**1. What is kops?**

it is way to set up k8 cluster instead of using kubeadmin. It is an automation tool. It will not help to set up managed k8 cluster. We can spin up a test cluster or small dev cluster quickly, test your application, deploy your service or do any kind of load balancing. It not only create the cluster but also helps in destroy, maintain and upgrade the cluster

Kubdeadm will not help in integrating will all the external services like Ec2, ASG, route 53, EBS. Kops will help in integrating all these external services.

Kops is an automation tool using which I can set up k8 cluster in a cloud account. It helps to automate the whole setup process, destroy the cluster or upgrade it or maintain it. If I will be given change to use k8ops then I will use it to set up a quick development or testing cluster in my company.

If we want to do certain kind of testing like load testing, application testing we can quickly spin a cluster, test it and destroy it.

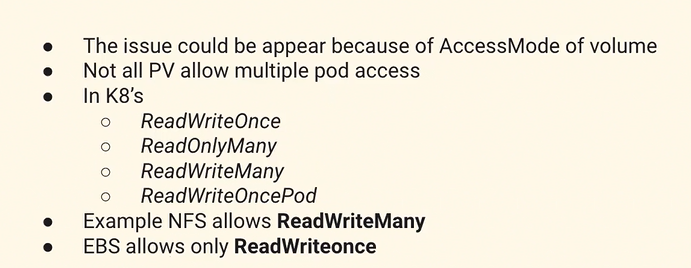
**2. Explain replication controller in k8?**

It is responsible for maintain the number of pods across the cluster**.** It is responsible for maintain x number of replicas in the cluster at any given point of time.

**3. What is pv and pvc? what role do they play?**

PVs and PVCs work together to provide persistent storage to applications running in Kubernetes Pods. The PV is the actual storage resource, while the PVC is a request made by a user or application for a certain amount and type of storage. When a PVC is created, Kubernetes finds an appropriate PV (if available) and binds them together. If no matching PV is available, Kubernetes can dynamically provision one based on the StorageClass specified in the PVC. So, PVCs help users to claim the storage they need, and PVs provide the actual storage resources to fulfill those claims.

****



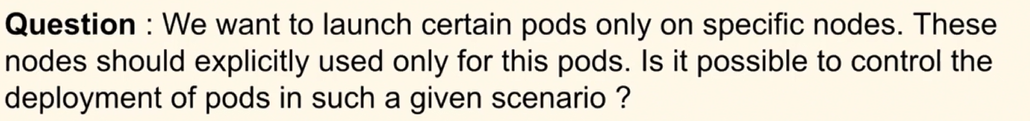
The error could be appearing because we could be trying to attach a particular volume which has only ReadWriteOnce access mode. The solution for this is to make changes in PV and PVC so that a separate volume is created for this or we make sure that the volume which we are trying to attach supports ReadWriteMany permissions.

5. What is a sidecar container?

These are the containers that helps the application running in the pod to perform the some common task. The most common example would be logging. we can have a side car container which has a purpose to send the logs from the appl container to centralized logging system. This mechanism will help us to make sure that a common use cases across ur application can be separated and a side car container will help us to make sure that our deployment is smooth and it will also remove certain complexities when we build the application.

In Kubernetes, a sidecar container is an additional container that runs alongside your main application container in the same pod. It provides extra functionality to support or enhance the main application. A sidecar container can handle tasks like logging, monitoring, or networking for your main application.

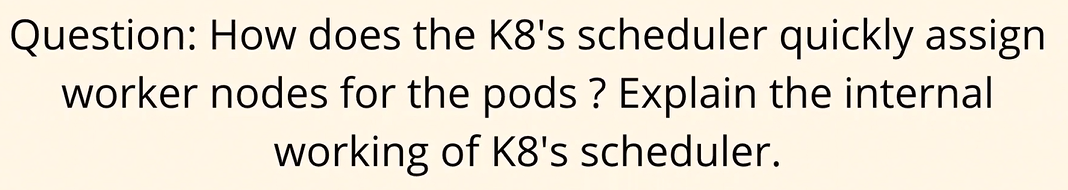
**Logging and Monitoring**: Sidecar containers can capture logs and metrics generated by the main application container and forward them to centralized logging and monitoring systems. This allows for centralized management and analysis of application logs and performance metrics.



We can use taints and toleration method in k8. Taint is a method in which we can label our node acc to our use case because of which it will repel all different kind of pods which doesn’t have the label. For the application which we want we can make sure it ahs the proper label which matches the taint label.

use taints and tolerations when you need to restrict pod scheduling based on node properties, node selector when you want to direct pods to nodes with specific attributes, and node affinity when you need more advanced rules for pod placement based on node characteristics.

in simple terms, use Deployments when you want things to be managed automatically, ReplicaSets when you want an exact number of workers, HPA when you want things to adjust automatically based on how busy it gets, and Manual Scaling when you want to decide things yourself.



We are aware that scheduler is responsible for scheduling the pods on the worker nodes. Scheduler already has the information or what we call the metadata of the worker node i.e how healthy a particular node is. Then this information is filtered and scored. There is a rank that is available for each worker node. When a pod comes for scheduling through its pod scheduling lifecycle, the information is then core related to the data that the scheduler already has. What is the requiremeng of the pod and which worler node is

The Kubernetes scheduler is a component responsible for assigning pods to worker nodes in the cluster. It continuously monitors the API server for newly created pods without a specific node assignment and selects a suitable node for them

1. \*\*Pod Scheduling Process\*\*:

- When a pod is created without a node assignment, it gets added to the scheduling queue.

