**Course-End Project**

**Application Metrics Monitoring**

Steps to be followed:

1. Log in to the lab to access the Linux terminal and install the Prometheus metric server
2. Configure Java Application to publish custom metrics for API responses to the Prometheus collector
3. Configure Grafana visualization tool to visualize metrics
4. Create a metrics dashboard for API responses

**Step 1:** **Log in to the lab to access the Linux terminal and install the Prometheus metric server**

**FIRST CHECK THE JAVA VERSION**

**java --version**

**if not 17 then:**

[**https://askubuntu.com/questions/84483/how-to-completely-uninstall-java**](https://askubuntu.com/questions/84483/how-to-completely-uninstall-java)

**remove java**

**sudo apt install openjdk-17-jdk**

**apt install maven -y**

* 1. Run the following command to switch to the root user:   
     **sudo su  
       
     A close up of a word

     Description automatically generated**
  2. Run the following command to update the package lists:  
     **apt update**

A screenshot of a computer

Description automatically generated

* 1. Run the following command to install Prometheus on the system:  
     **apt install prometheus**  
       
     A close-up of a computer screen

     Description automatically generated
  2. Run the following command to open the Prometheus configuration file for editing:  
     **nano /etc/prometheus/prometheus.yml  
       
     **
  3. Scroll to the bottom of the file, locate the **scrape\_configs** section, and add the following content:  
     **- job\_name: 'books\_app'**

**metrics\_path: /actuator/prometheus**

**static\_configs:**

**- targets: ['localhost:8089']**  
A screenshot of a computer

Description automatically generated

* 1. Restart the Prometheus service and check its status by running the following commands:  
     **service prometheus restart**

**service prometheus status**  
A close-up of a document

Description automatically generated

**Step 2: Configure Java Application to publish custom metrics for API responses to the Prometheus collector**

* 1. Run the following command to clone the git repository:

**git clone** <https://github.com/simplilearn10/SpringBootPrometheusGrafana.git>



* 1. Navigate to the project folder using the following command:  
     **cd SpringBootPrometheusGrafana/**

* 1. Run the following command to clean and build the Maven project:   
     **mvn clean install**A screen shot of a computer

     Description automatically generated
  2. Run the Spring Boot application using the following command:  
     **java -jar target/spring-boot-metrics-0.0.1-SNAPSHOT.jar**  
     A screenshot of a computer

     Description automatically generated
  3. Fetch data from the API using the following commands:  
     **curl -XGET http://localhost:8089/api/books**

**curl -XGET http://localhost:8089/api/books?title=Domain+Driven+Design**

**curl -XGET http://localhost:8089/api/books?title=Clean+code**

**curl -XGET http://localhost:8089/api/books?title=Fundamental+Algorithms**  
  
A close up of a text

Description automatically generated

**Step 3: Configure Grafana visualization tool to visualize metrics**

1. Create a directory for storing APT keyrings

**sudo mkdir -p /etc/apt/keyrings/**

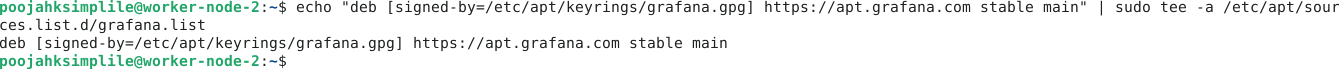


1. Run the following command to download the Grafana GPG key, dearmor it, and store it in the keyring directory:  
   **wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null**



