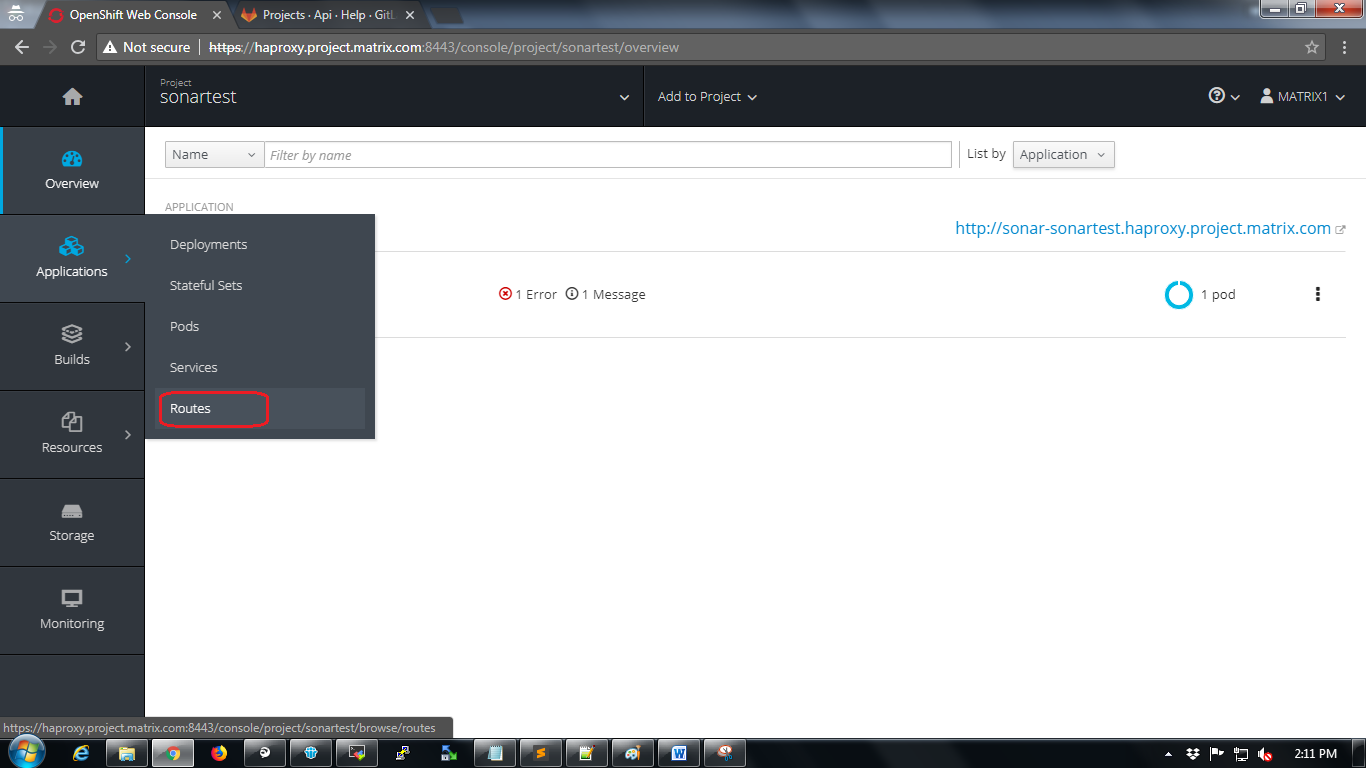
The moment storage is added the pod will restart and will be created with new configuration (persistent storage).

