

TDDE46: Software Quality

Lab 5

METRICS VISUALIZATION

Objectives:

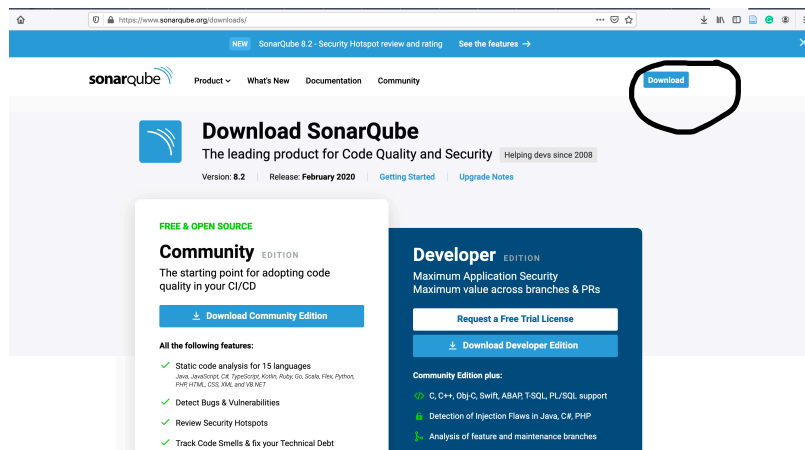
1. The main objective of this lab to let student familiar with the need for visualization and what tools are available for visualization
2. To understand the problem scenario to achieve metrics visualization
3. To make comparison between different types of metrics visualization
4. To work with SonarQube and Grafana and Prometheus for metric visualization

Lab Learning Outcomes (Reference to learning outcomes on course page):

1. Analyze the most common quality factors of software (L1)
2. Apply tools to improve product quality (L3)
3. Assess the product quality of a programming project (L7)
4. Integrate quality evaluations in an ongoing programming project (L9)

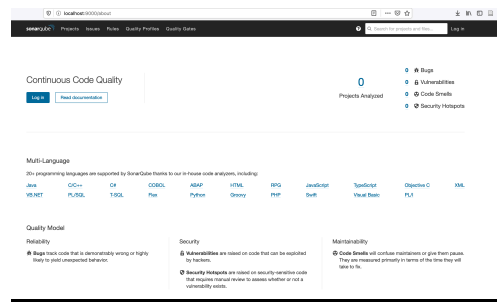
Prerequisites (Part 1):

1. Download the SonarQube from <https://www.sonarqube.org/downloads/>. It will be a zip file



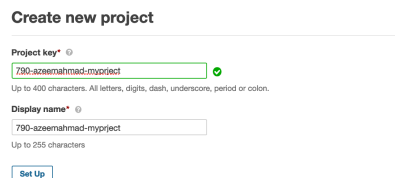
2. Unzip it to any directory, i.e., Dekstop
3. Go to Bin folder. You will select the folder based on the operating system, you have installed.
4. Go to folder according to your operating system (i.e., linux-x86-64) in the unzipped directory in terminal.
5. Run the command: `./sonar.sh start ; xdg-open http://localhost:9000`

The SonarQube is running. It may take some time to appear in browser. Wait for it and refresh the page. You will see the following screen.

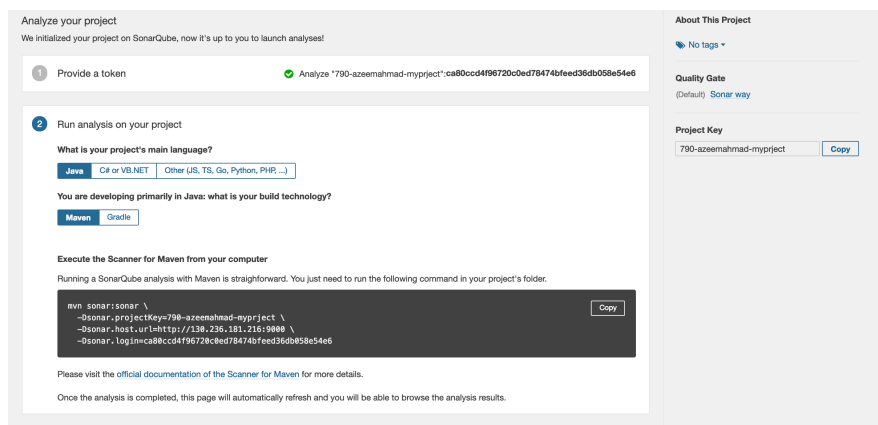


Your job is to do the following:

1. Click on Login:
 - a. UserName: admin
 - b. Password: admin
2. On the top right, click the “+” button and create on “Create New Project”
3. You will see the following screen. Enter project key such as “YourName-RegistrationNumber-AnyProjectName” and click “Set Up”