1. Add a repository for stable releases

**echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list**

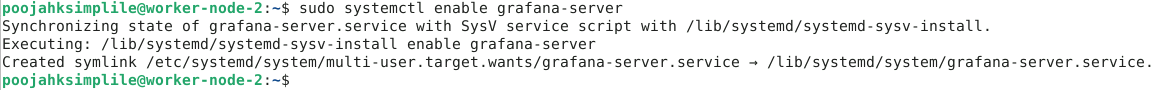


1. Run the following command to update the list of available packages:  
   **sudo apt-get update**

**A computer screen shot of a computer

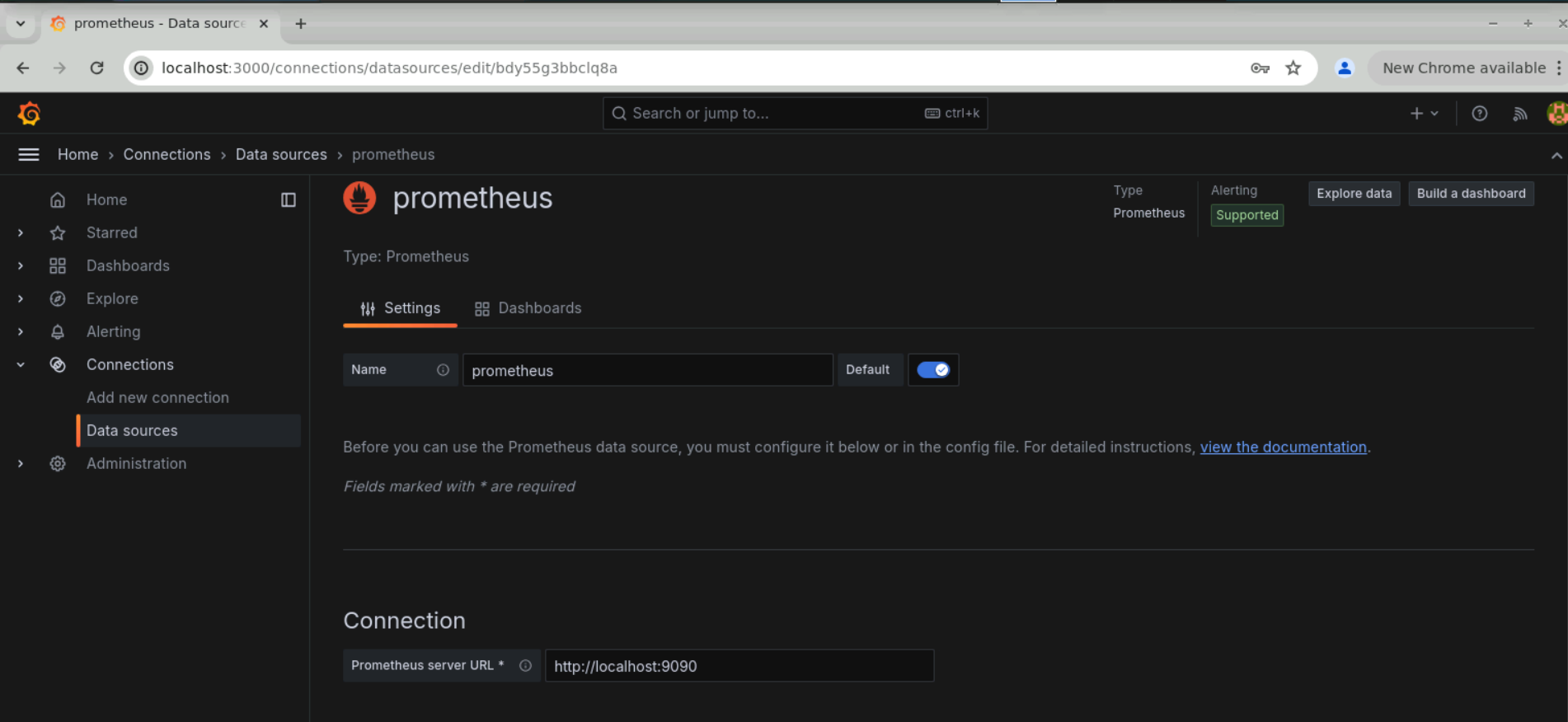
Description automatically generated**

1. Run the following command to install Grafana OSS:   
   **sudo apt-get install grafana  
     
   A screenshot of a computer

   Description automatically generated**
2. Start the Grafana service  
   **sudo systemctl start grafana-server  
     