Wait for the pod to recreate

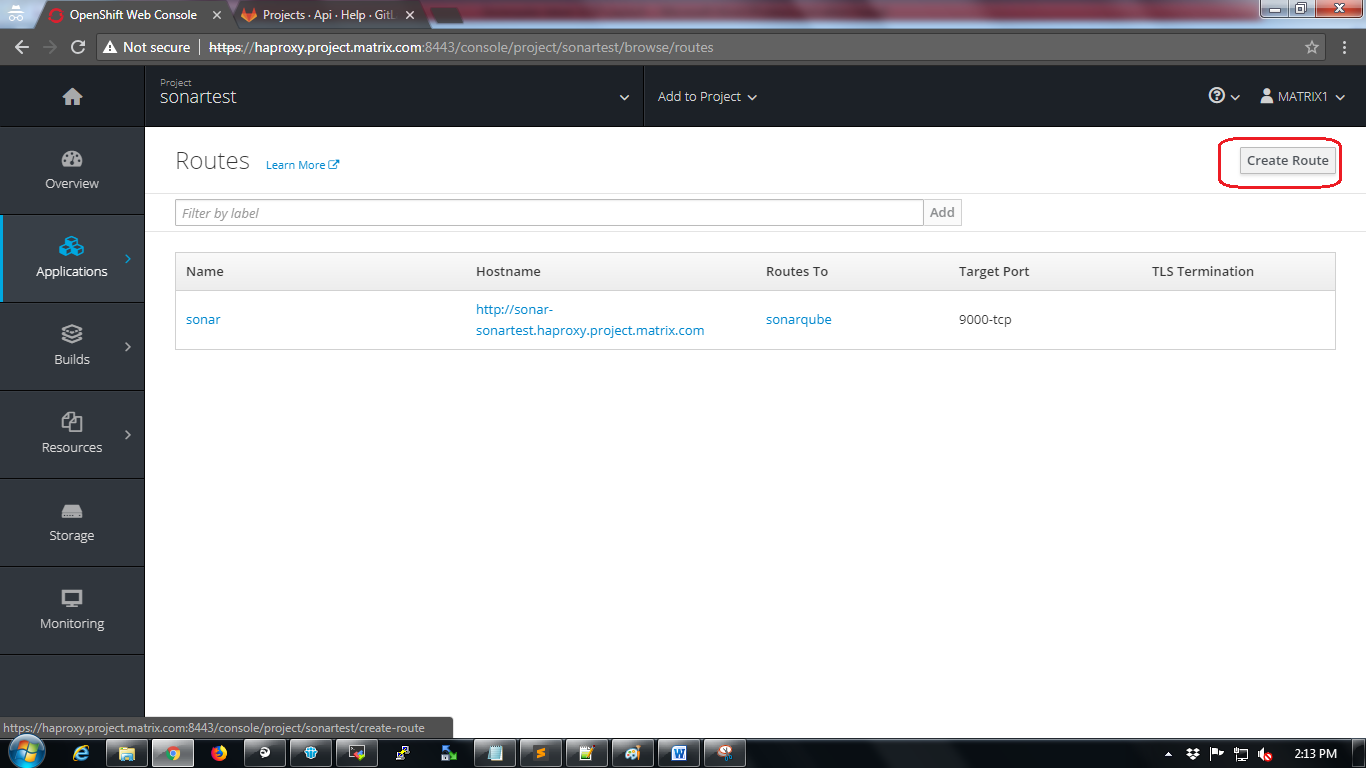


**Access the sonarqube UI**

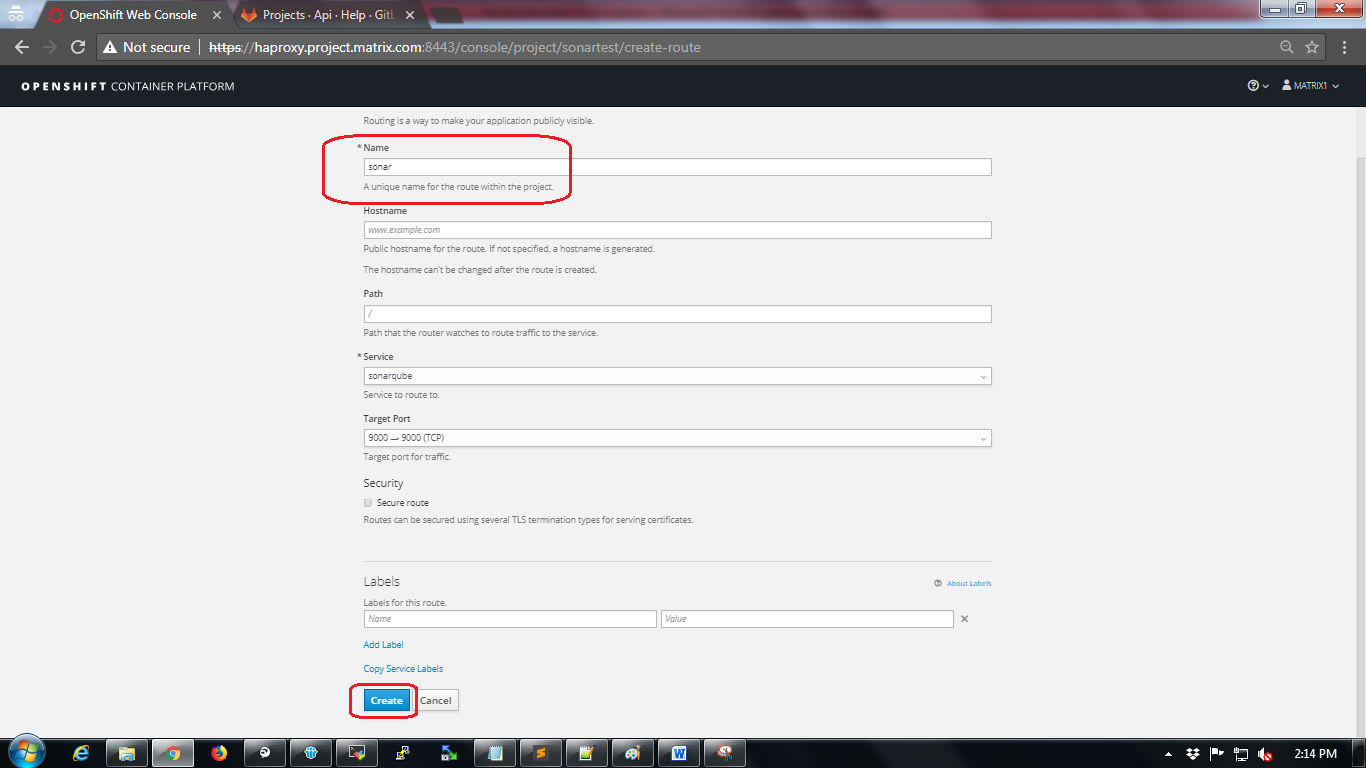
