K8S TOPICS

POD LIMITS:

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  containers:
  - name: my-container
    image: nginx
    resources:
      requests:
        memory: "64Mi"    # Request 64 megabytes of memory
        cpu: "250m"       # Request 250 milliCPUs (0.25 CPU cores)
      limits:
        memory: "128Mi"   # Limit memory usage to 128 megabytes
        cpu: "500m"       # Limit CPU usage to 500 milliCPUs (0.5 CPU cores)
```

KUBERNETES METRIC SERVER:

In Kubernetes, the Metric Server is a component that collects resource usage metrics from the Kubernetes API server and provides them to other components such as the Horizontal Pod Autoscaler (HPA). It is commonly used to scale applications based on CPU and memory usage.

RESOURCE MONITORING IN PODS:

Resource monitoring in Kubernetes refers to the process of collecting and analyzing metrics related to the resource usage of various components within a Kubernetes cluster. These resources typically include CPU, memory, and network usage.

There are several tools to monitor

- **Kubernetes Metrics Server**
- **Prometheus**
- **Third-party monitoring tools**

WATCH POD METRICS:

To watch pod metrics in Kubernetes, you can use the kubectl top command. This command provides resource usage metrics for pods, nodes, and other Kubernetes objects. Here's how to use it to watch pod metrics:

kubectl top pods

kubectl top nodes

HOW TO WATCH POD LEVEL ISSUES:

To watch pod level issues we need to follow the steps:

- Get pods: kubectl get po
- Describe the pod: kubectl describe pod pod-name
- Check the logs of a pod: kubectl logs pod -c container
- Check the events: kubectl get events

HOW TO WATCH NODE LEVEL ISSUES:

- Get nodes: kubectl get no
- Describe the node: kubectl describe node node-name
- Check the mem/cpu utilization: kubectl top node
- Utilize monitoring systems like Prometheus and Grafana to collect and visualize node-level metrics over time.