Recitation 7

Monitoring

Shreyans Sheth
July 4th, 2020

Why do you need monitoring ?

- Suppose you
  - Receive complaints about your service giving users unexpected errors...
  - Your application response is extremely slow despite enough resources…
  - A component of your system doesn’t get triggered despite repeated fixes...

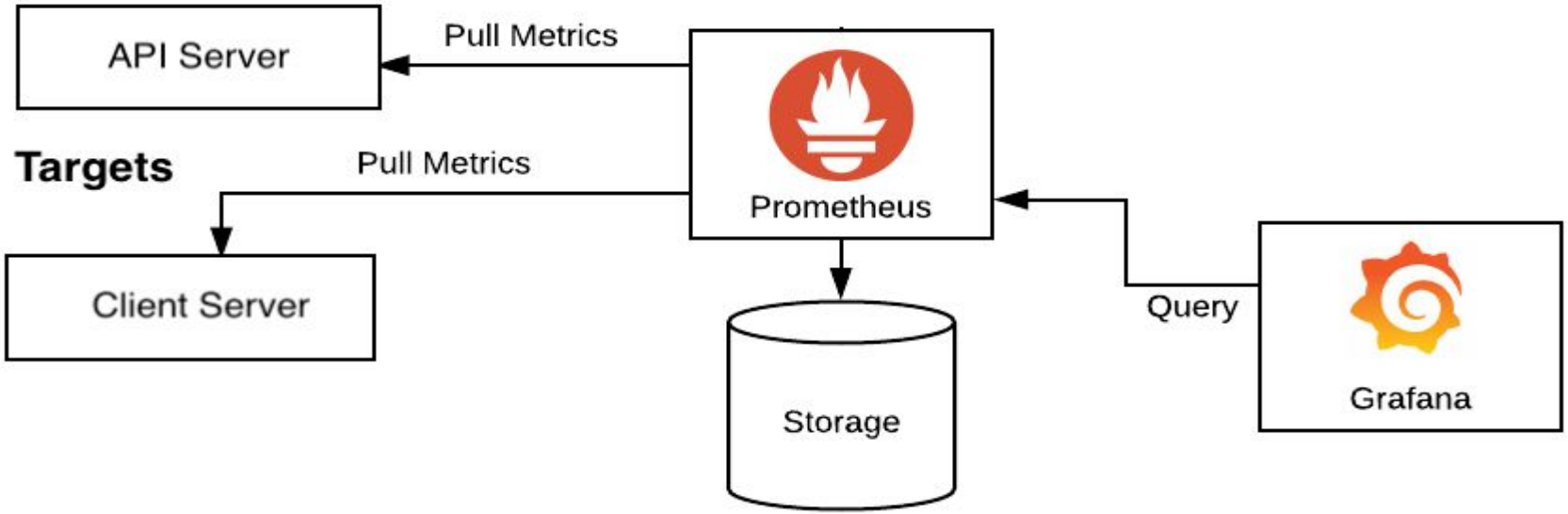
How can you address this ?

Metrics captured with monitoring help us understand the behavior of our application, identify unexpected behavior, and scale up or down the underlying software infrastructure.

It helps ensure the best possible experience for the application’s consumers, as well as utilize the system infrastructure judiciously.

Prometheus + Grafana - Modern Monitoring Stack

- **Prometheus** - A timeseries DB that PULLS and stores metrics from your app.
- **Grafana** - Reads these metrics from prometheus using PromQL and helps us visualize them as dashboards.



Overview - Python webapp monitoring tutorial

- We will now see, how we can instrument a simple python webapp to see it's metrics such as **request count** and **request latency** on prometheus

Here is the overview:

1. Explain webapp structure
   a. How we attach metrics via middleware
   b. The metrics endpoint
2. Prometheus configuration
3. Docker compose setup with the app and monitoring
4. Script that sends random requests to the running app

Queries - Python webapp monitoring tutorial

- ***request_count_total{http_status="500"}***
  - Shows us the count of queries that have add status 500

- ***rate(request_latency_seconds_count[1h])***
  - Shows us the request latency over time

Prometheus provides several metric types and ways to query them for advanced usecases (beyond current scope). We can experiment with them in the next exercise. See Concepts - Metric Types, Query-examples

Exercise - Gather data about your app!

Refer to Prometheus documentation and derive insights into any 3 metrics for your team by writing PromQL queries.

- Head over to http://17645-teachers.isri.cmu.edu:9090/ .
- This prometheus instance captures data for the movie simulator. You can get data from the movielog stream about your application here.
- For example, when you type in
  - ***rate(seaisim_recommendation_request_counter{team="movielog 1"}[5m])***

  And set the time window to 24 hours, you can see how your team 1's application responds successfully and unsuccessfully to requests in that time frame

Thanks!

References

1. Rollout.io - Monitoring your python application
2. Monitor your django application
3. Prometheus docs