Click on Applications🡪Routes



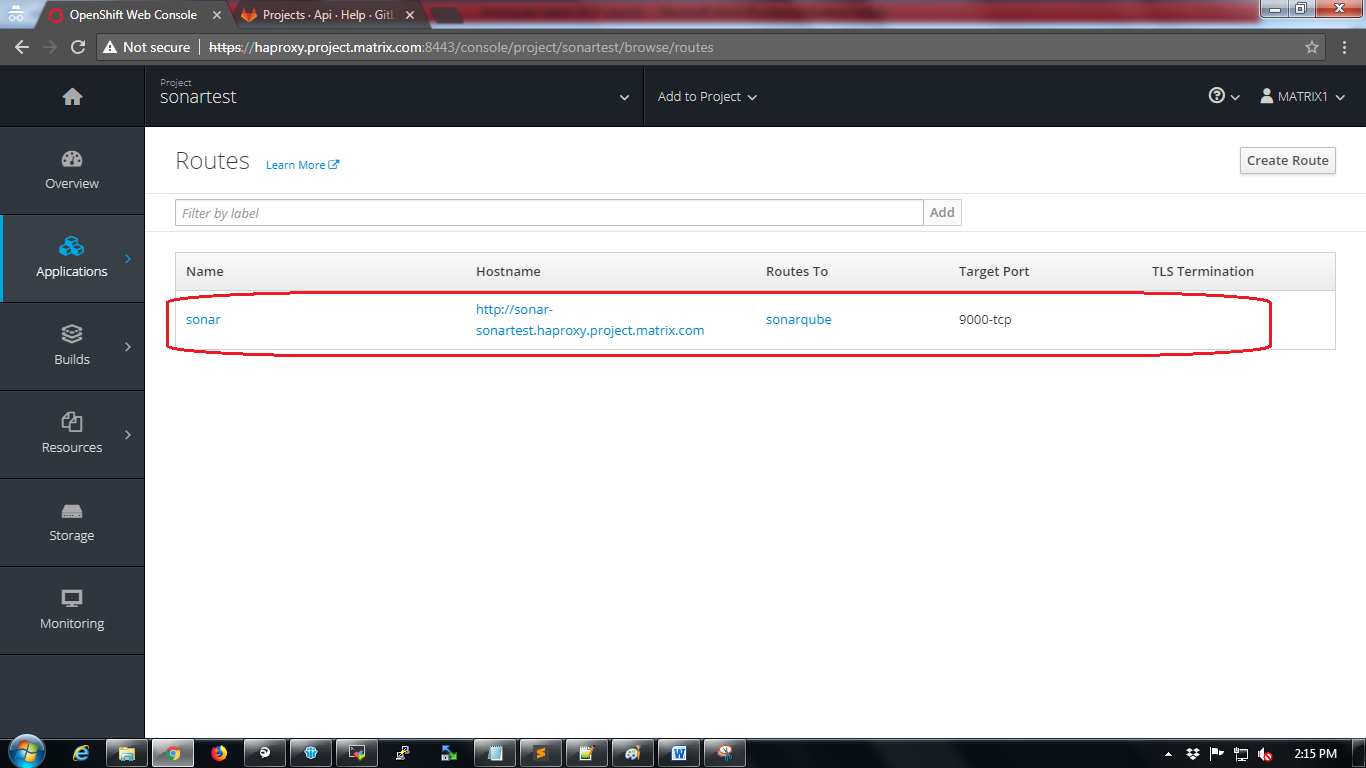
Click on “Create Route” button