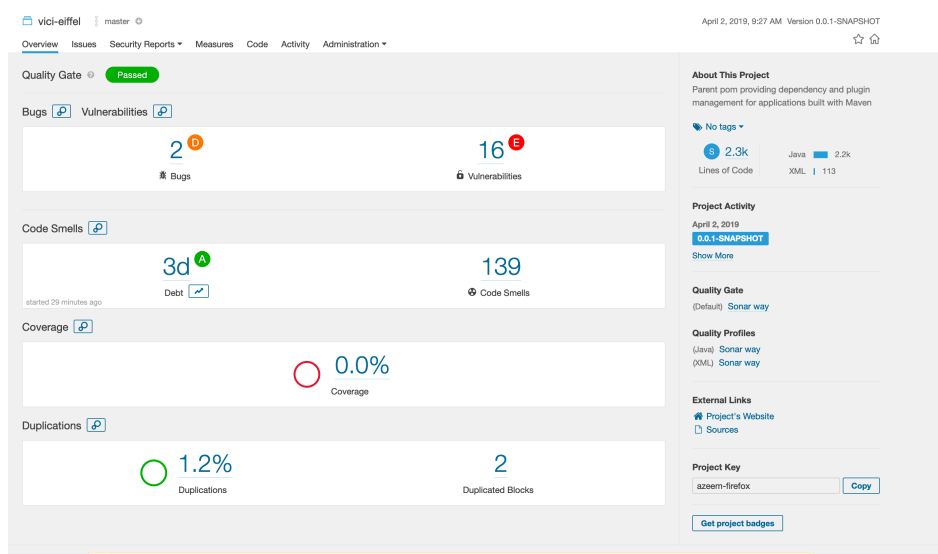
4. Write any text and Click “Generate” to create token and copy the token to a place for future use. Click on “Continue”
5. Now, it will ask you which language, you want to use for analysis as well as either ‘Maven’ or ‘Gradle’ as per below image. I have selected ‘Java’ and ‘Maven’. After the selection, the GUI will give you a number of commands to run as shown in black area in the following diagram. Copy these commands.



Download/Clone any project (e.g., source code) /from GitHub which was built with maven. You can download any project from online or you can use any project you ever built earlier with Maven. **NOTE: Since you have selected ‘Java’ and ‘Maven’ earlier, your project must fulfill the above conditions. This might be a challenge to select a maven project**

that has issues such as code smells, vulnerabilities, etc. You may use the following project if you want.

- a. <https://github.com/eiffel-community/eiffel-vici>
 - i. Go to cloned folder and first run the command in terminal: `mvn spring-boot:run`
 - ii. When you see in terminal that application is started and running, then open a new terminal and go to “Eiffel-vici” folder which you just cloned and copy the instructions (i.e., from step 5 above), you received from SonarQube.
6. After the build is successful, you will see the URL (<http://localhost:9000>) refreshed automatically and providing you information as shown in the following image (The image will be changes based on the input project). if the page does not refreshed automatically, try clicking on your project.



6. Once the static code analysis is done. Your job is to explore the following output/metrics on the webpages:
 - a. Bugs
 - b. Vulnerabilities
 - c. Debt.
 - d. Code Smells
 - e. Coverage
 - f. Duplications
 - g. Duplicated Blocks
 - h. Click on Security Report Menu
 - i. OWASP 10
 - ii. SANS Top 25
 - i. Click on Measure Menu
 - i. Reliability
 - ii. Security
 - iii. Maintainability
 - iv. Coverage
 - v. Size
 - vi. Complexity
 - j. Click on Code Menu

You are answering the following questions:

- A. What does the above factors (i.e., bullets) represent?
- B. How can you use this information to determine software quality? Explain only five factors.
- C. How important is this information to determine software quality?
- D. Is the information above, enough to decide if the product has a good quality?
- E. What is the most and less important information, you have seen, on this dashboard?
- F. Is there any information missing on the dashboard, that you believe, should be there to improve software quality?

Part 2:

Problem Scenario:

You have been hired as a web quality assurance engineer at 'Svenska Spel'. This is a soccer match season. There are two matches per day for whole month. As we are all aware, people place their bid on matches using website even during last minute. Your job is to keep an eye on the status of website. If you suspect anything, you should direct your queries to dedicated IT team who will address your queries. Last year, the website was down for half minute on different occasion and lost much revenue.

Exercise 2.1:

Draw a sketch of the visualization/dash board, on paper, that you believe should be beneficial for you to do your job. Consider the following:

- What information you need to keep an eye to stay up-to-date?
- What information can help you in achieving good quality?
- What metrics should be available in your dash board. Show the sketch to your instructor/ or upload on LISAM as per instruction during the lab.

Exercise 2.2:

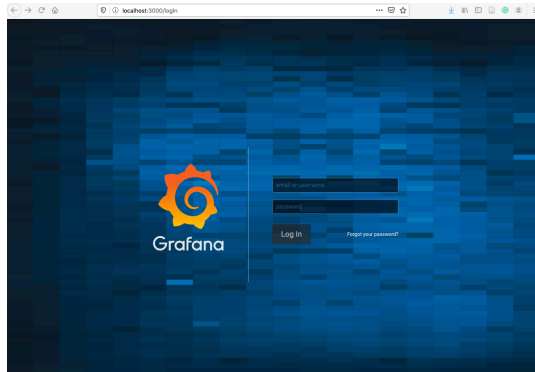
You decided to use some online dashboards to visualize metrics for your website. Write any two dashboard or visualization tool, available, in market that can help you solve your problem and compare these with Grafana tool which is a next exercise.

