Kubernetes from Basic to Advanced

kubernetes

Session #10
Monitoring

kubernetes

Session Contents

- Use Kubectl
- Kubernetes Dashboard
- Log management
- Prometheus & Grafana

kubernetes

Use Kubectl

Kubectl for monitoring

- Many times, kubectl can be the first (and only) tool available to access the cluster
- Cannot be used to have a proactive approach to monitoring but can be used to get detailed understanding about cluster issues
- With the needed permissions you can have complete understanding about cluster behavior

kubernetes

Kubectl describe

```
kubectl describe pod <pod> [-n <namespace>]
```

- Shows details about pod
  - Metadata
  - Network
- Lists all events occurred during pod lifecycle
- First place to go when pod don't have "Running" status

kubernetes

Kubectl logs

kubectl logs <pod> [-n <namespace>]

- Shows pod stdout and stderr
- Flag –f blocks the console and show new lines

kubernetes

Kubectl port-forward

```
kubectl port-forward pod <pod> [-n <ns>] hostport:podPort
kubectl port-forward svc <svc> [-n <ns>] hostport:podPort
```

- Maps a port on machine with pod port
- Allow to make direct requests
- When using service, maps directly to only on container (no load balancing)

kubernetes

Kubectl top

```
kubectl top node <node>
kubectl top pod <pod> [-n <ns>]
```

- Display resource (CPU/memory) usage of the resources (nodes or pods)
- Due to the metrics pipeline delay, they may be unavailable for a few minutes since pod creation

kubernetes

Kubernetes Dashboard

Kubernetes Dashboard

- Web-based Kubernetes user interface to have a more user-friendly way to look into your cluster.
- Kubernetes Dashboard were created and is maintained by Kubernetes Community
- Initially, was the only Web-based tool to monitor your cluster
- Now, is not so used on production environment
  - Newer and better tools arrive on Kubernetes Landscape
  - Limitation on metrics since it uses only Kubernetes Vanilla metrics
- You can use Dashboard to deploy containerized applications to a Kubernetes cluster, troubleshoot your containerized application, and manage the cluster resources.

kubernetes

Kubernetes Dashboard

- You can use Dashboard to:
  - Monitor your cluster resources
  - Manage Kubernetes resources
  - Get an overview of applications running on your cluster
  - Troubleshoot your containerized application
  - Deploy containerized applications to a Kubernetes cluster
- Provide wizards to scale a Deployment, initiate a rolling update, restart a pod or deploy new applications using a deploy wizard.

kubernetes

Demo | Kubernetes Dashboard

Log Management

Motivation

- Containers runs on top of an ephemeral layer that is deleted each time a pod is deleted
- If you write your logs to a file in this layer you may lose them
- Even during execution can be hard to reach them
- How to have access to these logs and keep them for as long as needed?

