**Sonarqube details**

<http://3.115.218.163:9000/>

username:admin

password:sonar1234

**Sonar qube integration into aws code build for dotnet appilication**

Key points are

1. Create the Ec2 instance for sonar server
2. Install the sonar in Ec2 instance
3. Make sure that instance type should be t2.medium or larger as we need at least 3GB of RAM to run SonarQube efficiently.
4. Open the port number 9000 in security group
5. Create the sonarqube user in sonarserver

Ex sonarserverurl is :http://sonarserver instance ipaddress:9000

1. Take the existing docker file for our application
2. Install the sonar scanner in the docker container
3. Install the java in the docker container .Because sonar scanner need for java
4. Connect to sonar server using sonarscanner begin command within the docker container. It should have sonar token id , sonar server url , sonar project key for every project and should be unique .

I:go to sonar server

ii:go to project option

iii:create project

iv:generate token

1. update the code quality results in the sonarserver
2. And also you need to update the projectguid in csproj file .

**1.Install the sonar server in ec2 instance**

1.create ec2 instance and name is sonarqube server

2. connect to ec2 instance and install the sonar server

To start, get a SonarQube server instance up and running

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

1. After it’s installed, verify you’re using this version of Java:

$ sudo alternatives --config java

1. Choose the Java 11 version you just installed.

Java -version

5.install the sonarqube

wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-6.4.zip

6.install unzip by running:

sudo yum -y install unzip

7.Unzip the archive using the following command.

sudo unzip sonarqube-6.4.zip -d /opt

8.Rename the directory:

sudo mv /opt/sonarqube-6.4 /opt/sonarqube

9.start the sonar server

/opt/sonarqube-6.4/bin/linux-x86-64/sonar.sh start

10.edit the inbound rules in security group with port number 9000

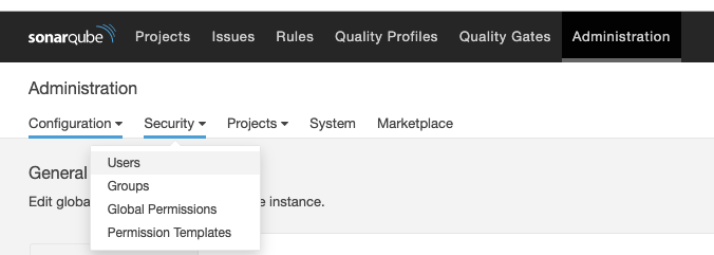
**2:Open** [**http://sonarqube**](http://sonarqube) **instance ipaddress:9000 in browser**

Login with default sonarqube credentials is username and password is admin

If you want create the user in sonarqube follow the below steps

Get started by creating a SonarQube user from your SonarQube webpage. This user is the identity used by the robot caller to your SonarQube for this workflow.

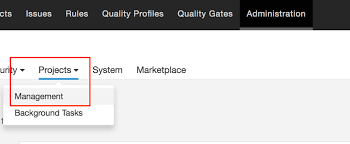
1. Go to the Administration tab on your SonarQube instance.
2. Choose Security, then Users, as shown in the following screenshot.

