Design of a Monitoring System

Learn about the initial design of a generic monitoring system.

We'll cover the following

Requirements

Let’s sum up what we want our monitoring system to do for us:

- Monitor critical local processes on a server for crashes.
- Monitor any anomalies in the use of CPU/memory/disk/network bandwidth by a process on a server.
- Monitor overall server health, such as CPU, memory, disk, network bandwidth, average load, and so on.
- Monitor hardware component faults on a server, such as memory failures, failing or slowing disk, and so on.
- Monitor the server’s ability to reach out-of-server critical services, such as network file systems and so on.
- Monitor all network switches, load balancers, and any other specialized hardware inside a data center.
- Monitor power consumption at the server, rack, and data center level.
- Monitor any power events on the servers, racks, and data center.

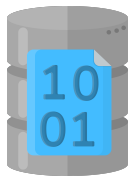
- Monitor routing information and DNS for external clients.
- Monitor network links and paths' latency inside and across the data centers.
- Monitor network status at the peering points.
- Monitor overall service health that might span multiple data centers—for example, a CDN and its performance.

We want automated monitoring that identifies an anomaly in the system and informs the alert manager or shows the progress on a dashboard. Cloud service providers provide a health status of their services:

- AWS: https://health.aws.amazon.com/health/status
- Azure: https://status.azure.com/en-us/status
- Google: https://status.cloud.google.com/

Building block we will use

The design of distributed monitoring will consist of the following building block:



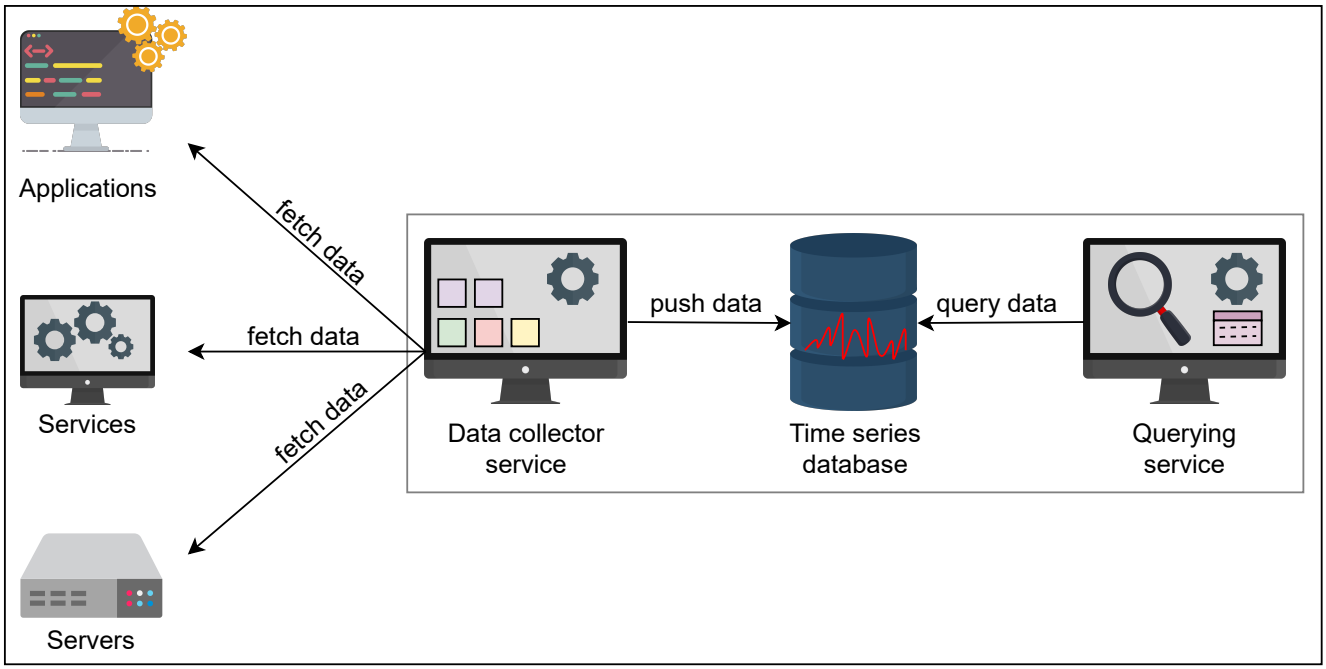
Blob storage

Blob storage: We'll use blob storage to store our information about metrics.

High-level design

The high-level components of our monitoring service are the following:

- **Storage**: A time-series database stores metrics data, such as the current CPU use or the number of exceptions in an application.
- **Data collector service**: This fetches the relevant data from each servi and saves it in the storage.
- **Querying service**: This is an API that can query on the time-series database and return the relevant information.



High-level design of a monitoring system

Let’s dive deep into the components mentioned above in the next lesson.

Back
Prerequisites of a Monitoring System
Next
Detailed Design of a Monitoring System
Mark as Completed