



Review 4

Logging and Monitoring



Business Training

1

Session 4 - Review

Types of Kubernetes Logging

- Within a Kubernetes system, we can name three types of logs:
 - **Container logs** are logs generated by the containerized applications.
 - **Node logs** are collected by the Kubelet running on each Kubernetes node from the stdout and stderr of each running pod . They are combined them into a log file that is managed by Kubernetes.
 - **Cluster logs** refer to Kubernetes itself and all of its system component logs, and we can differentiate between components that run in a container and components that do not run in a container.

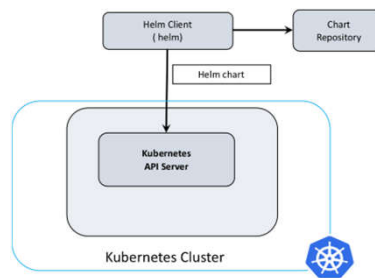
M.Romdhani, 2020

2

2

What is Helm?

- Helm is a **package manager for Kubernetes** that allows developers and operators to more easily package, configure, and deploy applications and services onto Kubernetes clusters.
- There are three basic concepts to understand.
 - A **Chart** is a Helm package.
 - A **Repository** is the place where charts can be collected and shared.
 - A **Release** is an instance of a chart running in a Kubernetes cluster



M.Romdhani, 2020

3

3

Kubernetes Metrics API Server

- Metric server collects metrics such as CPU and Memory by each pod and node from the Summary API, exposed by Kubelet on each node.
- Metrics Server registered in the main API server through Kubernetes aggregator, which was introduced in Kubernetes 1.7
- Used with horizontal pod autoscaler, kube-dashboard, kubectl etc

M.Romdhani, 2020

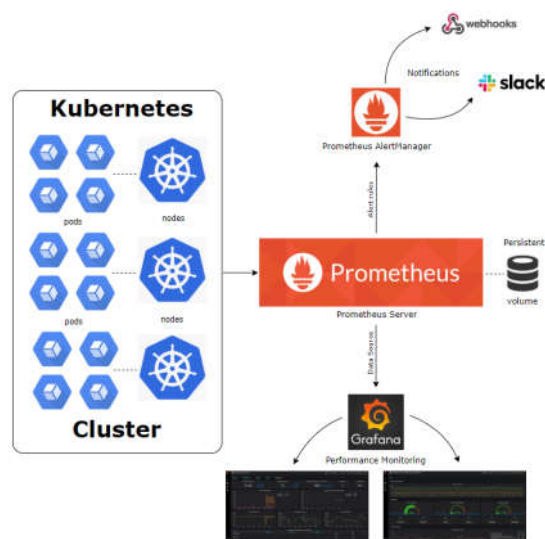
4

4

What is Prometheus ? What is Grafana ?

- **Prometheus is an open-source monitoring system that was originally built by SoundCloud. It consists of the following core components -**
 - A **data scraper** that pulls metrics data over HTTP periodically at a configured interval.
 - A **time-series database** to store all the metrics data.
 - A **simple user interface** where you can visualize, query, and monitor all the metrics.
- **Grafana** allows you to bring data from various data sources like Elasticsearch, Prometheus, Graphite, InfluxDB etc, and visualize them with beautiful graphs.
 - It also lets you set alert rules based on your metrics data. When an alert changes state, it can notify you over email, slack, or various other channels. Prometheus dashboard also has simple graphs. But Grafana's graphs are way better.

Prometheus integration in Kubernetes



Session 4 - Review

Debugging Pods

- **The first step in debugging a Pod is taking a look at it. Check the current state of the Pod and recent events with the following command:**
 - `kubectl describe pods ${POD_NAME}`
- **Look at the state of the containers in the pod. Are they all Running? Have there been recent restarts?**
 - Continue debugging depending on the state of the pods.
- **My pod stays pending**
 - If a Pod is stuck in Pending it means that it can not be scheduled onto a node. Generally this is because there are **insufficient resources** of one type or another that prevent scheduling. Look at the output of the **kubectl describe ... command** above. There should be messages from the scheduler about why it can not schedule your pod.
- **My pod stays waiting**
 - If a Pod is stuck in the Waiting state, then it has been scheduled to a worker node, but it can't run on that machine. Again, the information from `kubectl describe ...` should be informative. The most common cause of Waiting pods is a failure to pull the image.

M.Romdhani, 2020
7

7

Session 4 - Review

Debugging Pods

- **My pod is crashing or otherwise unhealthy**
 - Once your pod has been scheduled, the methods described in Debug Running Pods are available for debugging.
 - First, look at the logs of the affected container:
`kubectl logs ${POD_NAME} ${CONTAINER_NAME}`
 - If your container has previously crashed, you can access the previous container's crash log with:
`kubectl logs --previous ${POD_NAME} ${CONTAINER_NAME}`
- **My pod is running but not doing what I told it to do**
 - If your pod is not behaving as you expected, it may be that there was an error in your pod description (e.g. `mypod.yaml` file on your local machine), and that the error was silently ignored when you created the pod.
 - Often a section of the pod description is **nested incorrectly**, or a **key name is typed incorrectly**, and so the key is ignored. For example, if you misspelled command as **commnd** then the pod will be created but will not use the command line you intended it to use.
 - The first thing to do is to delete your pod and try creating it again with the `--validate` option. For example, run `kubectl apply --validate -f mypod.yaml`. If you misspelled command as `commnd` then will give an error.
 - The next thing to check is whether the pod on the apiserver matches the pod you meant to create (e.g. in a `yaml` file on your local machine). For example, run `kubectl get pods/mypod -o yaml > mypod-on-apiserver.yaml` and then manually compare the descriptions..

M.Romdhani, 2020
8

8

Debugging Services

■ Services provide load balancing across a set of pods. There are several common problems that can make Services not work properly.

- First, verify that there are endpoints for the service. For every Service object, the apiserver makes an endpoints resource available.
 - You can view this resource with: `kubectl get endpoints ${SERVICE_NAME}`
- Make sure that the endpoints match up with the number of containers that you expect to be a member of your service. For example, if your Service is for an nginx container with 3 replicas, you would expect to see three different IP addresses in the Service's endpoints.

■ My service is missing endpoints

- If you are missing endpoints, try listing pods using the labels that Service uses. Imagine that you have a Service where the labels are:

```
...
spec:
  - selector:
      name: nginx
      type: frontend
```

- You can use: `kubectl get pods --selector=name=nginx,type=frontend`
- to list pods that match this selector. Verify that the list matches the Pods that you expect to provide your Service.
- If the list of pods matches expectations, but your endpoints are still empty, it's possible that you don't have the right ports exposed.

M.Romdhani, 2020

9