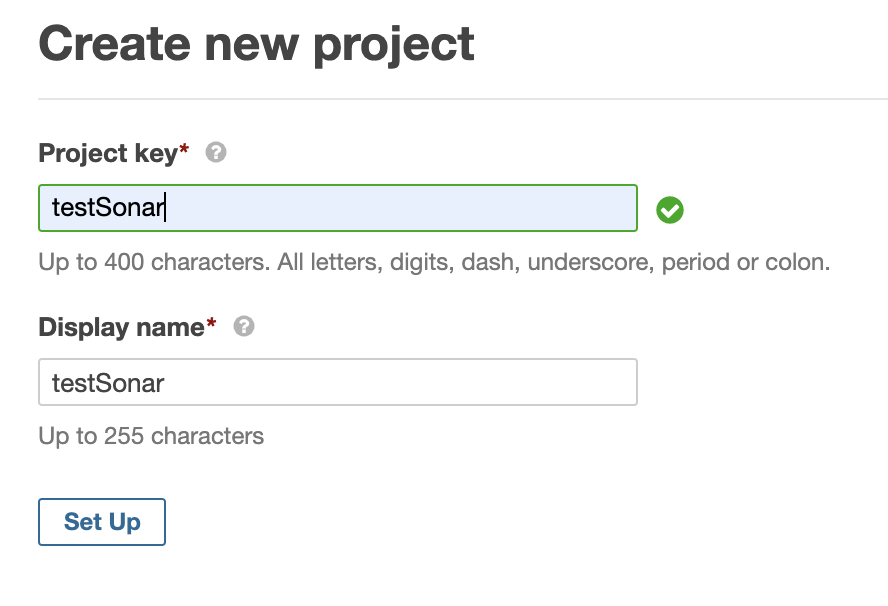


1. Choose Create User. Fill in the form, and note down the Login and Password
2. Choose Create.

Every project You need to create unique token and project key that details will specify in the sonarscanner begin command . follow the below steps

Go to projects under administration tab





Click setup

And generate token key in the testsonar project

Copy the token key into your nodepad

**3:Install the sonarscanner and call the sonarserver into ur docker container using docker file and it will pass the code quality results in ur sonar server**

Update the sonar commands in the existing docker file

FROM mcr.microsoft.com/dotnet/core/aspnet:3.0

WORKDIR /app

EXPOSE 80

EXPOSE 443

ENV SONAR\_SCANNER\_OPTS="-Xmx512m"

FROM mcr.microsoft.com/dotnet/core/sdk:3.0

#build

RUN apt-get update -y

RUN apt-get install default-jdk -y

WORKDIR /src

COPY . .

COPY ["testsonar.sln", "testSonar/"]

COPY ["testsonar.csproj", "testSonar/"]

COPY ["testsonar.xml", "testSonar/"]

COPY ["NuGet.Config", "testSonar/"]

COPY ["SharedSettings.json", "testSonar/"]

RUN dotnet restore "testSonar"

COPY . "testSonar/"

RUN dotnet tool install --global dotnet-sonarscanner

ENV PATH="${PATH}:/root/.dotnet/tools"

WORKDIR "/src/testSonar"

RUN dotnet sonarscanner begin /k:"testSonar" /d:sonar.host.url=sonarqubeinstamceipaddress /d:sonar.login=f9a140acacc58e6f5a9aa966065bab21b105b9da(provide sonar token here)

RUN dotnet build "testsonar.csproj" -o /app

RUN dotnet sonarscanner end /d:sonar.login=f9a140acacc58e6f5a9aa966065bab21b105b9da

FROM build AS publish

RUN dotnet publish "testsonar.csproj" -o /app

FROM base AS final

WORKDIR /app

COPY --from=publish /app .

ENTRYPOINT ["dotnet", "testsonar.dll"]

**Sonarscanner:** The SonarScanner for MSBuild is the recommended way to launch an analysis for projects/solutions using MSBuild or dotnet command as a build tool. It is the result of a [collaboration between SonarSource and Microsoft](http://www.sonarqube.org/announcing-sonarqube-integration-with-msbuild-and-team-build/).

**/k:<project-key>:[**required] Specifies the key of the analyzed project in SonarQube

**/d:sonar.login**=<username> or <token>:[optional] Specifies the username or access token to authenticate with to SonarQube. If this argument is added to the begin step, it must also be added on the end step.

**dotnet sonarscanner begin command:Begin**. The **begin** step is executed when you add the **begin command** line argument. It hooks into the MSBuild pipeline, downloads SonarQube quality profiles and settings and prepares your project for the analysis.

### Dotnet sonarscanner end command:

The end step is executed when you add the "end" command line argument. It cleans the MSBuild hooks, collects the analysis data generated by the build, the test results, the code coverage and then uploads everything to SonarQube.