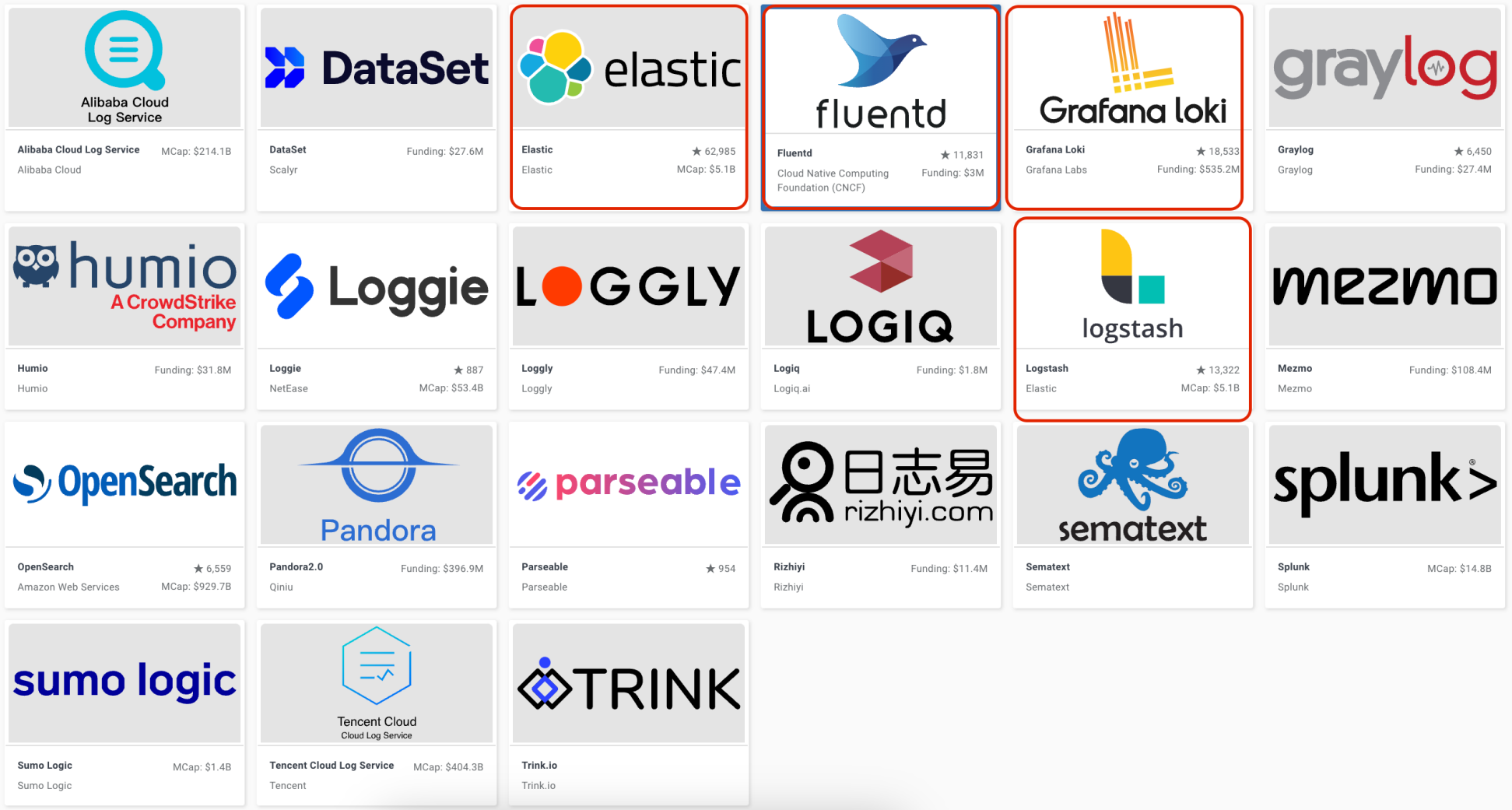
kubernetes

Logs on Kubernetes

- As a best practice, everything that needs to be logged should be write to standard output or standard error of each container
- Log Management tools for Kubernetes uses a concept of creating a DaemonSet to have a pod in each node to access to those streams
- Then, after collecting the data, send them to a centralized server where the logs are kept for as long as needed
- Finally, the full solution have a visualization layer where logs can be queried and accessed from outside of the cluster

kubernetes

CNCF Logging



bernetes

Prometheus & Grafana

Motivation

- Kubernetes grant basic metrics about pods (memory, cpu)
- Those metrics are not sufficient when you want to have a better monitorization from your cluster and your applications
- Not only you need to gather new metrics, but you need a better visualization for them
- Dynamic and sharable dashboards are crucial for an efficient and proactive monitorization of any system and infrastructure

kubernetes

Prometheus

- Prometheus an open-source systems monitoring and alerting toolkit originally built at SoundCloud.
- It is now a standalone open source project and maintained independently of any company.
- Prometheus joined the Cloud Native Computing Foundation in 2016 as the second hosted project, after Kubernetes.
- Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels.

