Lesson End Project
Monitoring MySQL Database Using Prometheus and Setting up Alerts

Project agenda: To configure Prometheus for monitoring a MySQL database, collect key performance metrics using MySQL Exporter, and set up alerting rules using Alertmanager to notify of any issues for proactive database management and performance optimization

Description: You are a DevOps engineer responsible for maintaining the performance and stability of MySQL databases within your organization. Due to performance bottlenecks during peak traffic periods, you have been tasked with implementing a monitoring solution. By using Prometheus and MySQL Exporter, you will gather essential MySQL metrics, analyze performance, and set up alerting rules through Alertmanager to notify the team in real time of any potential performance degradation or failures.

Tools required: Linux operating system and Docker

Prerequisites: Ensure that Docker is installed in the lab before proceeding.

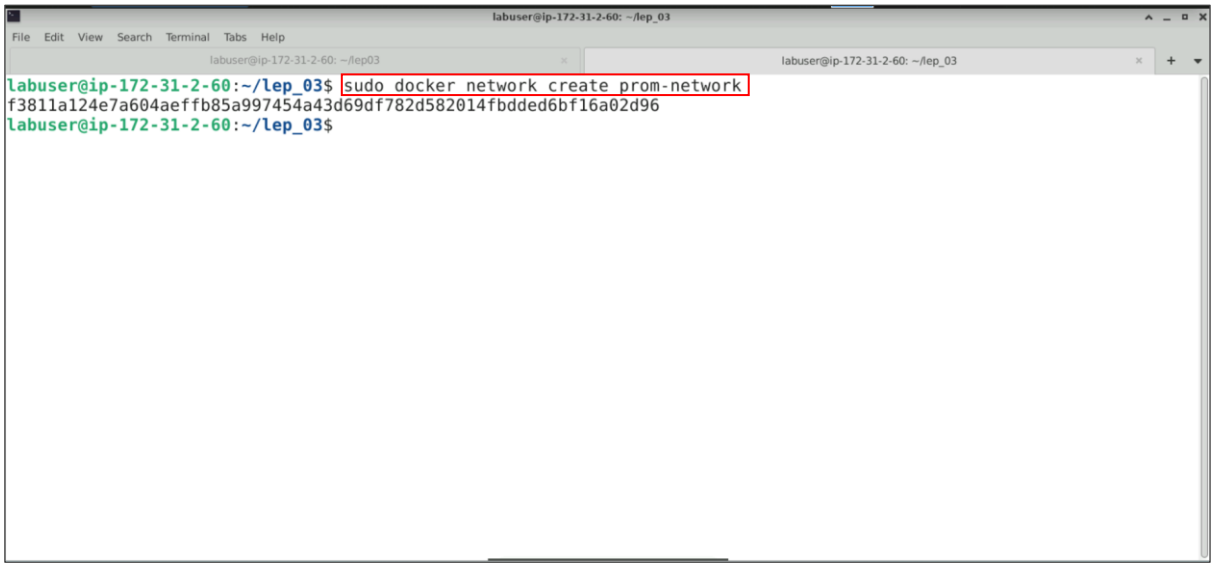
Expected deliverables: A setup for MySQL monitoring with Prometheus, including MySQL Exporter configuration, PromQL queries for performance analysis, and alerting rules for key MySQL performance indicators

Steps to be followed:

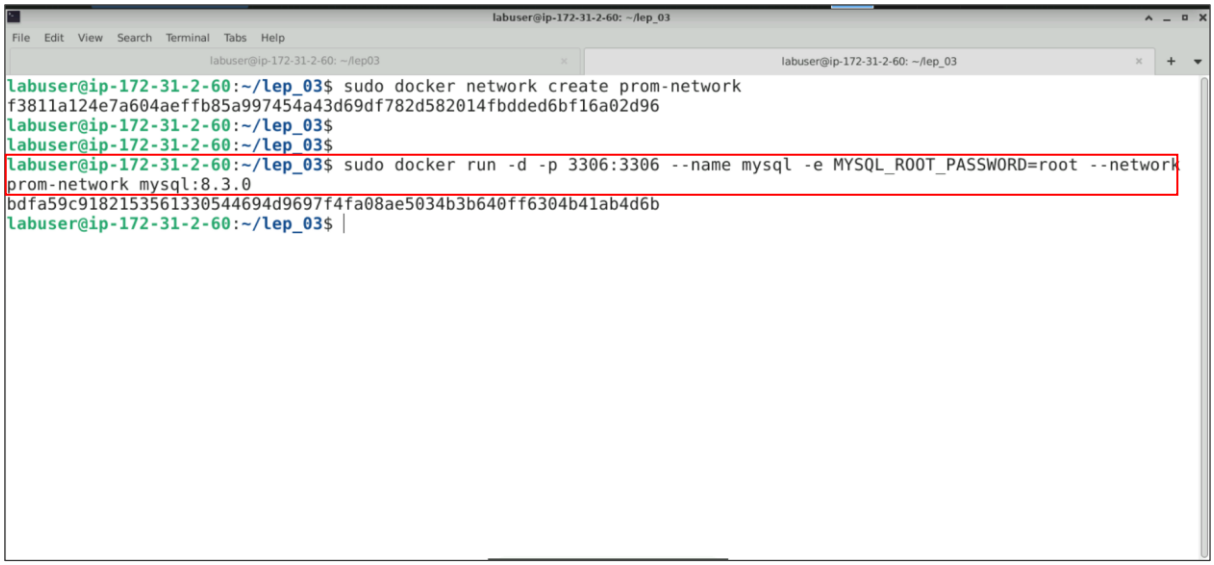
1. Set up MySQL using Docker
2. Set up MySQL Server Exporter with Docker
3. Configure Alertmanager using Docker
4. Configure and start Prometheus in a Docker container
5. Explore MySQL Exporter metrics and alerting config on Prometheus
6. Simulate MySQL failure

Step 1: Set up MySQL using Docker

1.1 Open the terminal and run the following command to create a Docker network:
sudo docker network create prom-network



1.2 Run the following command to set up MySQL in a Docker container:
sudo docker run -d -p 3306:3306 --name mysql -e MYSQL_ROOT_PASSWORD=root --network prom-network mysql:8.3.0



Step 2: Set up MySQL Server Exporter with Docker

2.1 Run the following command to list all the running containers to identify the MySQL container id:
docker ps -a

```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$ sudo docker run -d -p 3306:3306 --name mysql -e MYSQL_ROOT_PASSWORD=root --network
prom-network mysql:8.3.0
bdfa59c918215356133054469 4d9697f4fa08ae5034b3b640ff6304b41ab4d6b
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$ docker exec -it 679ad782cde7 /bin/bash
Error response from daemon: No such container: 679ad782cde7
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$ docker ps -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED              STATUS
         PORTS                                                   NAMES
bdfa59c91821   mysql:8.3.0                    "docker-entrypoint.s…"   About a minute ago   Up About a minute
         0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysql
37c9b2772110   prom/prometheus:latest         "/bin/prometheus --c…"   3 hours ago          Exited (0) About an hour
 ago                                                             metrics-demo_prometheus_1
48ac42966aa4   metrics-demo_flask-app         "python3 server.py"      3 hours ago          Exited (137) About an ho
ur ago                                                           metrics-demo_flask-app_1
d1f548ce38be   bitnami/node-exporter:latest   "/opt/bitnami/node-e…"   3 hours ago          Exited (2) About an hour
 ago                                                             metrics-demo_node-exporter_1
09740a73cc4d   grafana/grafana                "/run.sh"                3 hours ago          Exited (0) About an hour
 ago                                                             grafana
labuser@ip-172-31-2-60:~/lep_03$ docker exec -it bdfa59c91821 /bin/bash
bash-4.4#
```