Exercise 2.3 (Running Grafana Instance Using Docker):

Grafana, based on different OS, can be installed using the below link:

<https://grafana.com/docs/grafana/latest/installation/>

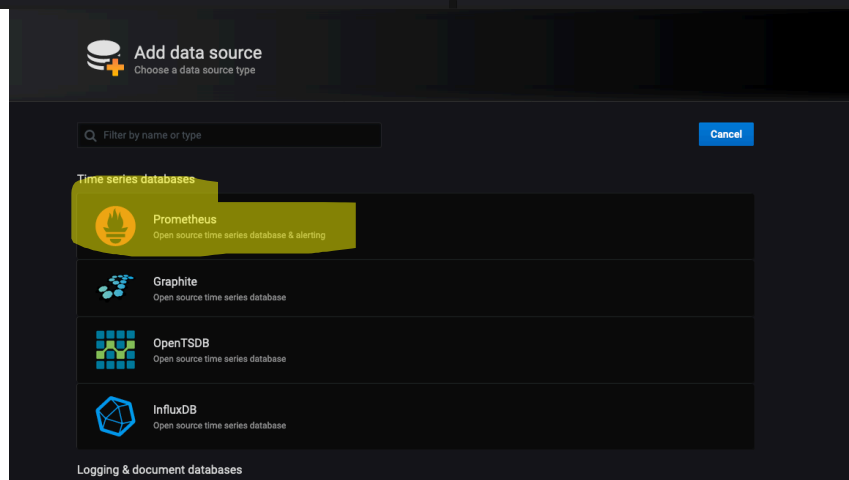
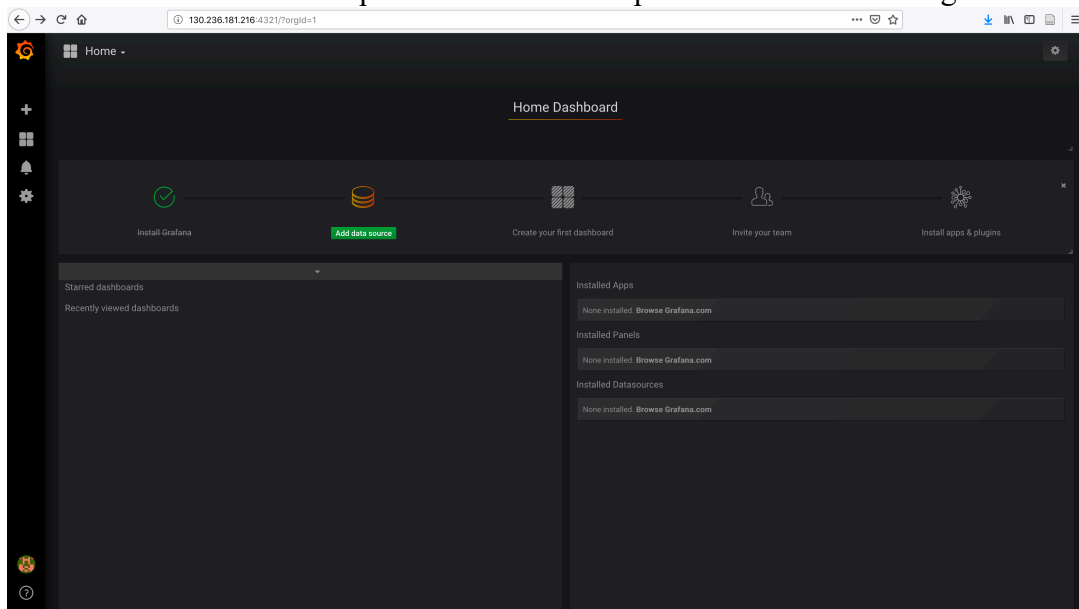
NOTE: You need to install it and run the Grafana server. If you need help in installation, either connect with me through Skype: Azeem_ahmad59 or send an email or we can do it during the lab time. Once the Grafana service is started, you can access it using 127.0.0.1:3000



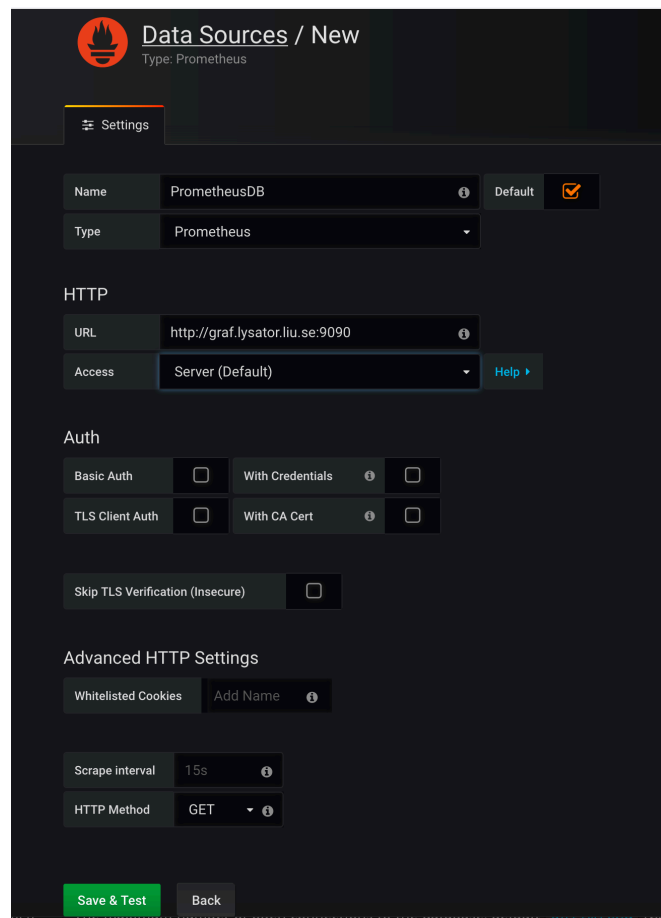
After URL, you should see user name and password screen for Grafana. At this moment, the Grafana is running successfully. Enter username and password as below.

