Prometheus Lab 2

1. **How do I trigger a Prometheus alert?**

To trigger a Prometheus alert, you need to define alert rules in the Prometheus configuration file, specify alerting channels to which the alerts should be sent, reload the Prometheus configuration, and monitor the alerting rules. Alert rules are defined using the alert expression type, which specifies a condition that must be met for an alert to be triggered, and alerting channels define how alerts should be sent to the user. Prometheus continuously evaluates alerting rules and generates alerts when the conditions are met.

2. **What is the difference between node exporter and MySQL exporter?**

Node Exporter:
Node Exporter is an exporter for machine metrics, such as CPU usage, memory usage, disk usage, and network activity. It runs on the machine that it monitors and collects various system-level metrics. Node Exporter is commonly used to monitor Linux/Unix servers, as well as Windows machines.

MySQL Exporter:
MySQL Exporter is an exporter for MySQL database metrics, such as number of queries, query execution time, and database locks. It collects metrics from a running MySQL server and exports them in a Prometheus-compatible format. MySQL Exporter is commonly used to monitor MySQL database servers.

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 determined by the storage.tsdb.retention.time parameter in the configuration file. By default, Prometheus retains data for 15 days, but this can be modified by changing the value of the parameter to the desired duration, and restarting the Prometheus server. It's important to note that increasing the retention period will also increase the disk space required and may impact query performance, so it's recommended to carefully consider the trade-offs before making changes to the retention time.

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

- Scalars: Scalars are used to represent numeric data. There are two types of scalars in PromQL: integers and floating-point numbers. Integers are used to represent whole numbers, while floating-point numbers are used to represent decimal values.
- Strings: Strings are used to represent text data in PromQL. Strings are enclosed in double quotes and can contain any sequence of characters.
- Time series: Time series are used to represent a sequence of data points over time. Each data point in a time series has a timestamp and a value. Time series are identified by a metric name and a set of key-value pairs called labels.
- Vectors: Vectors are used to represent a set of time series that share the same metric name and label set. Vectors are the primary data type used in PromQL queries and are used to select and aggregate time series data.

- Instant vectors: Instant vectors represent a set of time series at a specific point in time, rather than over a range of time. Instant vectors are obtained by specifying a timestamp in a PromQL query.

5. How To calculate the average request duration over the last 5 minutes from a histogram?

avg(avg_over_time(my_histogram_bucket[5m]))

6. What is Thanos Prometheus?

Thanos is an open-source platform that extends the capabilities of Prometheus. It provides long-term storage and high availability for Prometheus metrics, allowing you to scale Prometheus horizontally and store metrics data in a distributed object storage system. Thanos also enables cross-cluster federation and advanced query capabilities, making it a powerful tool for managing and scaling Prometheus over long periods of time.

7. What types of Monitoring can be done via Grafana?

- Infrastructure monitoring: Grafana can be used to monitor the health and performance of servers, databases, networks, and other infrastructure components. It supports a wide range of data sources, including Prometheus, InfluxDB, Graphite, Elasticsearch, and more.
- Application monitoring: Grafana can be used to monitor the performance and availability of applications, including web applications, microservices, and containerized applications. It supports integration with popular application performance monitoring (APM) tools like New Relic and AppDynamics.
- IoT monitoring: Grafana can be used to monitor and visualize data from IoT devices, including sensors, actuators, and other connected devices. It supports integration with popular IoT platforms like MQTT, InfluxDB, and Azure IoT Hub.
- Business intelligence: Grafana can be used to create and display business intelligence dashboards, providing real-time insights into key performance indicators (KPIs) and other business metrics.
- Log analysis: Grafana can be used to analyze logs from various sources, including system logs, application logs, and security logs. It supports integration with popular log analysis tools like Loki and Elasticsearch.

8. Can we see different Servers' CPU comparisons in Grafana?

Yes, you can see different servers' CPU usage comparisons in Grafana by using a data source that provides CPU metrics for each server, such as Prometheus, InfluxDB, or Graphite. You can use various visualization techniques like Singlestat panel, Graph panel, Bar gauge panel, or Table panel to compare CPU usage values across different servers. Overall, Grafana provides many options to visualize and compare CPU usage values depending on your specific needs and preferences.