2.2 Run the given command to open an interactive Bash shell inside the MySQL Docker container:
docker exec -it bdfa59c91821 /bin/bash

```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$ sudo docker run -d -p 3306:3306 --name mysql -e MYSQL_ROOT_PASSWORD=root --network
prom-network mysql:8.3.0
bdfa59c918215356133054469 4d9697f4fa08ae5034b3b640ff6304b41ab4d6b
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$ docker exec -it 679ad782cde7 /bin/bash
Error response from daemon: No such container: 679ad782cde7
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$ docker ps -a
CONTAINER ID   IMAGE                          COMMAND                  CREATED              STATUS
         PORTS                                                   NAMES
bdfa59c91821   mysql:8.3.0                    "docker-entrypoint.s…"   About a minute ago   Up About a minute
         0.0.0.0:3306->3306/tcp, :::3306->3306/tcp, 33060/tcp   mysql
37c9b2772110   prom/prometheus:latest         "/bin/prometheus --c…"   3 hours ago          Exited (0) About an hour
 ago                                                             metrics-demo_prometheus_1
48ac42966aa4   metrics-demo_flask-app         "python3 server.py"      3 hours ago          Exited (137) About an ho
ur ago                                                           metrics-demo_flask-app_1
d1f548ce38be   bitnami/node-exporter:latest   "/opt/bitnami/node-e…"   3 hours ago          Exited (2) About an hour
 ago                                                             metrics-demo_node-exporter_1
09740a73cc4d   grafana/grafana                "/run.sh"                3 hours ago          Exited (0) About an hour
 ago                                                             grafana
labuser@ip-172-31-2-60:~/lep_03$ docker exec -it bdfa59c91821 /bin/bash
bash-4.4#
```

Note: Make sure to replace the **CONTAINER ID** in the above command with the correct **CONTAINER ID** of your MySQL Docker container per the list.

2.3 Run the following command to connect to the MySQL server using the root credentials (**password=root**):
mysql -u root -p -h 127.0.0.1

```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
37c9b2772110   prom/prometheus:latest         "/bin/prometheus --c…"   3 hours ago          Exited (0) About an hour
 ago                                                                   metrics-demo_prometheus_1
48ac42966aa4   metrics-demo_flask-app         "python3 server.py"      3 hours ago          Exited (137) About an ho
ur ago                                                                 metrics-demo_flask-app_1
d1f548ce38be   bitnami/node-exporter:latest   "/opt/bitnami/node-e…"   3 hours ago          Exited (2) About an hour
 ago                                                                   metrics-demo_node-exporter_1
09740a73cc4d   grafana/grafana                "/run.sh"                3 hours ago          Exited (0) About an hour
 ago                                                                   grafana
labuser@ip-172-31-2-60:~/lep_03$ docker exec -it bdfa59c91821 /bin/bash
bash-4.4# mysql -u root -p -h 127.0.0.1
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

2.4 Enter the following commands to create a new MySQL user **exporter** with a password **pass123** and grant privileges to the **exporter** user on all databases; exit MySQL and then exit the bash session
CREATE USER 'exporter'@'localhost' IDENTIFIED BY 'pass123' WITH MAX_USER_CONNECTIONS 3;
GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'localhost';
quit;
exit

```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'exporter'@'localhost' IDENTIFIED BY 'pass123' WITH MAX_USER_CONNECTIONS 3;
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'localhost';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> quit;
Bye
bash-4.4# exit
exit
labuser@ip-172-31-2-60:~/lep_03$
```

2.5 Run the following command to create the configuration file **config.my-cnf** using **vim** editor:

sudo vim config.my-cnf

```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.3.0 MySQL Community Server - GPL

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'exporter'@'localhost' IDENTIFIED BY 'pass123' WITH MAX_USER_CONNECTIONS 3;
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'localhost';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> quit;
Bye
bash-4.4# exit
exit
labuser@ip-172-31-2-60:~/lep_03$ sudo vim config.my-cnf
```

2.6 Copy and paste the following configuration, then save and exit the file:

[client]

user = exporter

password = pass123

host = mysql

```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
[client]
user = exporter
password = pass123
host = mysql
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                        5,0-1         All
```

2.7 Run the following command to initialize the MySQL exporter container:
sudo docker run -d -p 9104:9104 --name mysql-exporter --network prom-network -v $(pwd)/config.my-cnf:/cfg/config.my-cnf prom/mysqld-exporter:main --config.my-cnf=/cfg/config.my-cnf

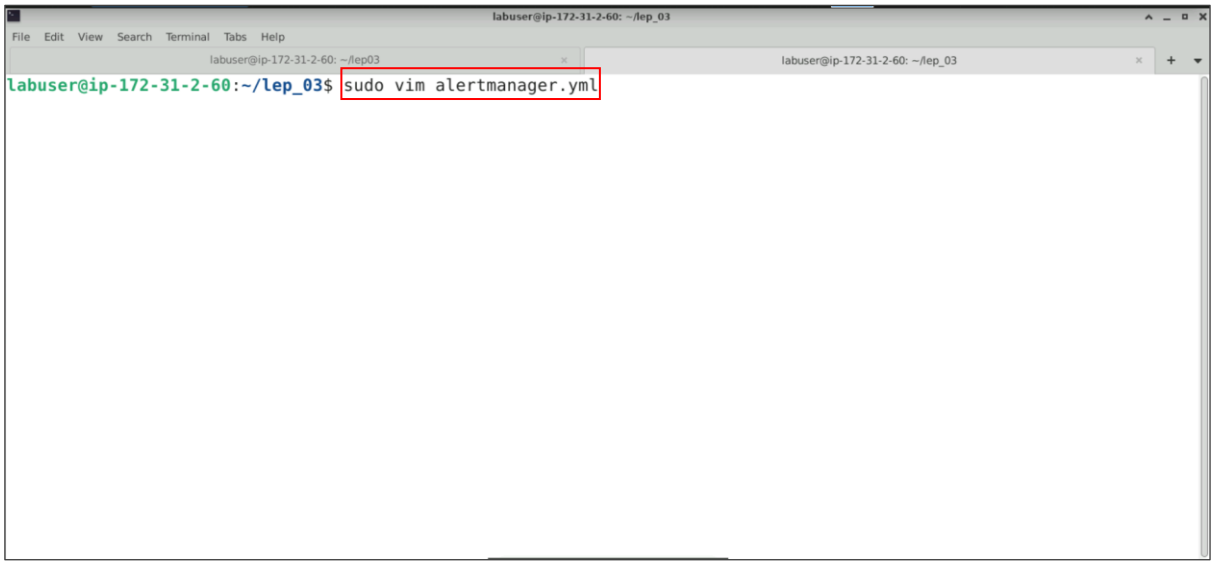
```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'exporter'@'localhost' IDENTIFIED BY 'pass123' WITH MAX_USER_CONNECTIONS 3;
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT PROCESS, REPLICATION CLIENT, SELECT ON *.* TO 'exporter'@'localhost';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> quit;
Bye
bash-4.4# exit
exit
labuser@ip-172-31-2-60:~/lep_03$ sudo vim config.my-cnf
labuser@ip-172-31-2-60:~/lep_03$
labuser@ip-172-31-2-60:~/lep_03$ sudo docker run -d -p 9104:9104 --name mysql-exporter --network prom-network -v $(p
wd)/config.my-cnf:/cfg/config.my-cnf prom/mysqld-exporter:main --config.my-cnf=/cfg/config.my-cnf
cb5ba9880fac1025613c9e9172cfa0e2b2fdd21f4cd8b831e5fcdaf3cefce38b
labuser@ip-172-31-2-60:~/lep_03$
```

2.8 Use the following command to verify if the MySQL Exporter is running:
curl http://localhost:9104/metrics

