**1-what is differnect http status code and explain meaning of each of them ?**

HTTP status codes are three-digit numbers that indicate the status of a Hypertext Transfer Protocol (HTTP) request made by a client (e.g., a web browser or API client) to a server. Each status code conveys a specific meaning, allowing the client to understand the outcome of the request. Here are some common HTTP status codes and their meanings:

1xx Informational Responses: These status codes indicate that the request has been received and the server is continuing to process it. They are used for informational purposes and not typically encountered in day-to-day usage.

* 100 Continue: The server has received the initial part of the request, and the client should proceed with the rest of the request.
* 101 Switching Protocols: The server is changing the protocol being used on this connection, such as upgrading from HTTP/1.1 to WebSocket.

2xx Success: These status codes indicate that the request was successfully received, understood, and accepted by the server.

* 200 OK: The request was successful, and the server has returned the requested data as part of the response.
* 201 Created: The request was successful, and a new resource has been created as a result (e.g., after a successful POST request).
* 204 No Content: The request was successful, but there is no data to return. This is often used for successful DELETE requests.

3xx Redirection: These status codes indicate that further action needs to be taken by the client to complete the request.

* 301 Moved Permanently: The requested resource has permanently moved to a new URL. The client should update its bookmarks or links to the new URL.
* 302 Found: The requested resource is temporarily available at a different URL. The client should continue using the original URL for future requests.
* 304 Not Modified: The client's cached version of the requested resource is still valid, and there is no need to re-download it from the server.

4xx Client Errors: These status codes indicate that there was a client-side error, such as a malformed request or insufficient permissions.

* 400 Bad Request: The server cannot understand the request due to a client error, such as malformed syntax or invalid request parameters.
* 401 Unauthorized: The client must authenticate itself to get the requested response. Typically used for authentication failures.
* 403 Forbidden: The client does not have the necessary permissions to access the requested resource.
* 404 Not Found: The server could not find the requested resource.

5xx Server Errors: These status codes indicate that there was an error on the server-side while processing the request.

* 500 Internal Server Error: The server has encountered a situation it doesn't know how to handle. This is a generic error message for any unhandled server-side error.
* 502 Bad Gateway: The server, acting as a gateway or proxy, received an invalid response from the upstream server.
* 503 Service Unavailable: The server is not ready to handle the request. It might be undergoing maintenance or overloaded.

These are some of the most commonly encountered HTTP status codes, but there are many others that can be used for specific scenarios and extensions. Properly interpreting and handling HTTP status codes is essential for effective communication between clients and servers in web applications and APIs.

**2-What database is used by Prometheus?**

Prometheus uses its custom time-series database called "TSDB" (Time Series Database). This database is purpose-built to store and handle time-series data, which is the core data format used for monitoring metrics in Prometheus.

The TSDB in Prometheus organizes data into time series, where each time series represents a sequence of timestamped data points. Each data point contains a timestamp and a corresponding value. Time series are identified by a unique combination of metric names and a set of key-value pairs called labels. This design allows Prometheus to efficiently store and query large volumes of time-series data for monitoring and alerting purposes.

The use of a specialized time-series database enables Prometheus to efficiently handle metric data over time, support queries for different time ranges, calculate aggregates, and perform various types of analysis on the collected metrics. The combination of Prometheus's querying capabilities and its custom TSDB is what makes it a powerful tool for monitoring and observability in modern distributed systems and cloud-native environments.

**3-what is the differnece between different metrics types ( counter , gauge , histogram) ?**

The different metric types in Prometheus (counter, gauge, and histogram) serve distinct purposes and capture different aspects of the monitored system. Each type is designed to handle specific kinds of metrics and provides valuable insights into the behavior and performance of the system being monitored.

1. Counter: A counter is a cumulative metric that represents a monotonically increasing value. It is used to track the number of occurrences of an event or the total count of a specific activity over time. Counters are typically used to measure things like the number of requests received, the total number of errors encountered, or the total number of tasks executed.

Important characteristics of counters:

* They only go up, never down.
* They reset to zero when the Prometheus server or the target being monitored restarts.

1. Gauge: A gauge is a metric that represents a single numerical value that can go up or down over time. It is used to track the current state of a value that can fluctuate, such as the number of active connections, the amount of free memory, or the current temperature. Unlike counters, gauges do not represent a cumulative quantity.

Important characteristics of gauges:

* They can increase or decrease as the value being measured changes.
* They retain their last observed value until a new value is provided.

1. Histogram: A histogram is used to sample and record observations of a value over time, allowing for the analysis of the distribution of values. Histograms are typically used to track things like request durations, response sizes, or any other value that varies and needs to be summarized into buckets.

Important characteristics of histograms:

* They count how many observations fall into predefined buckets or buckets of varying widths.
* They expose quantiles (e.g., 90th percentile, 99th percentile) to understand data distribution.

In the example above, **http\_request\_duration\_seconds\_bucket** is a histogram representing the number of requests falling into predefined time buckets (e.g., 0.1 seconds).

In summary, counters are used to measure cumulative counts, gauges track instantaneous values that can increase or decrease, and histograms provide insights into the distribution of values over time. Properly choosing the appropriate metric type is crucial for effective monitoring and analysis of the system's behavior.

**4-install prometheus on your localhost or on server in any cloud provider ?**

**5-add any new target to prometheus.yaml file and apply any query on it using promql langauge ?**