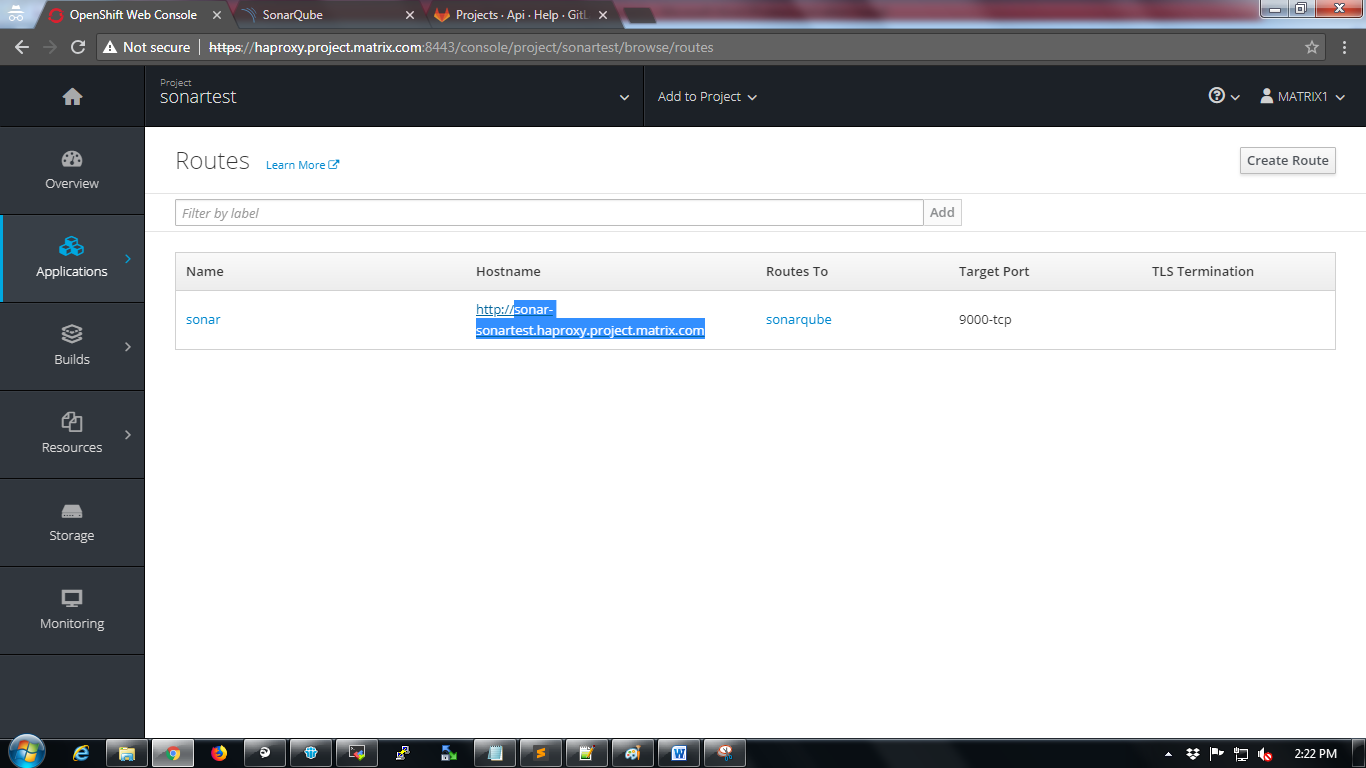


Provide name to the new route and click on “Create” button.



After route is created it will be displayed.

