Lab (2)

1-How do I trigger a Prometheus alert?

Create an alert rule defines the conditions that need to be met in order for an alert to be triggered and once you have created an alert rule, you can use the Prometheus Alertmanager to send alerts to your team. The Alertmanager can send alerts via email, Slack, PagerDuty, or any other notification system that you choose

2-What is the difference between node exporter and mysql exporter?

- **Node exporter** is a simple exporter that can be installed on any host. It collects a variety of metrics about the host, such as CPU usage, memory usage, disk usage, network usage, and process counts. Node exporter is easy to install and configure, and it does not require any dependencies.
- **MySQL exporter** is a more complex exporter that collects metrics about MySQL databases. It collects metrics about queries per second, connections, latency, and more. MySQL exporter requires a MySQL database to be installed, and it is more difficult to install and configure than node exporter.

3-What is the maximum retention period to save data in Prometheus and how to increase it ?

The maximum retention period to save data in Prometheus is 15 years. This is the default value, but it can be increased by setting the (storage.tsdb.retention) flag. The value of this flag is specified in seconds, so to increase the retention period to 10 years, you would set the flag to 3155760000.

4-What are the different PromQL data types available in Prometheus Expression language?

- Scalars

  These are the simplest type of data in PromQL. They are single values, such as integers, floating-point numbers, and strings.

  For example, the value 10 is a scalar.

- Vectors

  These are collections of scalars. They are typically used to represent time series data, such as metrics.

  For example, the vector [10, 20, 30] is a vector of three integers.

- Instant vectors

  These are vectors of scalars that are all at the same point in time. They are typically used to represent the current value of a metric.

  For example, the vector [10, 20, 30] is an instant vector if all three values are at the same time.

- Range vectors

  These are vectors of scalars that are spread out over a period of time. They are typically used to represent the history of a metric.

  For example, the vector [10, 20, 30] is a range vector if the three values are at different times.

5-How to calculate the average request duration over the last 5 minutes from a histogram?

To calculate the average request duration over the last 5 minutes from a histogram, you can use the following Prometheus expression:

```
avg(http_request_duration_seconds_sum[5m]) / avg(http_request_duration_seconds_count[5m])
```

This expression first calculates the sum of all the request durations in the last 5 minutes using the http_request_duration_seconds_sum metric. Then, it calculates the number of requests in the last 5 minutes using the http_request_duration_seconds_count metric. Finally, it divides the sum of the request durations by the number of requests to get the average request duration.

6-What is Thanos Prometheus?

Thanos is a set of components that can be composed into a highly available metric system with unlimited storage capacity, which can be added seamlessly on top of existing Prometheus deployments.
Thanos is a good choice for organizations that need a highly available, scalable, and cost-effective metric system. It is also a good choice for organizations that want to store Prometheus data indefinitely.

Here are some of the benefits of using Thanos:

High availability: Thanos can be used to create a highly available Prometheus system by replicating Prometheus data across multiple nodes. This ensures that your data is always available, even if one of the nodes fails.

Unlimited storage: Thanos can be used to store Prometheus data indefinitely, even if your Prometheus servers have limited storage capacity. This is because Thanos can store Prometheus data in object storage, such as AWS S3 or Google Cloud Storage.

Seamless integration with Prometheus: Thanos can be integrated with Prometheus with minimal changes to your existing Prometheus configuration. This makes it easy to add Thanos to your existing Prometheus deployment.

Cost-effective: Thanos is a cost-effective way to store Prometheus data. Thanos can store Prometheus data in object storage, which is typically much cheaper than storing data on Prometheus servers.

7-what is promtool and how I can use it?

Promtool is a command-line tool that can be used to interact with Prometheus. It can be used to do a variety of things, including:

Querying Prometheus: Promtool can be used to query Prometheus for metrics. This can be useful for debugging or for getting a quick overview of your system's health.

Validating Prometheus configuration files: Promtool can be used to validate Prometheus configuration files. This can help you to avoid errors when you start up Prometheus.

Generating Prometheus rules: Promtool can be used to generate Prometheus rules. This can help you to create alerts for your system.

Backfilling Prometheus data: Promtool can be used to backfill Prometheus data. This can be useful if you have recently added new data sources to Prometheus.

Here are some examples of how you can use promtool:

To query Prometheus for the average CPU usage of the last 5 minutes, you would use the following command:

```
promtool query 'avg(node_cpu_usage_seconds_total{job="my_server"}[5m])'
```

To validate a Prometheus configuration file, you would use the following command:

```
promtool check config my_prometheus.yaml
```

To generate Prometheus rules, you can use the following command:

```
promtool rules > my_rules.yaml
```

To backfill Prometheus data, you would use the following command:

```
promtool backfill my_prometheus.yaml my_data.csv
```

8-What types of Monitoring can be done via Grafana?

Grafana is a popular open-source tool that can be used to visualize time series data. It can be used to monitor a variety of systems, including:

Web applications: Grafana can be used to monitor the performance of web applications by tracking metrics such as request latency, response time, and error rate.

Databases: Grafana can be used to monitor the performance of databases by tracking metrics such as query volume, CPU usage, and memory usage.

Infrastructure: Grafana can be used to monitor the performance of infrastructure components such as servers, networks, and storage devices.

IoT devices: Grafana can be used to monitor the performance of IoT devices by tracking metrics such as sensor readings, actuator states, and network traffic.

Here are some of the specific types of monitoring that can be done via Grafana:

Health monitoring: Grafana can be used to monitor the health of systems by tracking metrics such as CPU usage, memory usage, and disk usage. This can help you to identify potential problems before they impact user experience.

Performance monitoring: Grafana can be used to monitor the performance of systems by tracking metrics such as request latency, response time, and error rate. This can help you to identify bottlenecks in your system and improve performance.

Availability monitoring: Grafana can be used to monitor the availability of systems by tracking metrics such as uptime and downtime. This can help you to ensure that your systems are always available to users.

Security monitoring: Grafana can be used to monitor the security of systems by tracking metrics such as authentication attempts, failed logins, and unauthorized access. This can help you to identify potential security breaches and take corrective action.

9-Can we see different Servers CPU comparison in Grafana?

Yes we can by doing this:
1- Create a new dashboard in Grafana.

2- Add a new panel to the dashboard.

3- Select the "Prometheus" data source.

4- In the "Query" field, enter the following query:

query = "sum(node_cpu_usage_seconds_total{job=~'my_server_.*'}) by (instance)"

5- In the "Time range" field, select the time range that you want to monitor.

6- Click the "Apply" button.