   **
3. Enable Grafana to start at system boot  
   **sudo systemctl enable grafana-server**  
   
4. Check the status of Grafana using the following command:  
   **sudo systemctl status grafana-server**  
   A computer screen with text

   Description automatically generated
5. Access the Grafana web interface using your server’s IP address:  
   [**http://your-server-ip:3000**](http://your-server-ip:3000) **Note:** Replace **your-server-ip** with your server's IP address  
     
   A screenshot of a computer

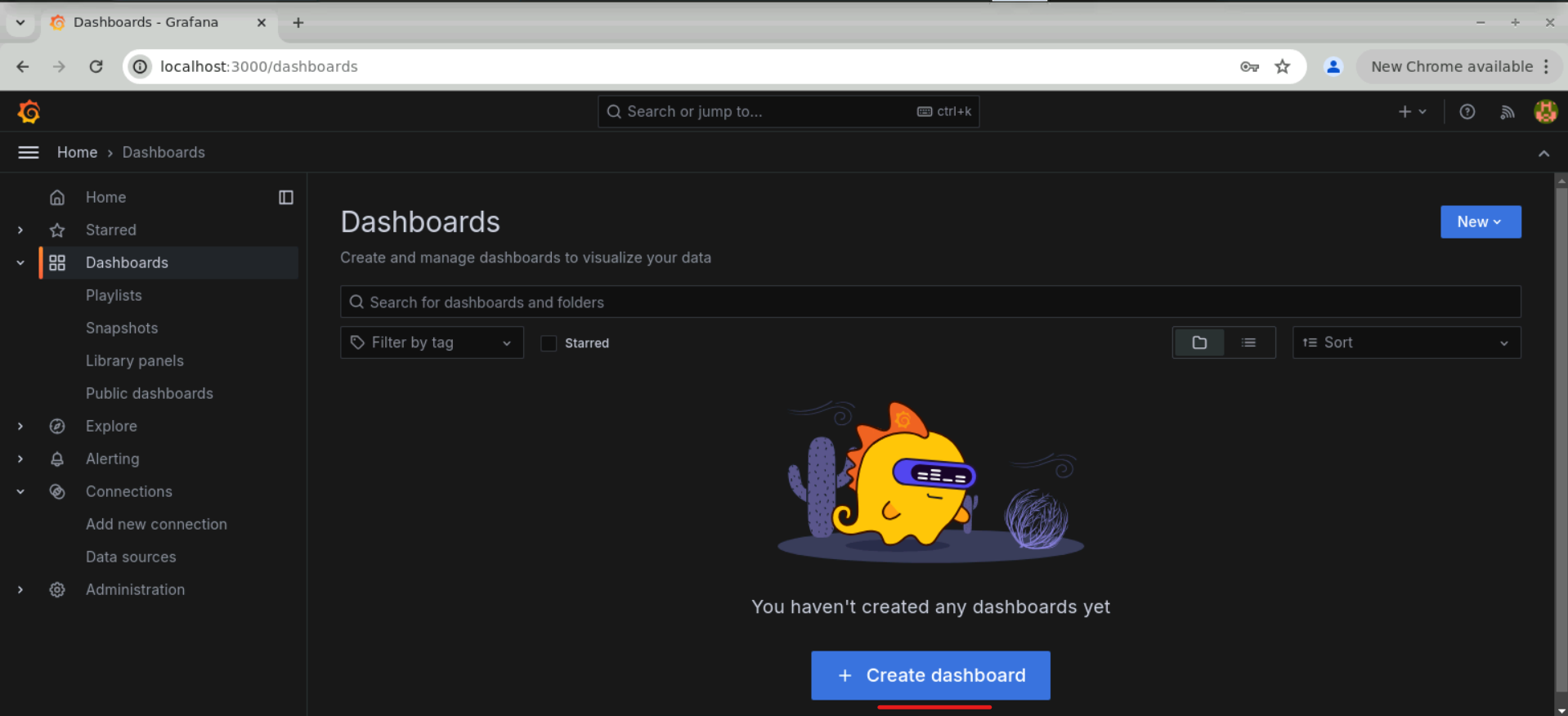
   Description automatically generated  
      
   Above is the landing page.
6. Log in to Grafana using the default credentials, then click **Submit** as shown in the screenshot below:  
   **Username: admin  
   Password: admin**  
   A screenshot of a computer