- UserName: admin
- Password: admin

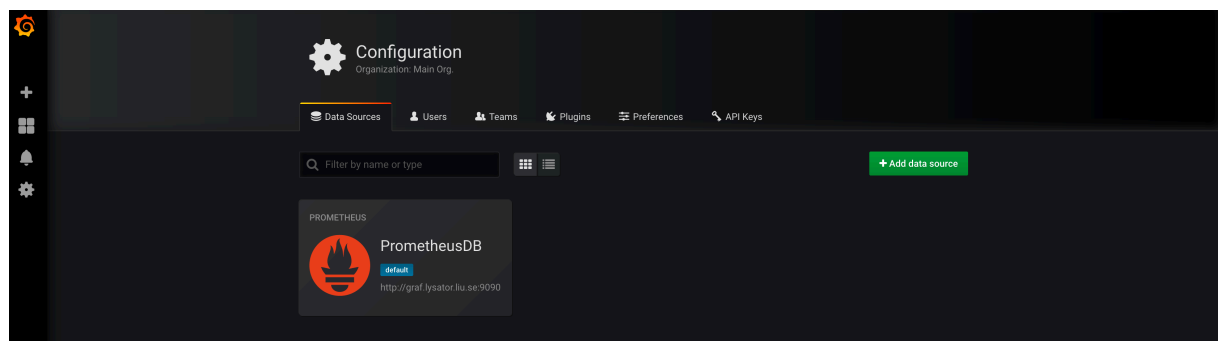
It will ask you to add the data source as per below image. Click on 'Add Data Source' and select 'Proetheus' and fill the required information as provided in another image.



- **You are about to visualize different metrics in a moment. You are using Grafana (i.e. front end) and Prometheus (DB). Prometheus is database that is available on ‘<http://graf.lysator.liu.se:9090>’.**



Once you click ‘Save & Test’. You should see ‘Green Message’ at top right stating that it has been successful and save. **Click ”Back” to go to main page.** The page should look like this now:

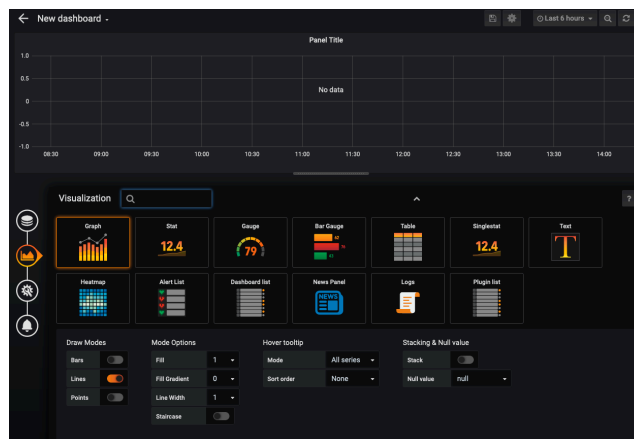


Now Grafana is running and has been connected to Prometheus database that contain information from the server called ‘Lysator’. This is a computer associated in LiU. You can visit this organization at <https://www.lysator.liu.se/>. You are about to visualize the data from these servers.

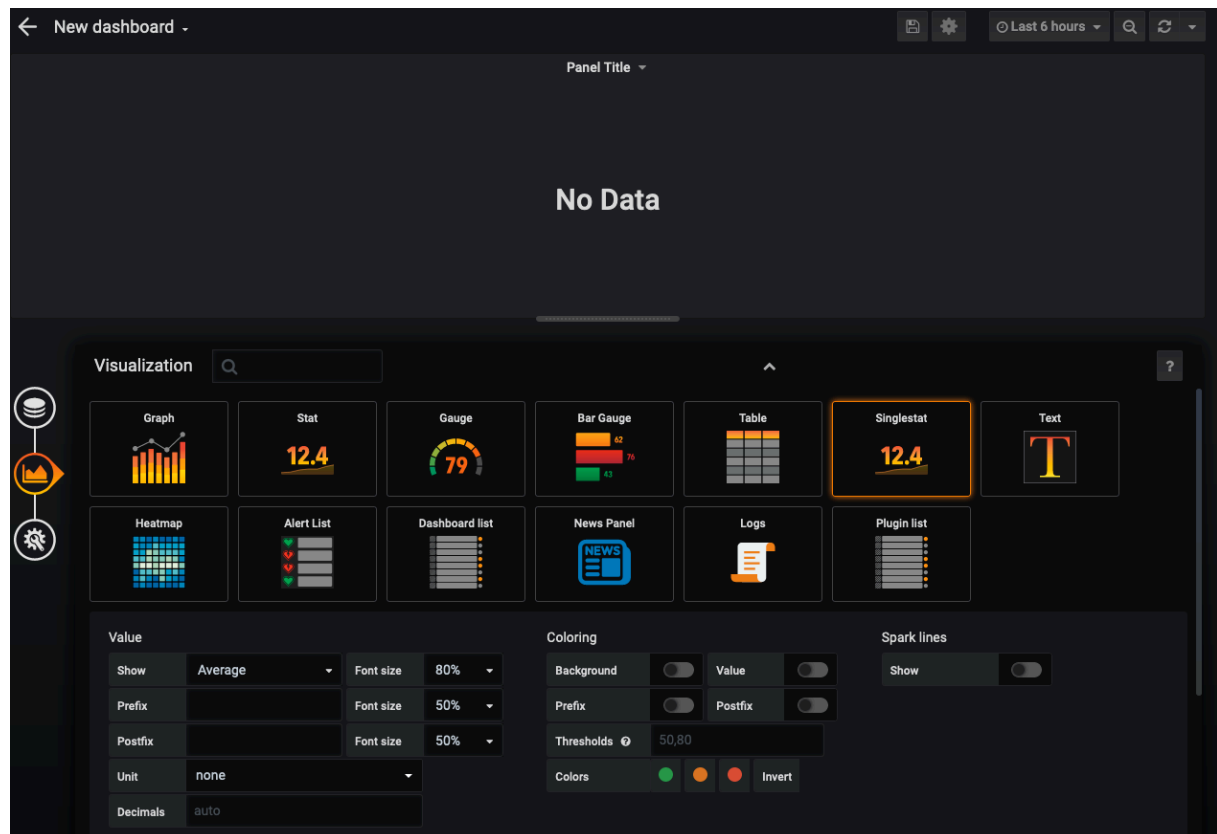
Exercise 2.4:

Add a new Dashboard:

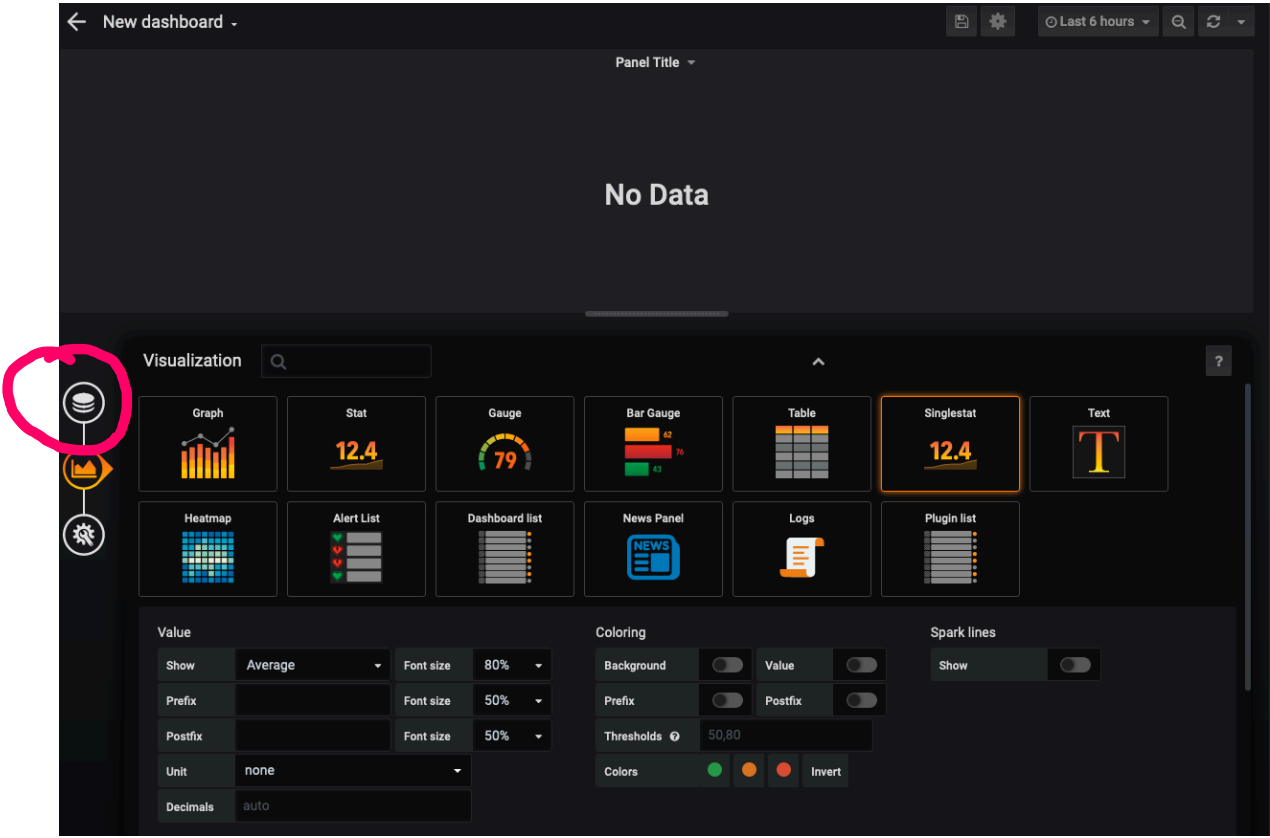
Bring mouse on top-left corner on “+” sign and select ‘Dashboard’. Click on ”Choose Visualization”. You will be provided with many different options such as graphs, tables, single test and rows as per image below. These are different panels depending on what you need to see. **An alternate way is to click on ”+” sign on top right corner.**