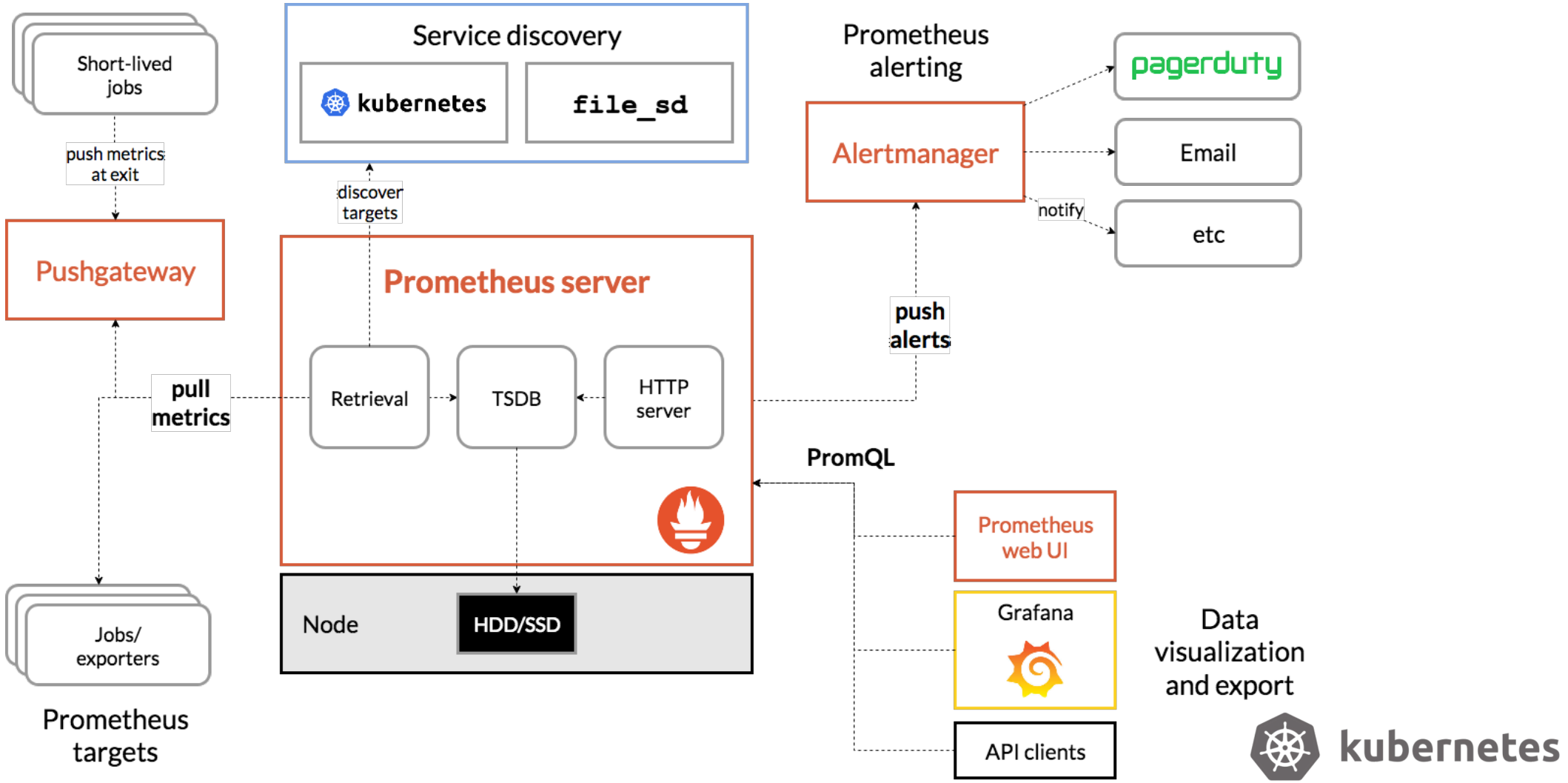
kubernetes

Grafana

- Prometheus's main features are:
  - a multi-dimensional data model with time series data identified by metric name and key/value pairs
  - PromQL, a flexible query language to leverage this dimensionality
  - no reliance on distributed storage; single server nodes are autonomous
  - time series collection happens via a pull model over HTTP
  - pushing time series is supported via an intermediary gateway
  - targets are discovered via service discovery or static configuration
  - multiple modes of graphing and dashboarding support

kubernetes

Prometheus Architecture



Prometheus Components

- Prometheus server which scrapes and stores time series data
- Client libraries for instrumenting application code
- Push gateway for supporting short-lived jobs
- Alertmanager to handle alerts
- Visualization tools are external but integration with Grafana is natural

kubernetes

Grafana

- Grafana allows you to query, visualize, alert on and understand your metrics no matter where they are stored.
- Allow you to unify your data from several sources and make interactive dashboards
- Have a great linkage with Prometheus using PromQL to create dashboards
- Dashboards are described JSON what made really easy to share between the community

kubernetes

Demo | Prometheus & Grafan

Questions?

kubernetes