**4.update projectguid in csproj file**

The project GUID in the solution is a unique identifier of a project across the solution inside VS. In non-SDK-based projects, it's a required element that must be present in the project. And anaylsis the results correctly in sonarqube, so its required for sonarqube and every project should be unique projectguid

To generate the projectguild

Go to <https://www.guidgenerator.com/> this link and generate projectguid into ur notedpa

And paste projectguid number in the below projectguid syntax

**<PropertyGroup>**

**<!-- other properties here -->**

**<!-- SonarQube needs this -->**

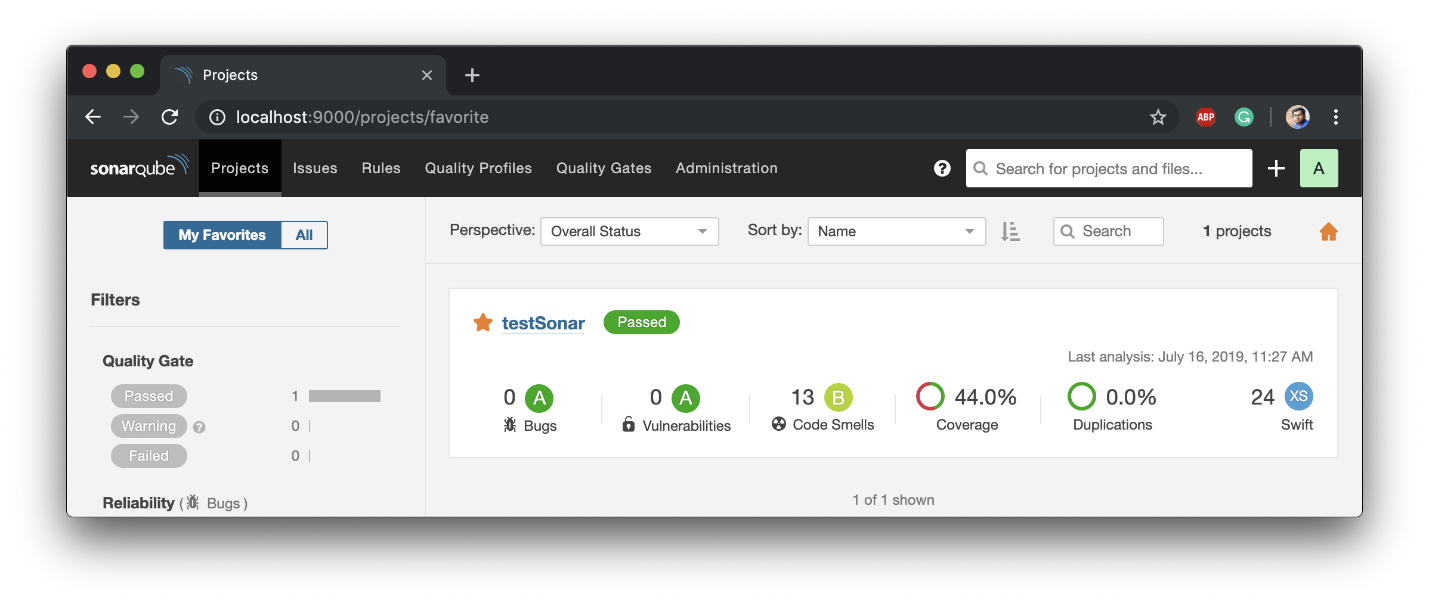
**<ProjectGuid>{d92d84e4-5d27-42a1-ac50-3d9d93351915}</ProjectGuid>**

**</PropertyGroup>**

And update the propertygroup in csproj in the root source folder

Then trigger the build pipeline and it will automatically the sonar results will be happened in the

sonarqube server

5**. open the sonarqubeinstanceipaddress with port number 9000**

**6.Sonarqubeserver details:**

**Serverurl**:<http://3.115.218.163:9000/> ( ipaddresss is the sonarqube instance ip addresss)

Username:admin

Password:sonar1234