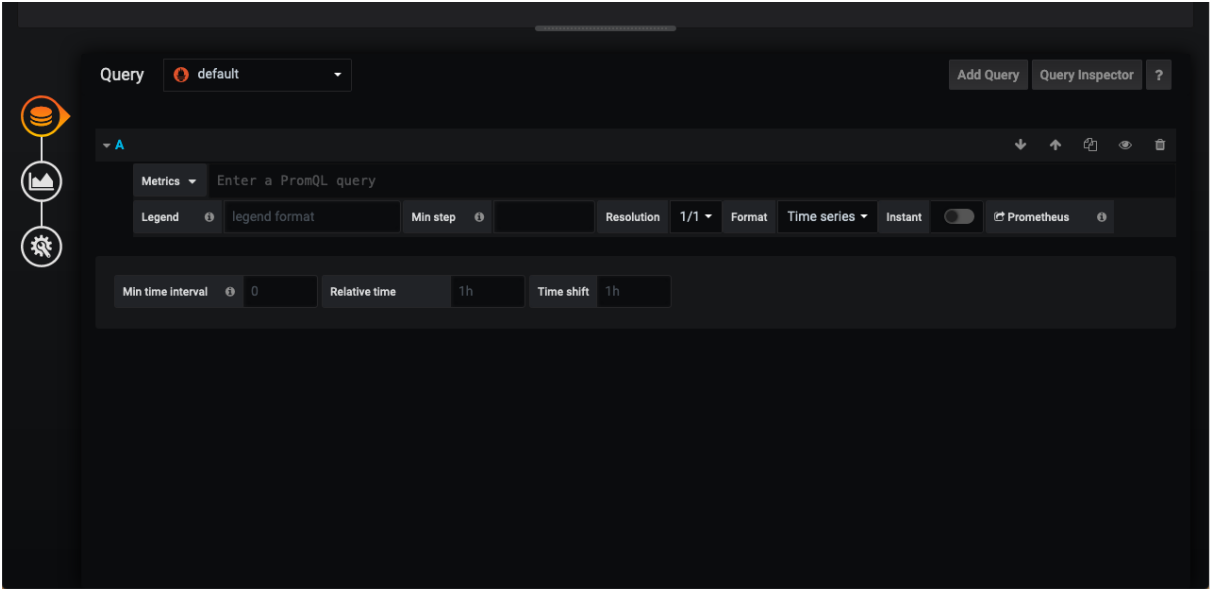
Click on ‘Singlestat’ and you will see the below image.



This is a panel that will help you visualize the data, but you still need to bind this panel with database so that it can retrieve the values and show you on screen. You will see the database sign as shown below. Click on it:

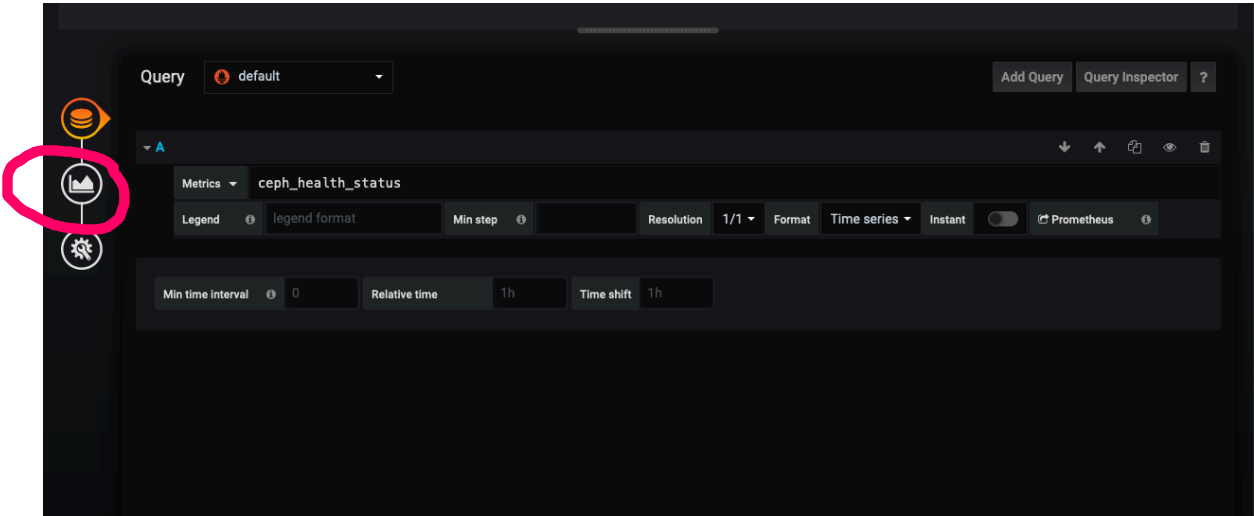


Once you click above circled database sign. You will see the new screen:

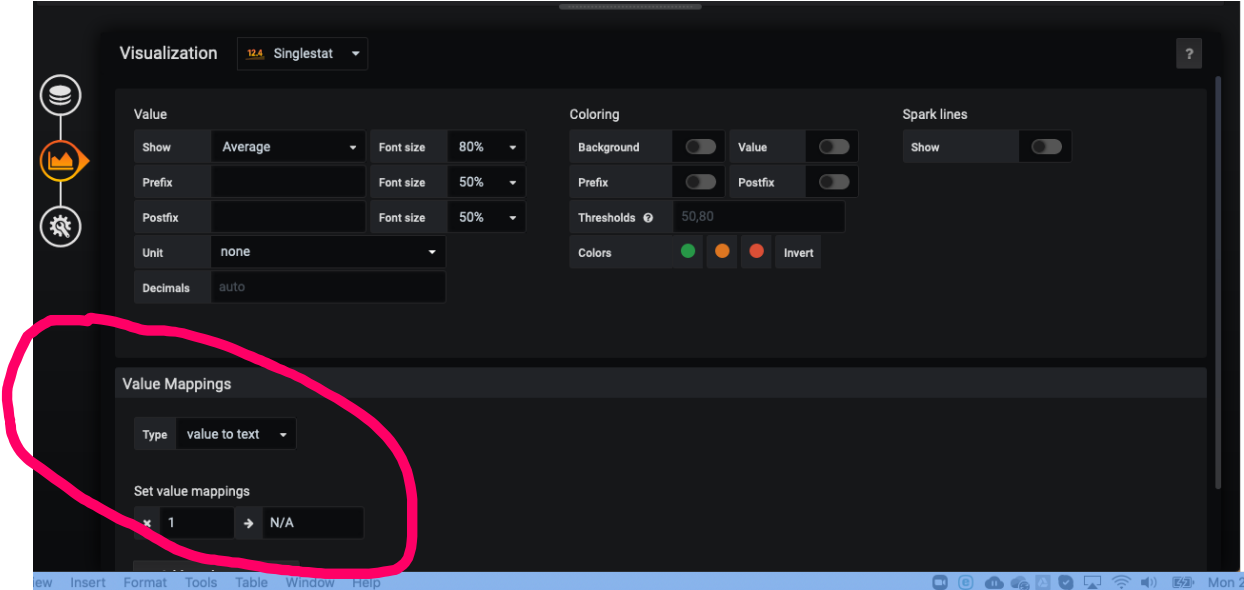


Write a query "ceph_health_status" in the "Metrics" field where it also said 'Enter a PromQL query'. This query extract the data before you visualize it. You will notice that now the panel says "N/A". The data, in the form of number, have been extracted by database by using the

query. We need to make sense of this data to human being. Now. Click on the 'visualization sign' as shown in the following diagram.

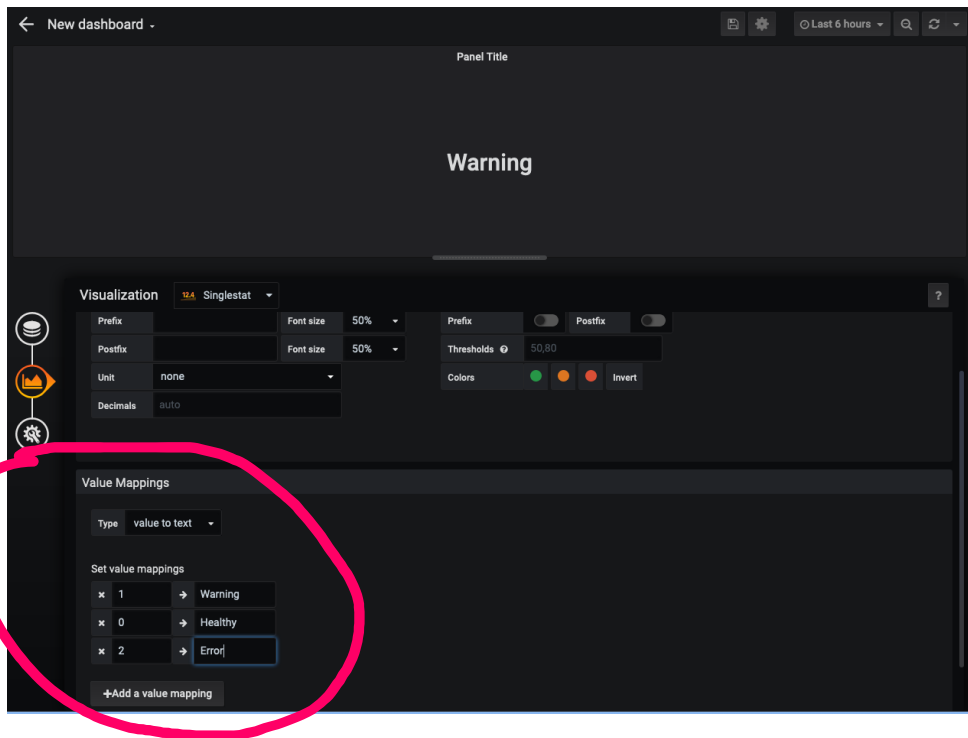


The screen will be changed as follows:

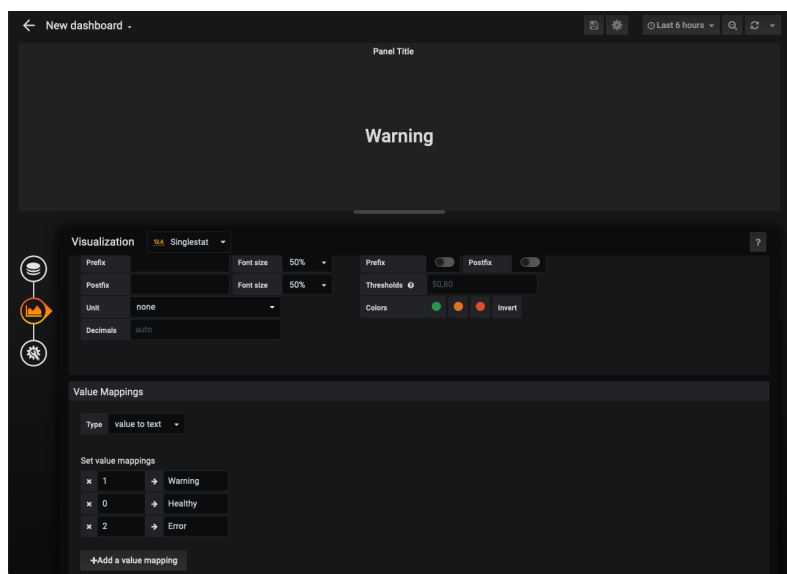


We need to fill information on "Set Value Mapping". You can add as many inputs by clicking on the button "Add a value mapping".