```
labuser@ip-172-31-2-60: ~/lep_03
File  Edit  View  Search  Terminal  Tabs  Help
labuser@ip-172-31-2-60: ~/lep03        labuser@ip-172-31-2-60: ~/lep_03
labuser@ip-172-31-2-60:~/lep_03$ curl http://localhost:9104/metrics
# HELP go_gc_duration_seconds A summary of the wall-time pause (stop-the-world) duration in garbage collection cycle
s.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0
go_gc_duration_seconds_sum 0
go_gc_duration_seconds_count 0
# HELP go_gc_gogc_percent Heap size target percentage configured by the user, otherwise 100. This value is set by th
e GOGC environment variable, and the runtime/debug.SetGCPercent function. Sourced from /gc/gogc:percent
# TYPE go_gc_gogc_percent gauge
go_gc_gogc_percent 100
# HELP go_gc_gomemlimit_bytes Go runtime memory limit configured by the user, otherwise math.MaxInt64. This value is
 set by the GOMEMLIMIT environment variable, and the runtime/debug.SetMemoryLimit function. Sourced from /gc/gomemli
mit:bytes
# TYPE go_gc_gomemlimit_bytes gauge
go_gc_gomemlimit_bytes 9.223372036854776e+18
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 7
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
```

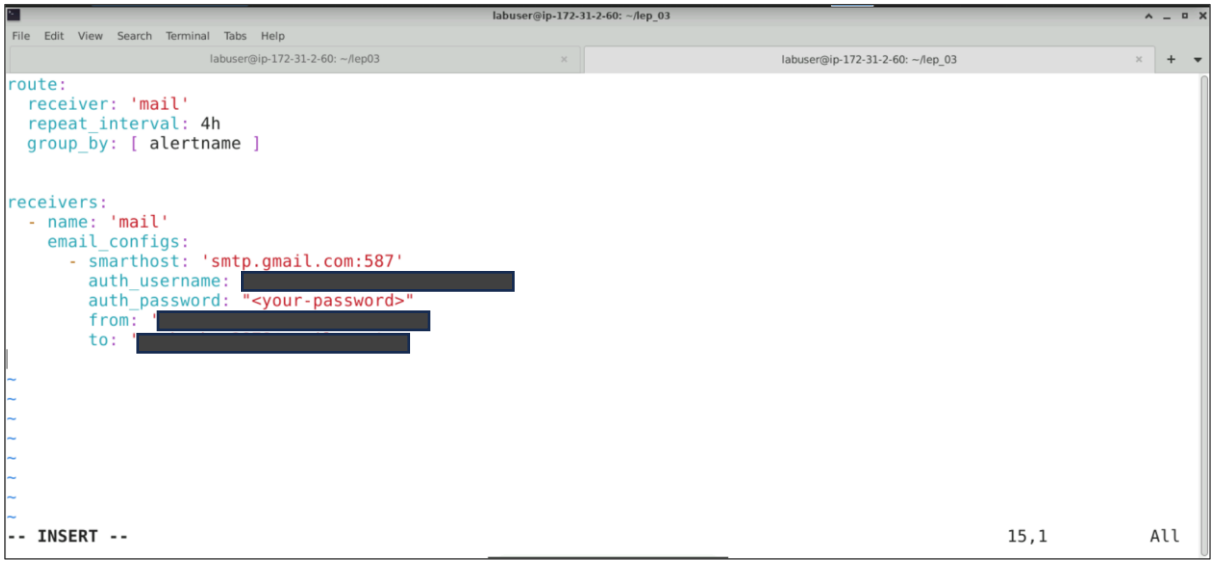
Step 3: Configure Alertmanager using Docker

3.1 Run the following command to create the configuration file **alertmanager.yml** using **vim** editor:
sudo vim alertmanager.yml

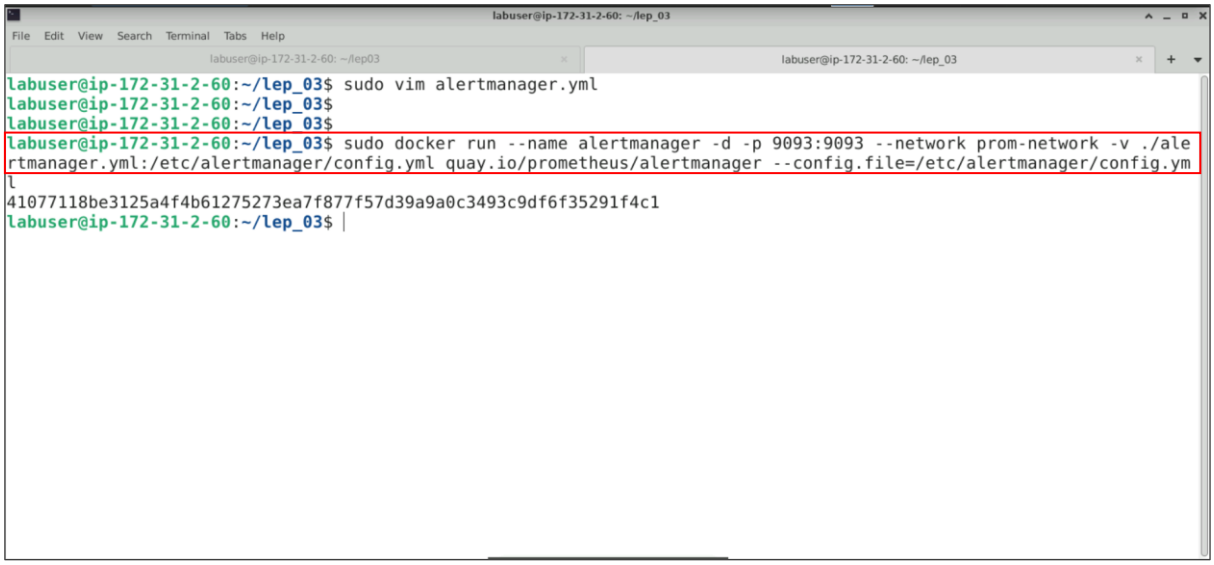


3.2 Copy and paste the following configuration, then save and exit the file:

```
route:
  receiver: 'mail'
  repeat_interval: 4h
  group_by: [ alertname ]

