9) Monitor Cluster Components

⇒ How do you monitor resources consumption on k8s
(or) ⇒ what would you like to monitor

⇒ I want to know node-level metrics such as
* 1) No. of. nodes & its resource utilization 2) no. of. pods & its resource utilization
No. of. nodes in the cluster, how many of them are healthy

* Performance metrics such as CPU, memory, n/w & disc utilization.

* Then pod-level metrics such as the no. of. pods, & the performance metrics of each pod such as CPU & memory consumption on them

So we need a solution that will monitor those metrics, store them & provide analytics around this data

Monitoring solutions:-
* Metric Server * Prometheus * Elastic stack

proprietary solns:-
Data dog & Dynatrace

Heapster Vs Metric Server

⇒ Heapster was one of the original projects that enabled monitoring & analysis feature for k8s

⇒ Heapster is deprecated now & a slimmed-down version was formed known as Metrics Server

⇒ we can have one Metrics Server per k8s cluster

⇒ The metrics server retrieves metrics from each of the k8s nodes & pods aggregates them, & stores them in memory

NOTE:-
The metrics server is only an in-memory monitoring soln & does not store metrics on the disk

⇒ As a result, you cannot see historical performance data, for that you must rely on one of the advanced monitoring solutions

How are the metrics generated for the pods on those nodes?

⇒ K8s runs an agent on each node known as kubelet which is responsible for receiving instructions from the K8s API master server & running pods on the nodes

⇒ The kubelet also contains a sub component known as the cAdvisor (or) Container Advisor

⇒ cAdvisor is responsible for retrieving performance metrics from pods & exposing them through kubelet API to make the metrics available for the Metrics Server

Metrics Server - Getting Started

If, we're using minikube for local cluster

```
minikube addons enable metrics-server
```

And for all other environments, deploy the Metrics Server by cloning the Metrics Server deployment files from Github repo.

```
git clone https://github.com/kubernetes-incubator/metrics-server
```

Then deploying the required components using the kubectl create command

```
kubectl create -f deploy/1.8+/
```

This command deploys a set of pods, services & roles to enable metric server to pull performance metrics from the nodes in the cluster

⇒ Once deployed, give the Metric Server some time to collect & process data.

⇒ Once processed cluster performance can be viewed by running the command kubectl top node

kubectl top node

NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%
kubemaster	166m	8%	1337Mi	70%
kubenode1	36m	1%	1046Mi	55%
kubenode2	39m	1%	1048Mi	55%

kubectl top pod

NAME	CPU(cores)	CPU%	MEMORY(bytes)	MEMORY%
nginx	166m	8%	1337Mi	70%
redis	36m	1%	1046Mi	55%

"kubectl top pod" command to view "performance metrics" of pods in k8s