Add values as shown in below image.



Now you should see the dashboard that will tell you what is the health of the website as per below image. I can see that there is a warning. We leave that.



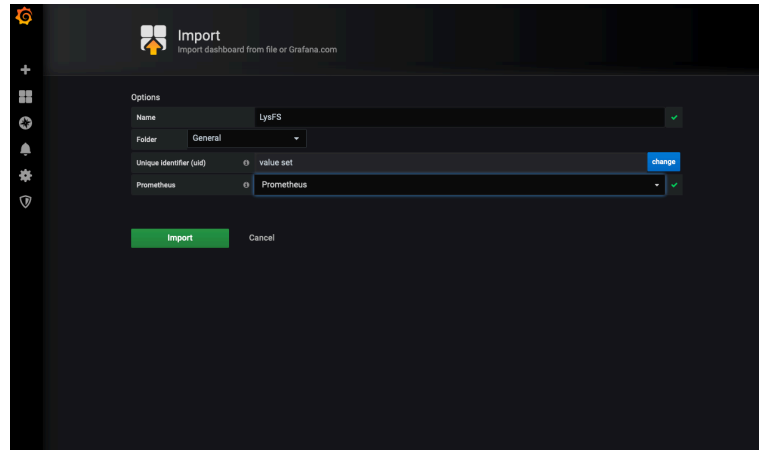
You can add many panels by clicking the 'orange + sign' at the top middle. Below are some sample queries that you can run to experiment.

Purpose	Query
To know free space	$(\text{ceph_cluster_total_bytes} - \text{ceph_cluster_total_used_bytes}) / \text{ceph_cluster_total_bytes}$
Write Throughput	ceph osd op w in bytes
IO Write	ceph osd op w

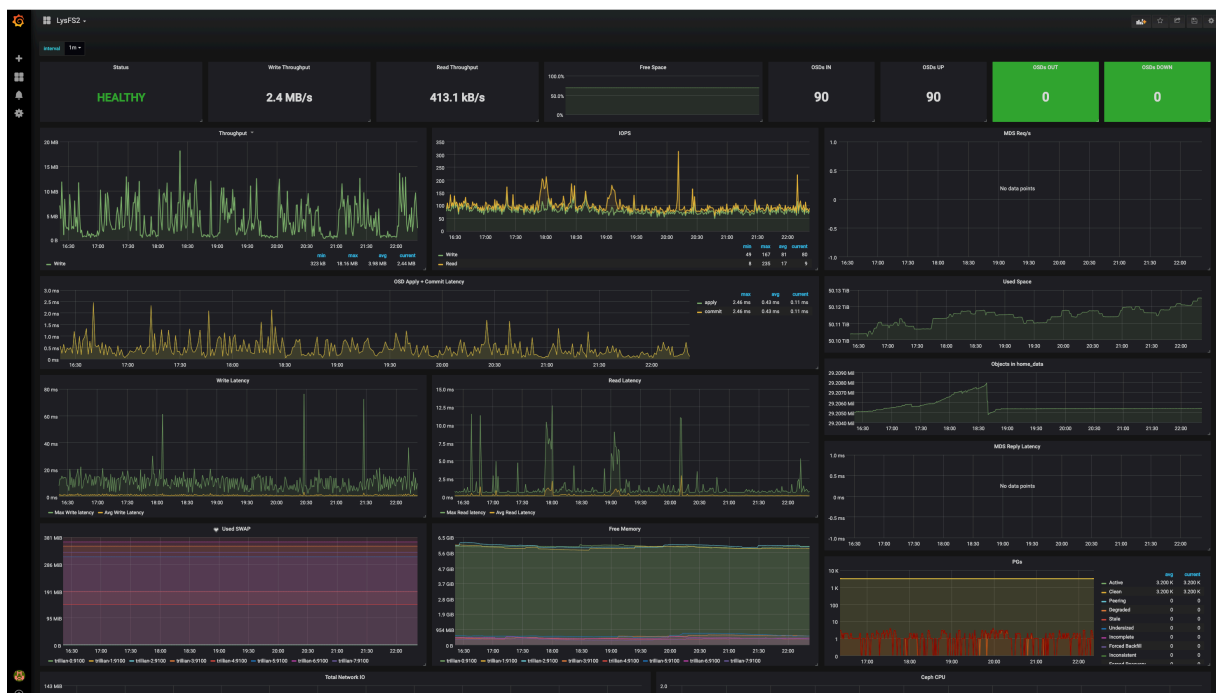
You can create as many dashboards as you want. You can create as many panels as you want.

Exercise 2.5:

You can also import pre-built dashboards, but you need to map it to your database. One pre-built dashboard is provided on course website. Download it. Bring it to Grafana (Find the import option once you click the "+" sign on top left corner). You may have to add the "Prometheus" database as shown in the following diagram:



adding JSON file, you should be able to see the following picture.



- How do you feel about these visualizations?
- What values do these visualizations bring to your professional life?
- Share your reflection of any 4 of the visualization metrics one paragraph each group such as what the metric represent, how effective it is and what do you feel about it with pros and cons.

Upload the screen shots with questions and answers and your reflections to LISAM.