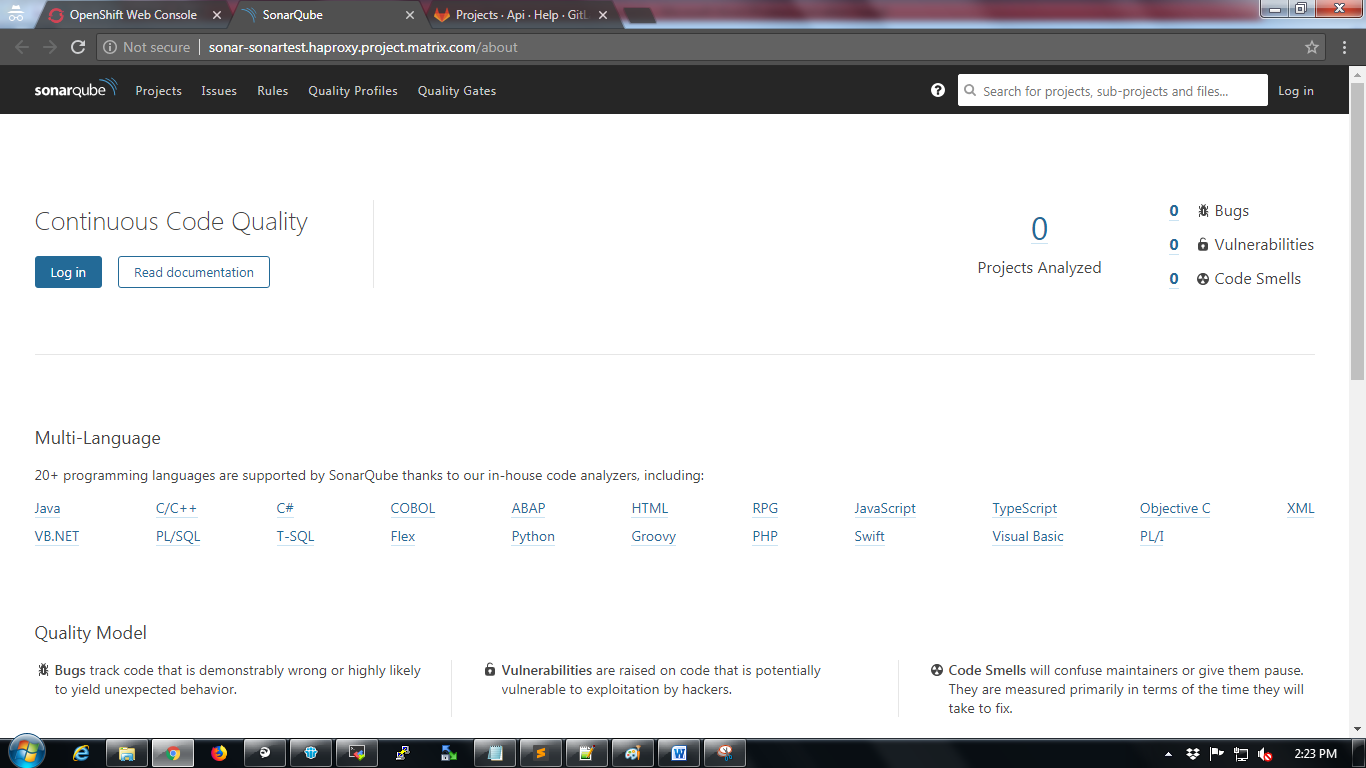




Now copy the hostname as highlighted in below screenshot and add to local computer’s hosts file to access using browser. While adding the hostname in hosts file make sure it is mapped to proper haproxy address of OCP.