receivers:
  - name: 'mail'
    email_configs:
      - smarthost: 'smtp.gmail.com:587'
        auth_username: '<your_username>'
        auth_password: "<your-password>"
        from: '<your-email>'
        to: '<some-email>'
```

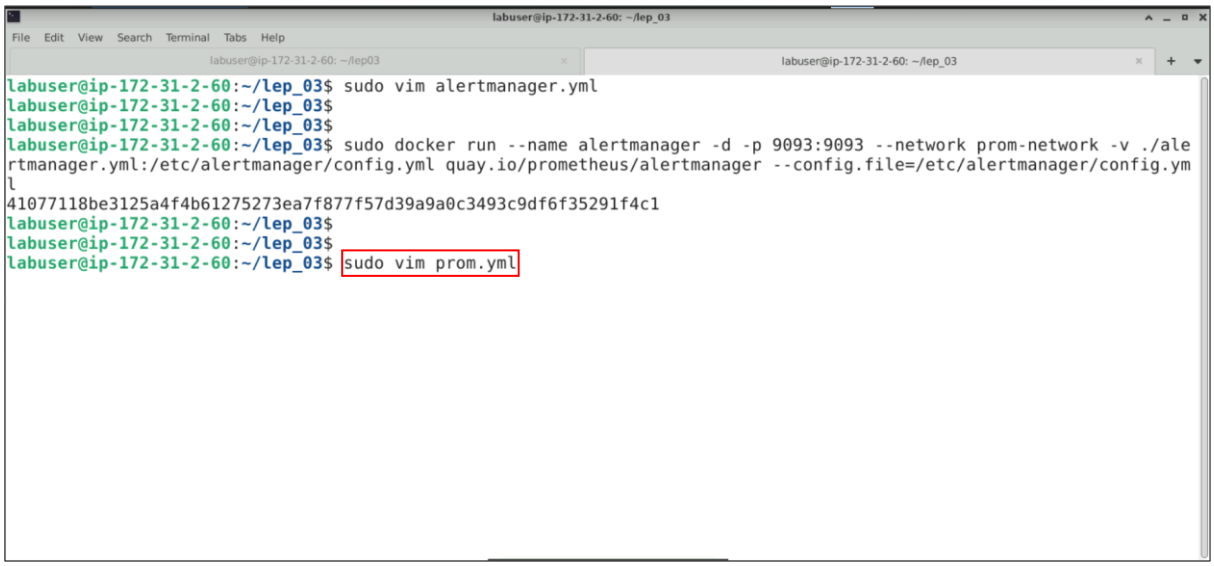


3.3 Run the following command to start the Alertmanager container:
sudo docker run --name alertmanager -d -p 9093:9093 --network prom-network -v ./alertmanager.yml:/etc/alertmanager/config.yml quay.io/prometheus/alertmanager --config.file=/etc/alertmanager/config.yml



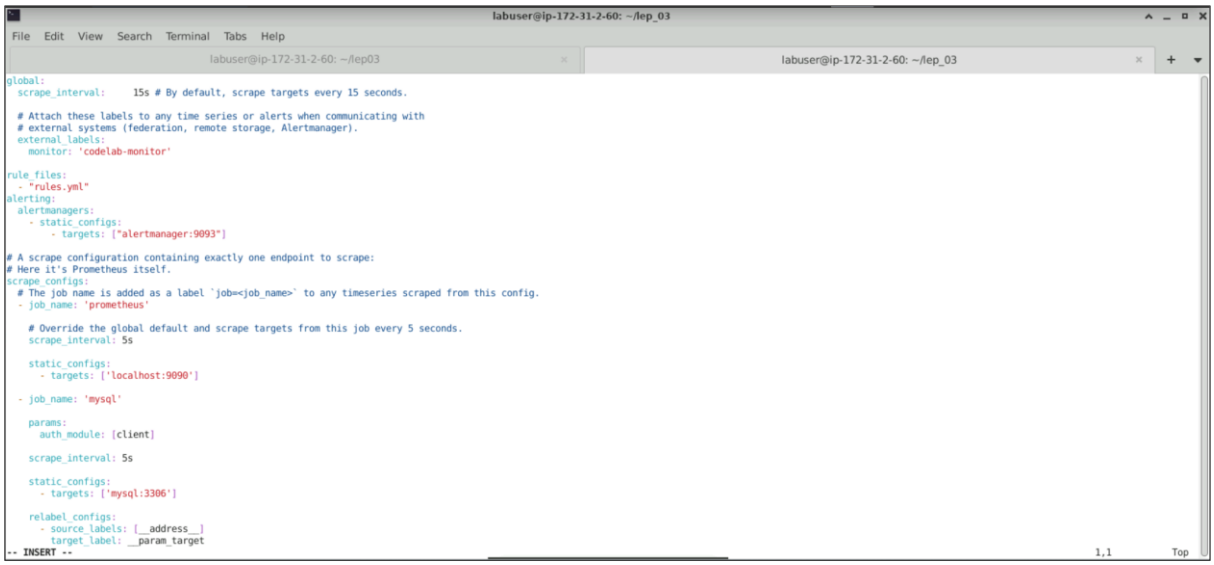
Step 4: Configure and start Prometheus in a Docker container

4.1 Run the following command to use **vim** editor to create the configuration file **prom.yml** for Prometheus:
sudo vim prom.yml



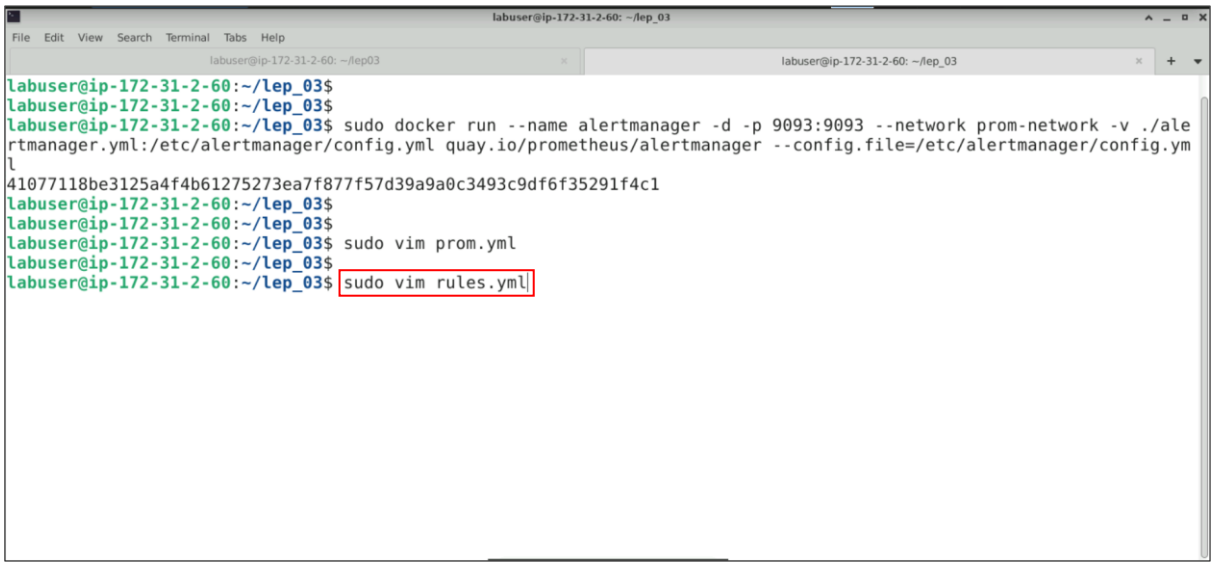
4.2 Copy and paste the following configuration, then save and exit the file:

```
global:
  scrape_interval:     15s # By default, scrape targets every 15 seconds.

  # Attach these labels to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
  external_labels:
    monitor: 'codelab-monitor'

rule_files:
  - "rules.yml"
alerting:
  alertmanagers:
    - static_configs:
        - targets: ["alertmanager:9093"]

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped
from this config.
  - job_name: 'prometheus'

    # Override the global default and scrape targets from this job every 5 seconds.
    scrape_interval: 5s

    static_configs:
      - targets: ['localhost:9090']

  - job_name: 'mysql'

    params:
      auth_module: [client]

    scrape_interval: 5s

    static_configs:
      - targets: ['mysql:3306']

