Monitoring

AI Engineering - Recitation 8

Monitoring

- Responsibility of a development team doesn’t end with code deployment to production
- Use cases (after deployment)
  - Track how the software is performing in production
  - Make decisions based on live metrics
  - Quickly inform developers / operations team of undesirable situations
  - Quickly respond to situations

Prometheus + Grafana

- Prometheus
  - A time series database that stores metrics from your application
  - It has client libraries that let you create and expose metrics
  - Metrics can be pulled by Prometheus from your app, or your app can push them to Prometheus

- Grafana
  - A visualization tool (dashboards, charts etc.)
  - Where to get the data to visualize - data sources (Prometheus, PostgreSQL etc.)
  - What do we want to visualize - write PromQL queries to configure

Prometheus + Grafana Architecture

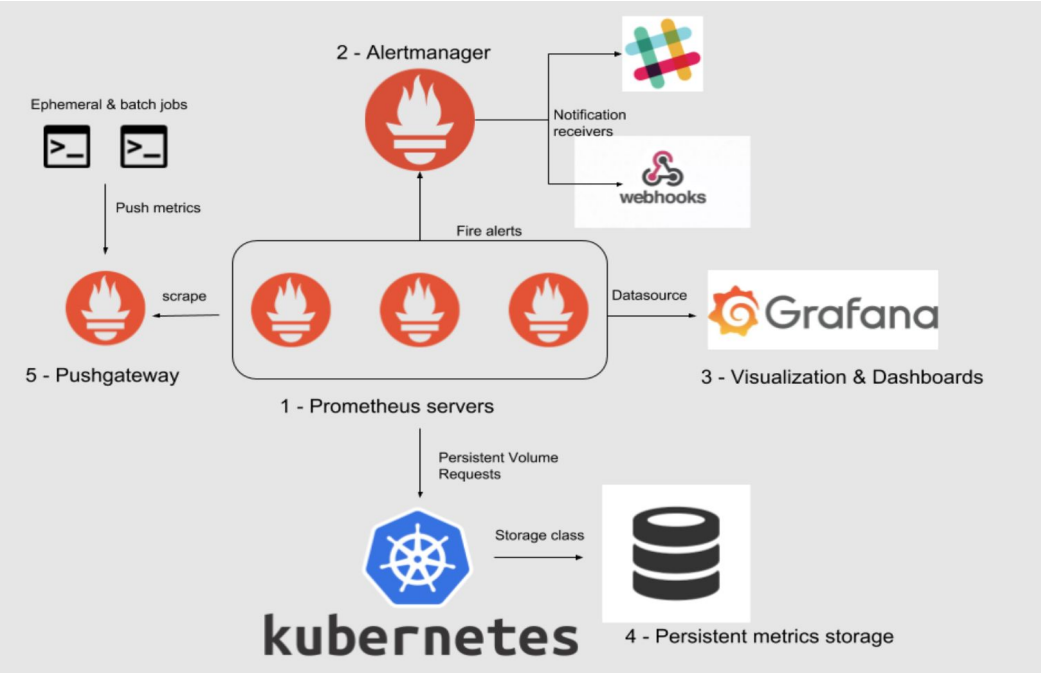

Image credit: Sysdig

Demo

- Monitor Kafka events
  - Export some metrics from Kafka using Prometheus client (Pull model)
- Run Prometheus and Grafana as containers
- Access Prometheus UI
- Build a dashboard in Grafana by connecting to Prometheus

Resources:

- Prometheus client for Python: https://github.com/prometheus/client_python
- https://neilkillen.com/2020/05/30/monitoring-sitecore-container-environment-with-prometheus

Push vs Pull

- Pull
  - Expose metrics from your application via an API, and let Prometheus poll it
  - Typically used when your app is going to be a long running process
- Push
  - Push metrics from your application / CI pipeline to a “push gateway” app
  - You need to run push gateway as a separate container in your VM
  - Prometheus will poll the push gateway to fetch metrics
  - Typically used when
    - You have a short-lived process
    - The metrics aren’t going to change that often

Common Metric Types

- Counter - Value can only be increased or reset to zero
  - Number of requests, errors, tasks completed
- Gauge - Value can go up or down
  - Number of concurrent requests, running containers

Resources:

- https://prometheus.io/docs/concepts/metric_types/
- https://tomgregory.com/the-four-types-of-prometheus-metrics/

PromQL - Examples

- request_count_total{http_status=”200”}
  - Shows the count of requests that have status 200

- rate(request_latency_seconds_count[1h])
  - Shows the request latency over 1 hour

Resources:

- https://prometheus.io/docs/prometheus/latest/querying/basics/

Additional Resources

- Prometheus: https://prometheus.io/docs/introduction/overview/
- Prometheus Latest: https://prometheus.io/docs/prometheus/latest/migration/
- Node Exporter: https://github.com/prometheus/node_exporter
- Alert Manager: https://prometheus.io/docs/alerting/latest/overview/
- Prometheus Best Practices: https://prometheus.io/docs/practices/
- Grafana: https://grafana.com/docs/