Now try to access the application using the url.

Sonarqube UI will be shown as in the below screenshot.

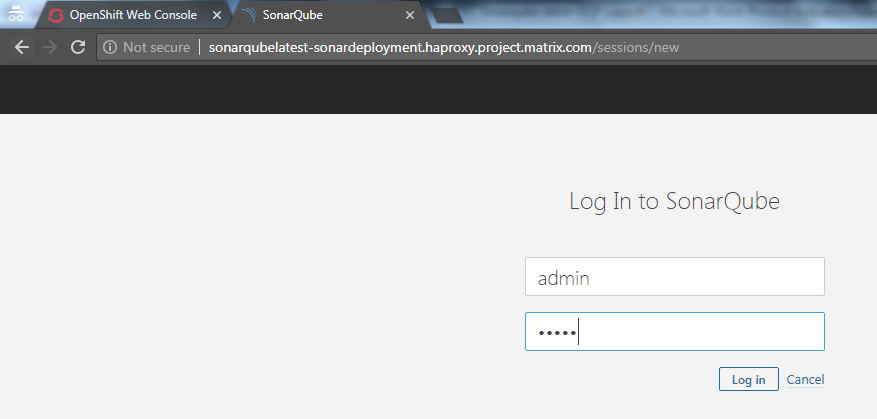


**Installing plugins in Sonarqube**

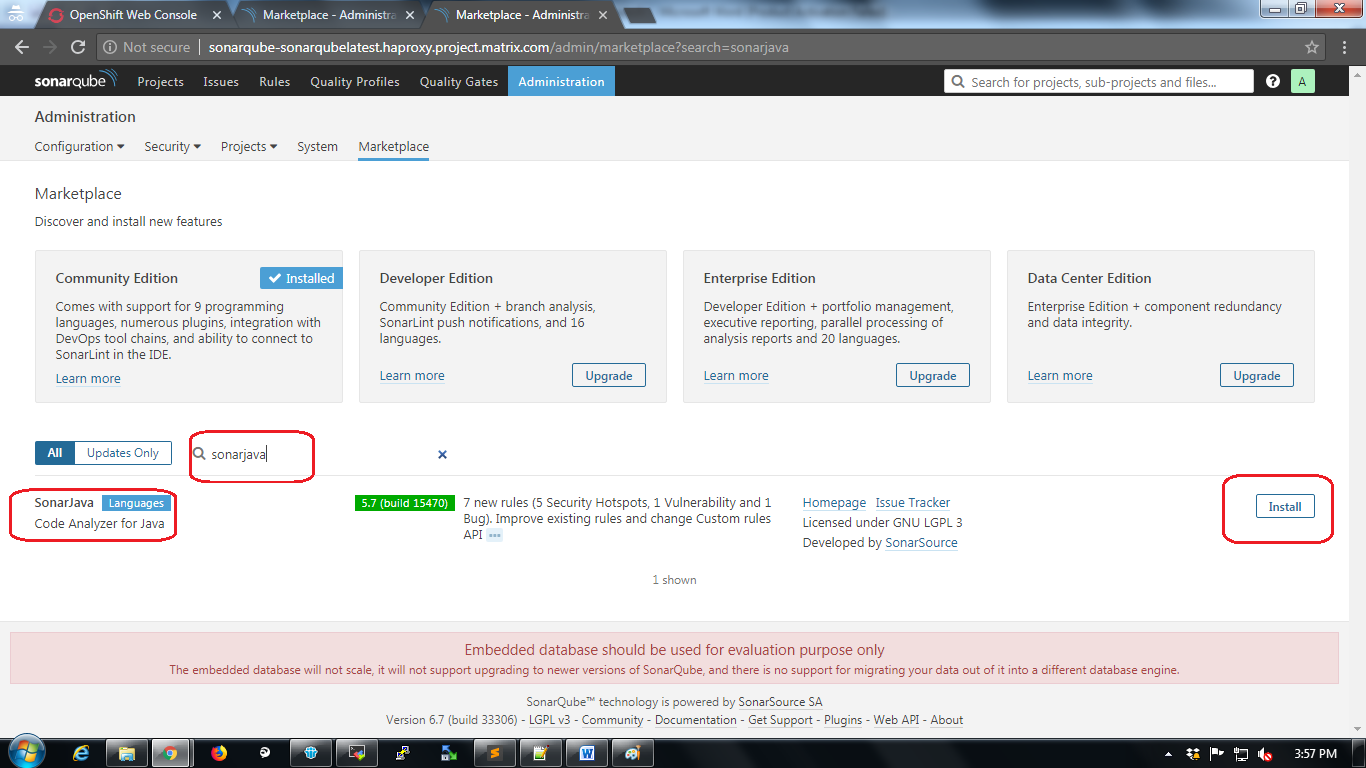
Login into sonarqube with default admin credentials