   Description automatically generated
7. Go to **Data sources** and select **Prometheus** as data source, as shown   
   in the screenshot below:  
     
   
8. Scroll to the bottom and click on **Save & test** button to validate Prometheus connectivity  
     
   A screenshot of a computer

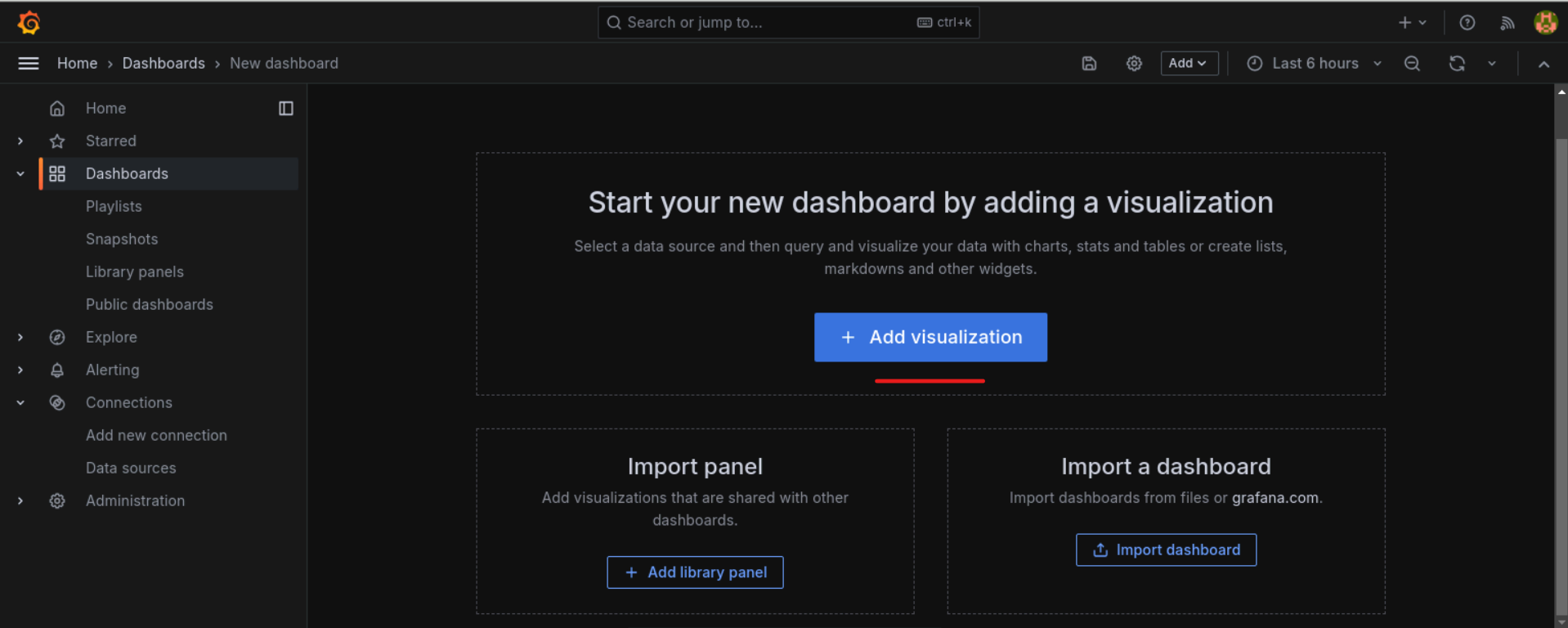
   Description automatically generated

**Step 4: Create a metrics dashboard for API