    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
          # The mysqld_exporter host:port
        replacement: mysql-exporter:9104
```



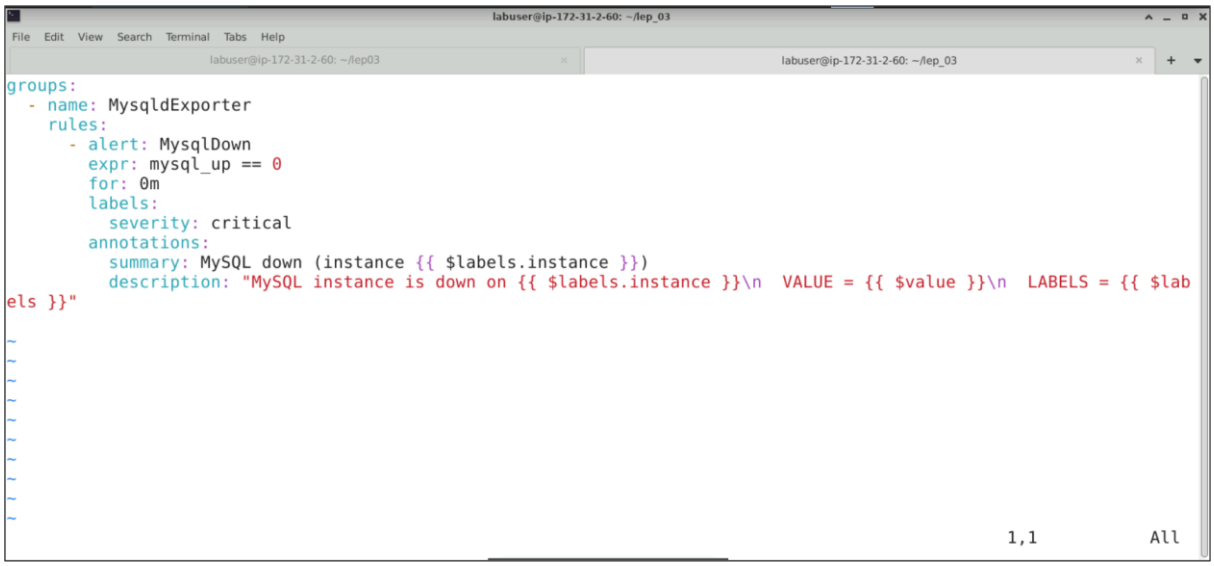
4.3 Run the following command to create the **rules.yml** configuration file using **vim** editor:

sudo vim rules.yml



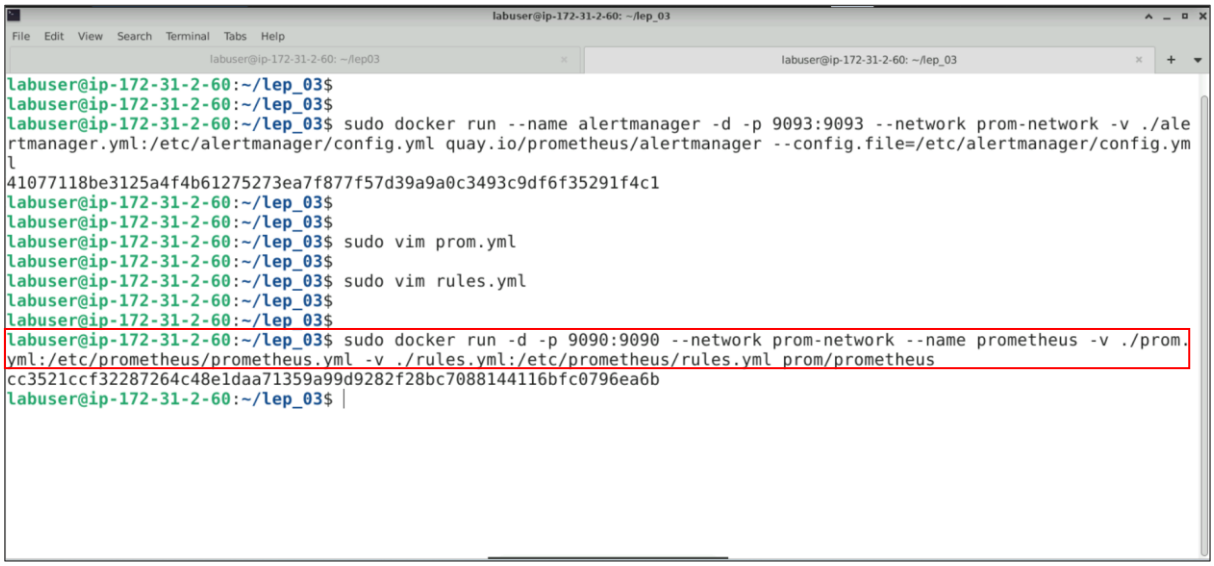
4.4 Copy and paste the following configuration, then save and exit the file:

```
groups:
  - name: MysqldExporter
    rules:
      - alert: MysqlDown
        expr: mysql_up == 0
        for: 0m
        labels:
          severity: critical
        annotations:
          summary: MySQL down (instance {{ $labels.instance }})
          description: "MySQL instance is down on {{ $labels.instance }}\n  VALUE = {{ $value }}\n  LABELS = {{ $labels }}"
```



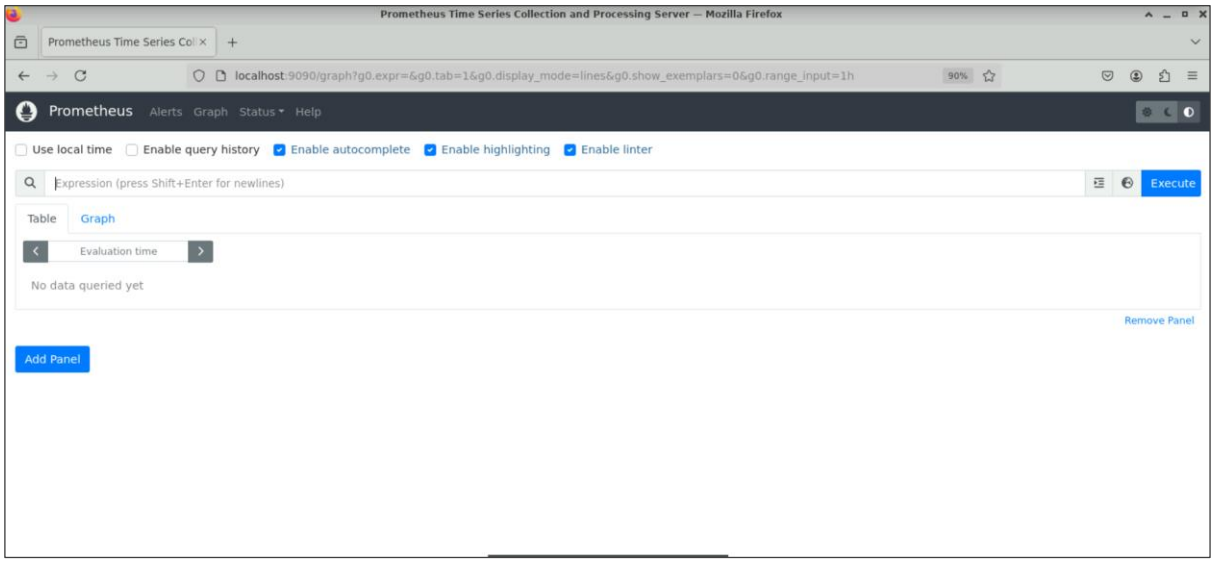
4.5 Execute the following command to run the container after both configuration files are created:

sudo docker run -d -p 9090:9090 --network prom-network --name prometheus -v ./prom.yml:/etc/prometheus/prometheus.yml -v ./rules.yml:/etc/prometheus/rules.yml prom/prometheus

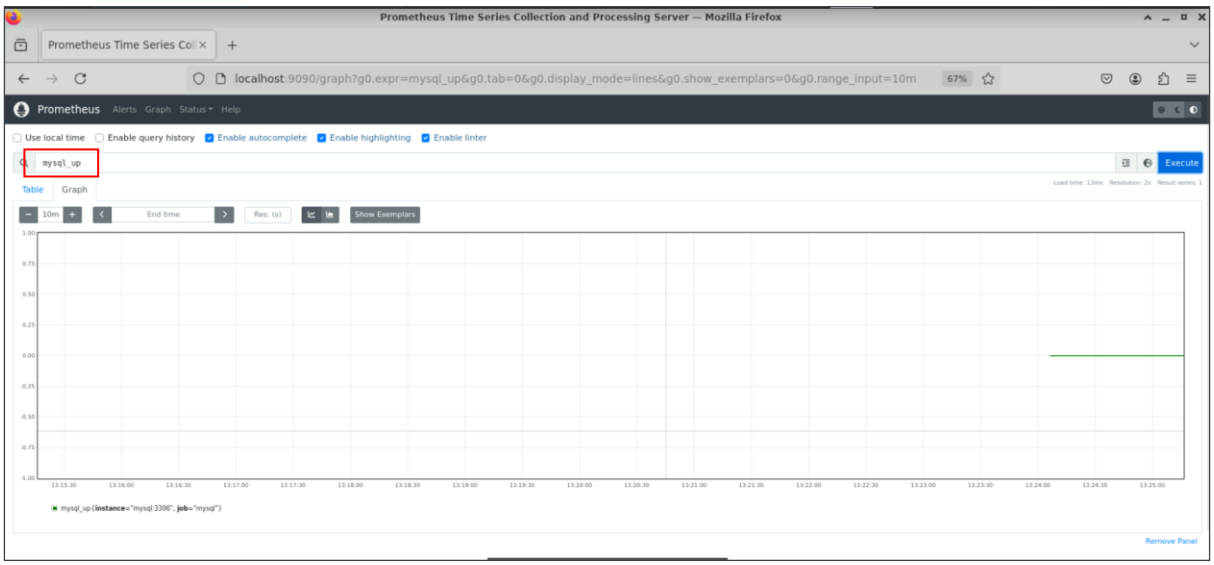


Step 5: Explore MySQL Exporter metrics and alerting config on Prometheus