Username: admin

Password: admin



Navigate to “Administration🡪Marketplace”



Type “sonarjava” in search bar and click on “Install” button.

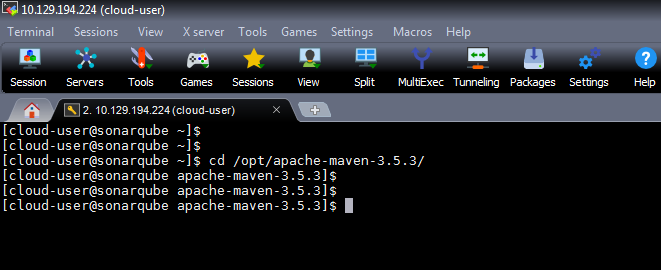
After installing the plugin Sonarqube will ask for restart. Restart Sonarqube.

**Configuring Sonarqube with Maven**

Sonarqube should be configured with maven for code analysis.

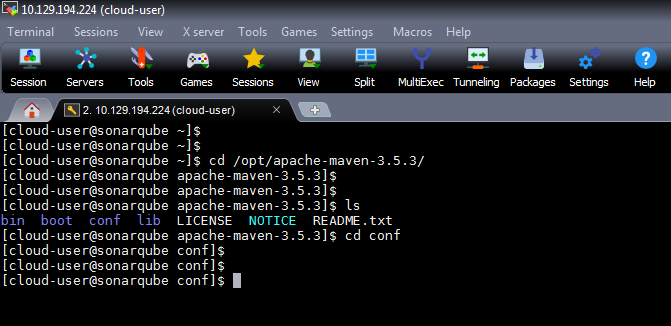
Change to maven home directory

$ cd /opt/apache-maven-3.5.3/



Change to “conf” directory

$ cdconf/



Open the “settings.xml” file and add following content in profile section

$ sudo vi settings.xml

*<profile>*

*<id>sonar</id>*

*<activation>*

*<activeByDefault>true</activeByDefault>*

*</activation>*

*<properties>*

*<!-- Optional URL to server. Default value is http://localhost:9000 -->*

*<sonar.host.url>*

[*http://sonarqube-sonarqubelatest.haproxy.project.matrix.com*](http://sonarqube-sonarqubelatest.haproxy.project.matrix.com)

*</sonar.host.url>*

*</properties>*

*</profile>*

**Note:**

***[Replace the sonarqubeurl mentioned in “sonar.host.url” block]***

*Add following line in “<pluginGroups>” section*

<pluginGroup>org.sonarsource.scanner.maven</pluginGroup>

Save the file.

**Note:**

By default sonarqube uses “Embedded H2” type database.