responses**

1. Open the Grafana main menu, select Dashboard, and click **Create dashboard**



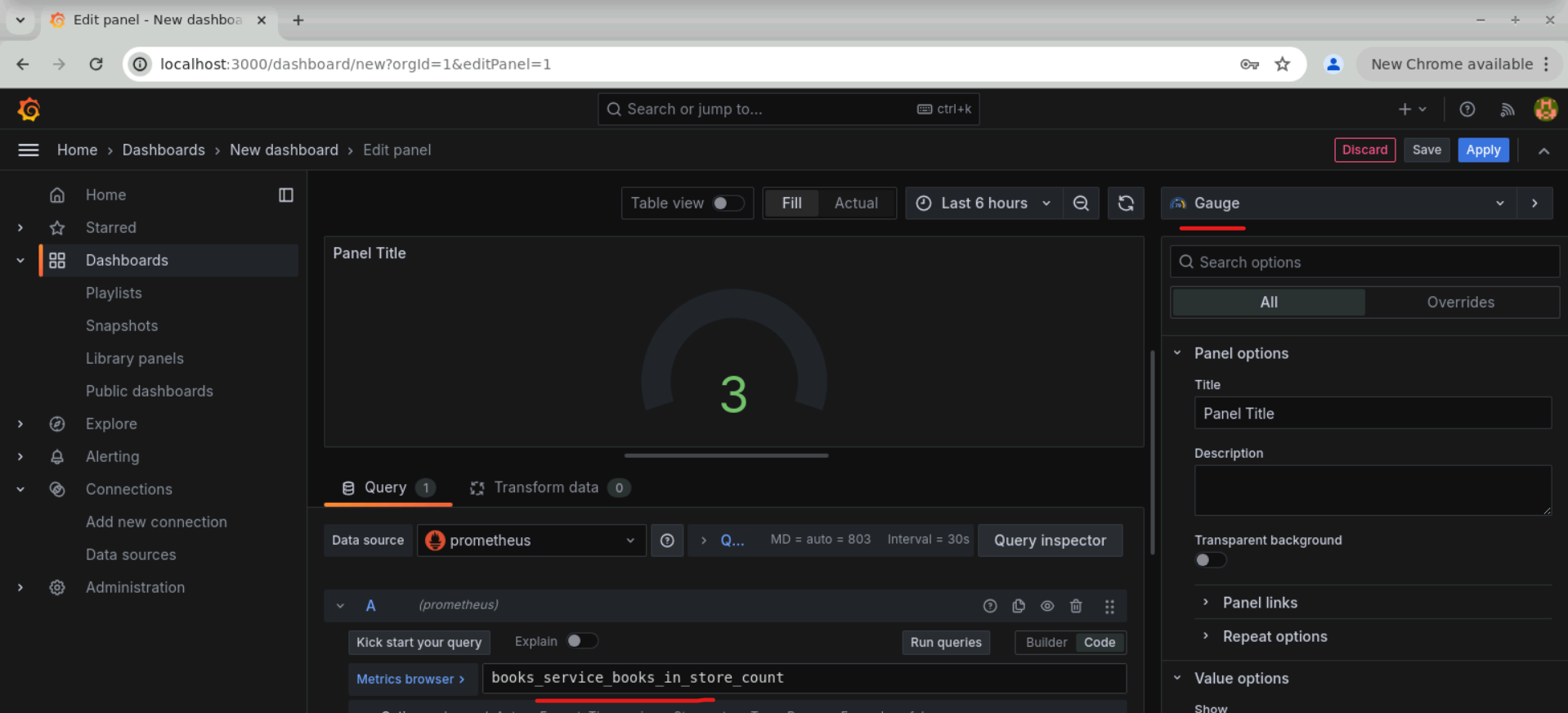
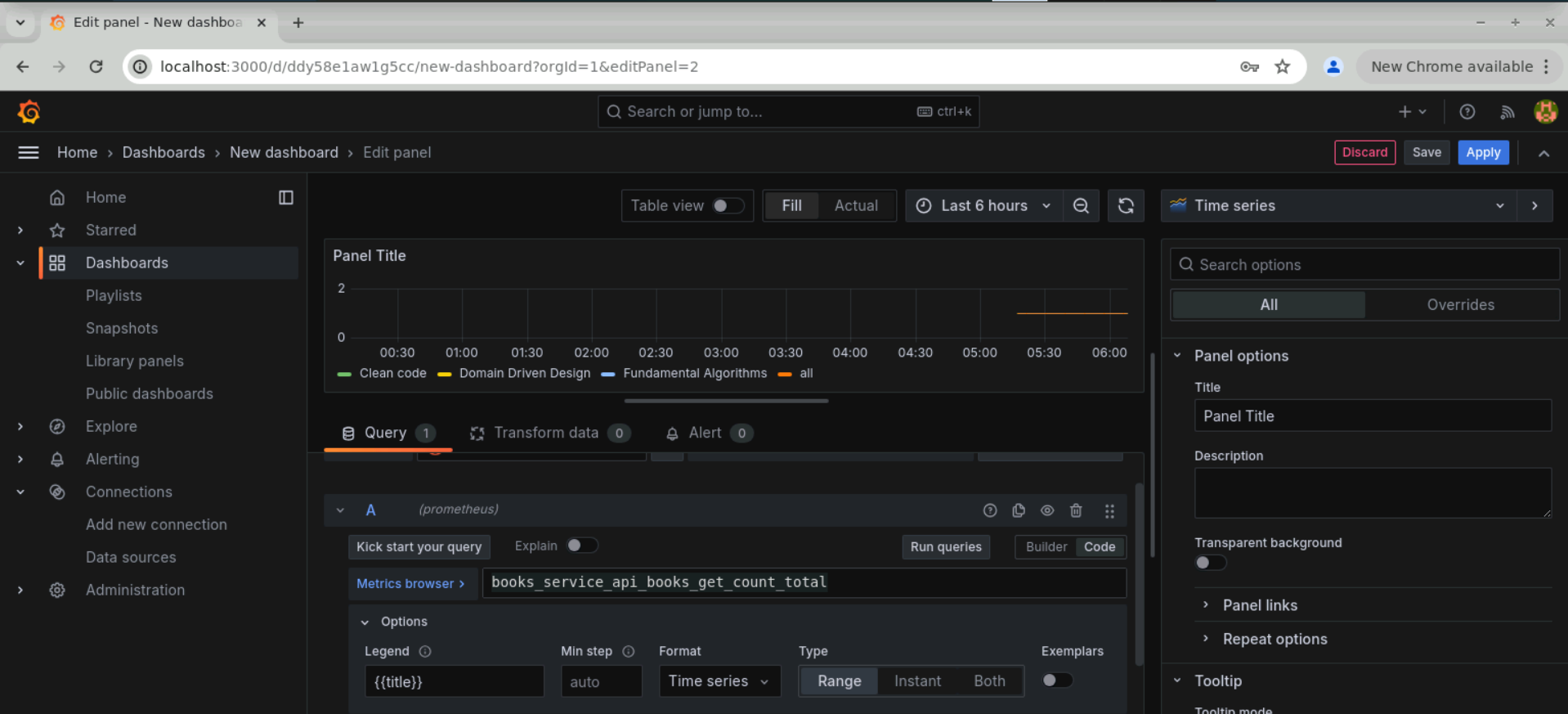
1. Click **Add visualization**