5.1 Open the browser and access the Prometheus UI using the URL **http://localhost:9090/**

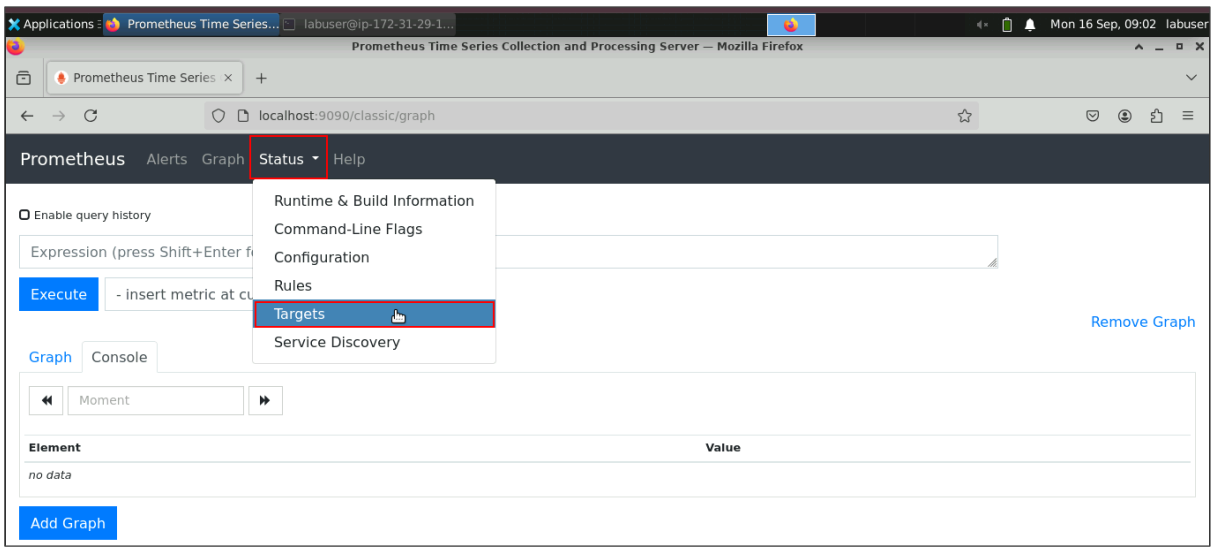


5.2 Execute the following query on the expression bar to visualize MySQL exporter metrics:
mysql_up

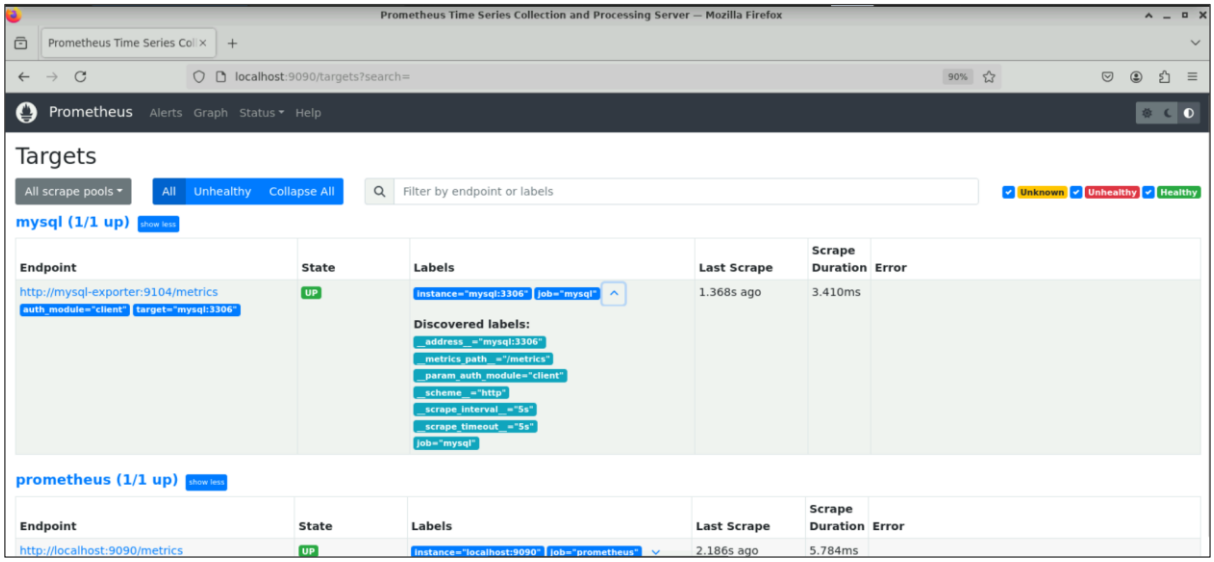


Step 6: Simulate MySQL failure

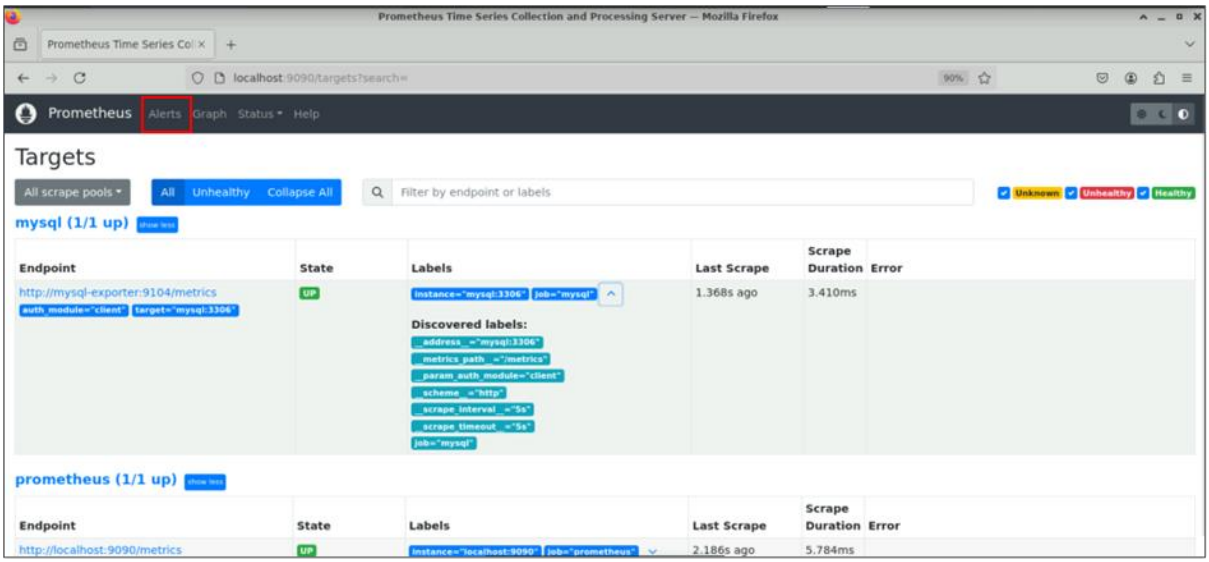
6.1 Navigate to the **Targets** section in the Prometheus UI and check the status of MySQL exporter

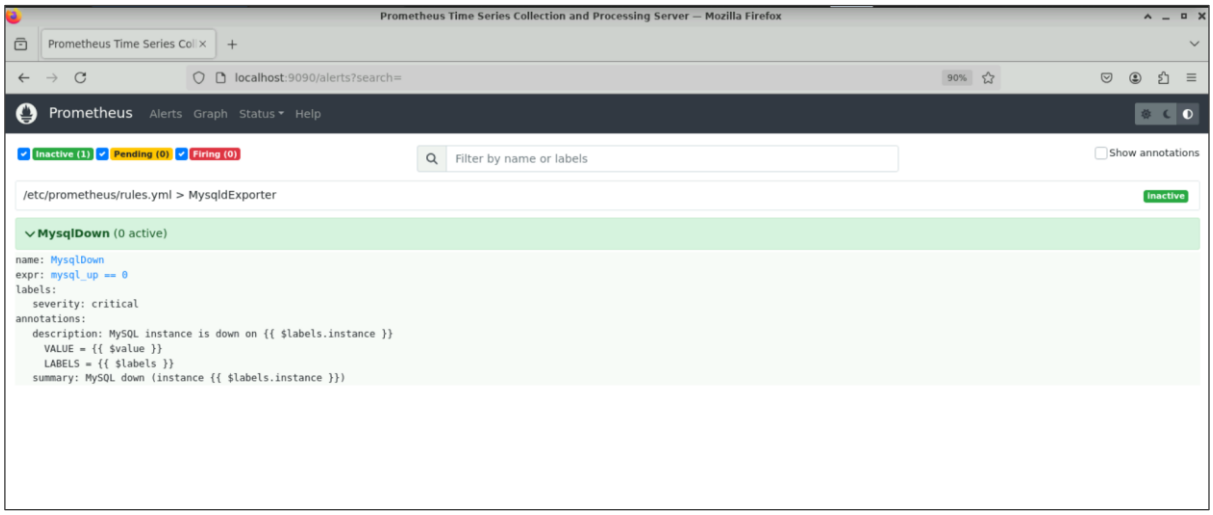


It appears as shown below:



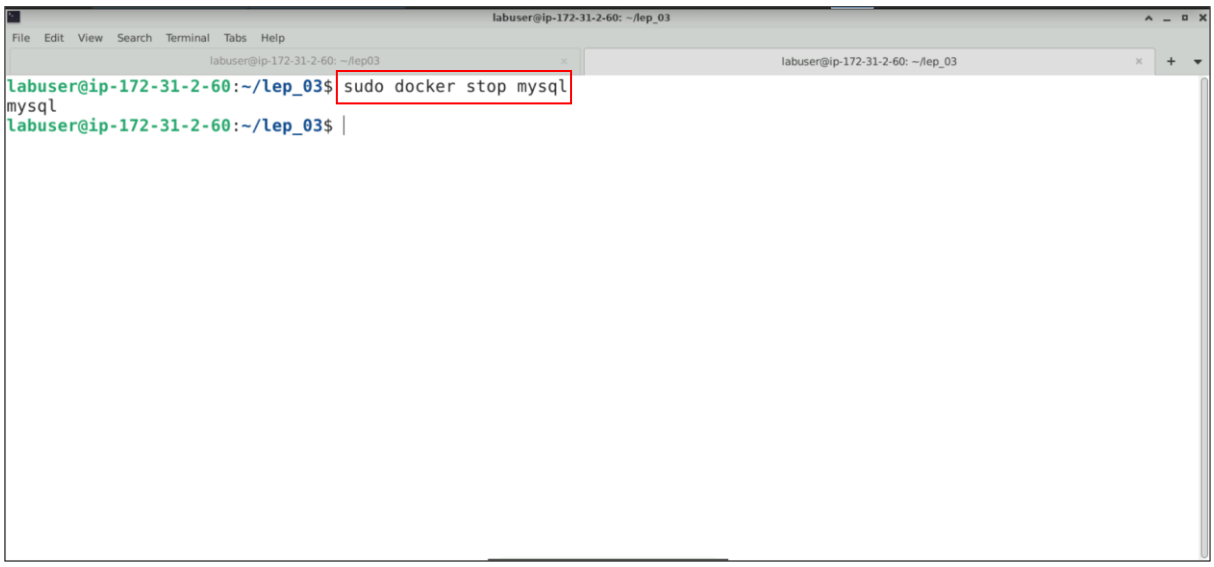
6.2 Navigate to the **Alerts** section and check the alert rule details



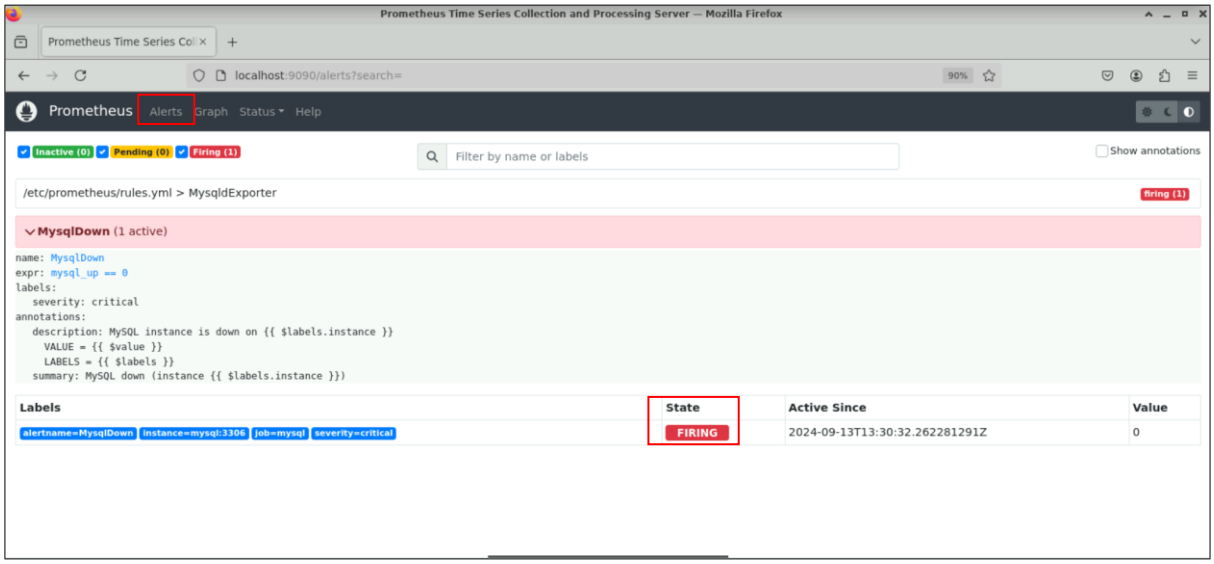


6.3 Navigate back to the terminal and run the following command to stop the MySQL container:

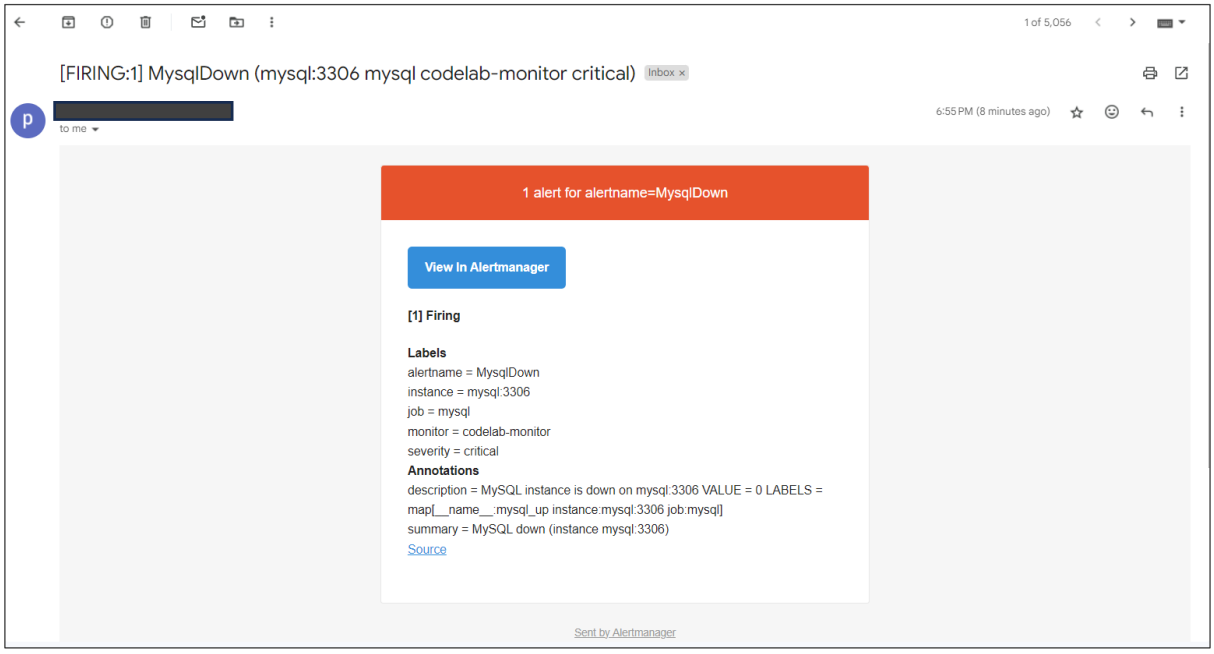
sudo docker stop mysql



6.4 Check the **Alerts** section in Prometheus for a MySQL alert trigger

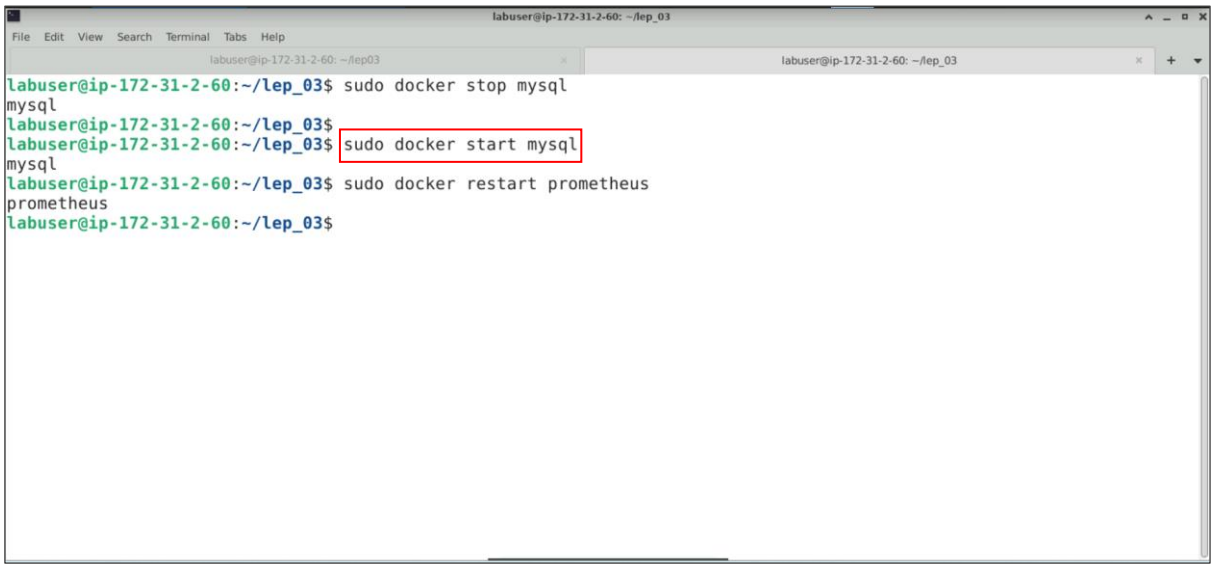


6.5 Check your email to verify that the alert notification has been received

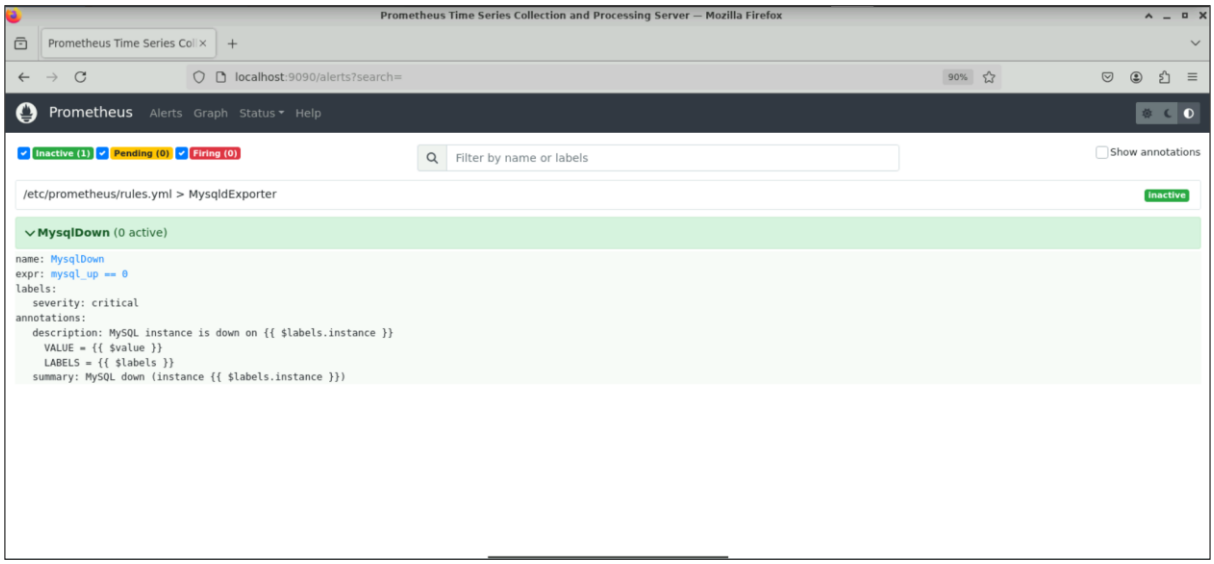


6.6 Run the following command to start the MySQL container to observe the alert status change:

sudo docker start mysql



Observe that the alert status has been updated in the Prometheus UI as shown below:



By following these steps, you have successfully configured Prometheus to monitor a MySQL database, collected key performance metrics using MySQL Exporter, and set up alerting rules with Alertmanager to notify of any issues, ensuring proactive database management and performance optimization.