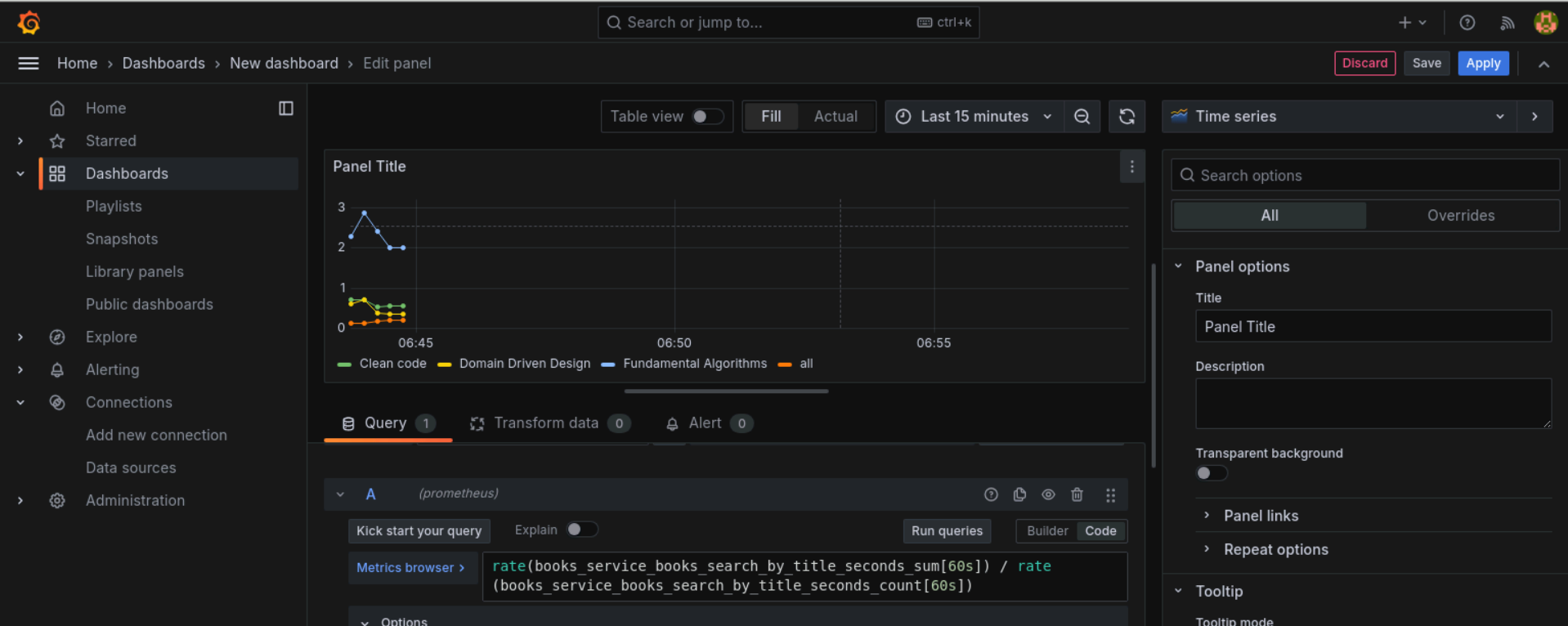


1. Select the **Prometheus** data source to proceed

A screenshot of a computer

Description automatically generated

1. Add a Gauge panel using the metric **books\_service\_books\_in\_store\_count** to represent the number of books available in stock  
     
   
2. Use the metric **books\_service\_api\_books\_get\_count\_total** to track total API requests. Set the **Legend** field to **{{title}}** for clarity  
     
   
3. Enter the following query to measure the average response time for book searches and set the **Legend** field to {{title}}. Then, click **Save**:

**rate(books\_service\_books\_search\_by\_title\_seconds\_sum[60s]) / rate(books\_service\_books\_search\_by\_title\_seconds\_count[60s])**

Below is the Grafana dashboard that visualizes API response data, including total API requests and average response time for book searches:  
  
A screenshot of a computer

Description automatically generated

By following these steps, you have successfully set up Prometheus and Grafana for monitoring application metrics, configured a Java application to expose custom metrics, and created a dashboard to visualize API response data. This setup provides valuable insights into application performance and API usage.