- The scheduler watches this queue and selects pods one by one for scheduling.

- For each pod, the scheduler evaluates the list of candidate nodes in the cluster.

2. \*\*Node Selection\*\*:

- It filters out nodes that are not eligible for the pod based on based on several factors, including resource requirements and constraints, node affinity, taints and tolerations or other scheduling policies.

3. \*\*Scoring and Prioritization\*\*:

- After filtering the candidate nodes, the scheduler assigns a score to each remaining node based on how well it matches the pod's requirements and constraints.

- The scheduler prioritizes nodes based on this score and selects the node with the highest score as the best fit for the pod.

4. \*\*Binding the Pod to a Node\*\*:

- Once a node is selected, the scheduler updates the pod's scheduling information to reflect the assignment.

- It then sends an API request to the API server to update the pod's status with the assigned node.



Eviction. It is a node setting not a pod setting and it can be done use kubelet. Soft eviction and Hard eviction. based on memory available, inode size, docer space.

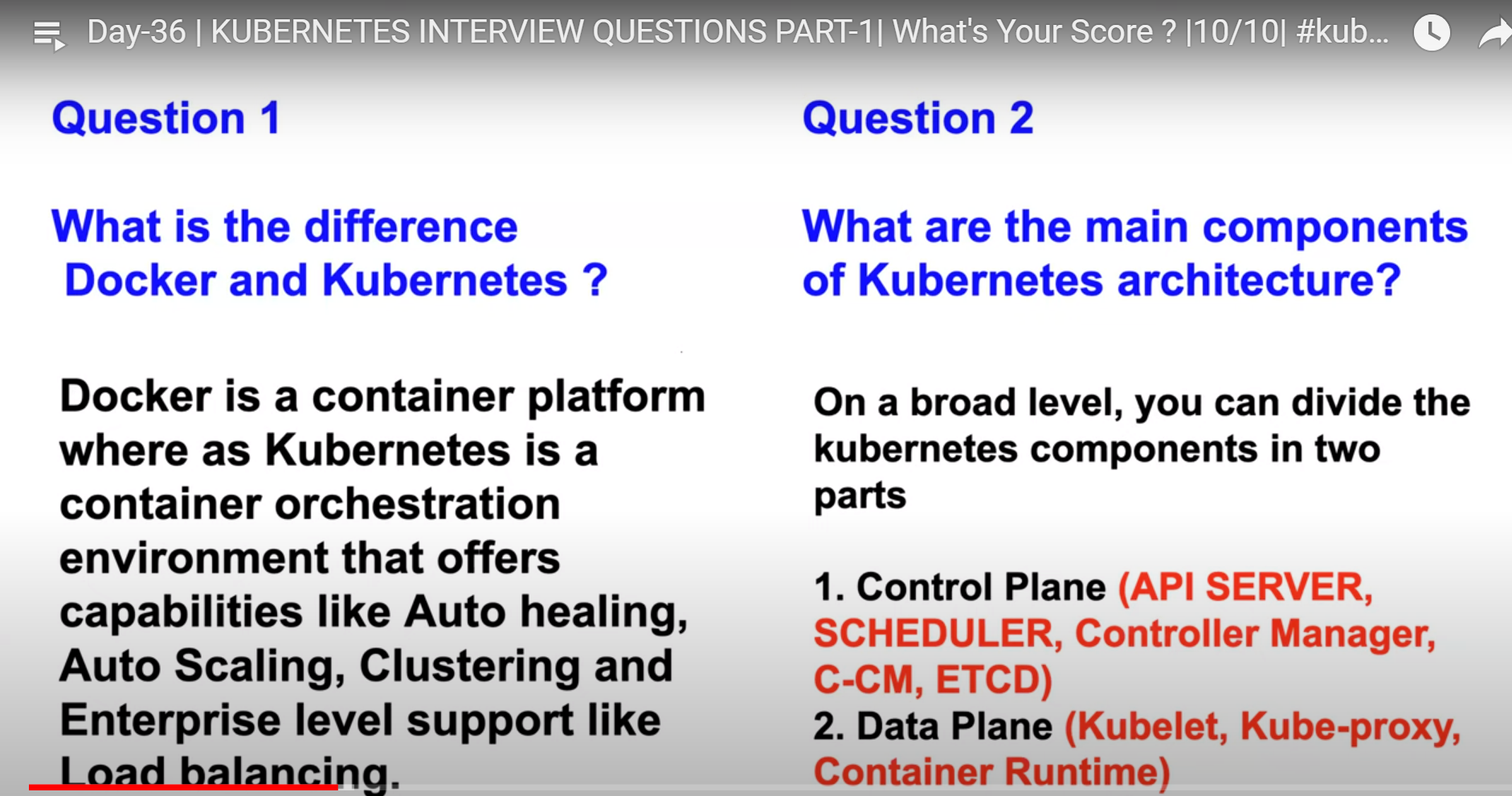
When a node has high disk utilization based on QOS it will select particular pods that it will evict. Once it has decided the pods it will evict then it will terminate these pods and try to shut down gracefully. If it is not possible then it will try to shut them forcefully. What will happen is that all the resouces utilized by those pods will be free

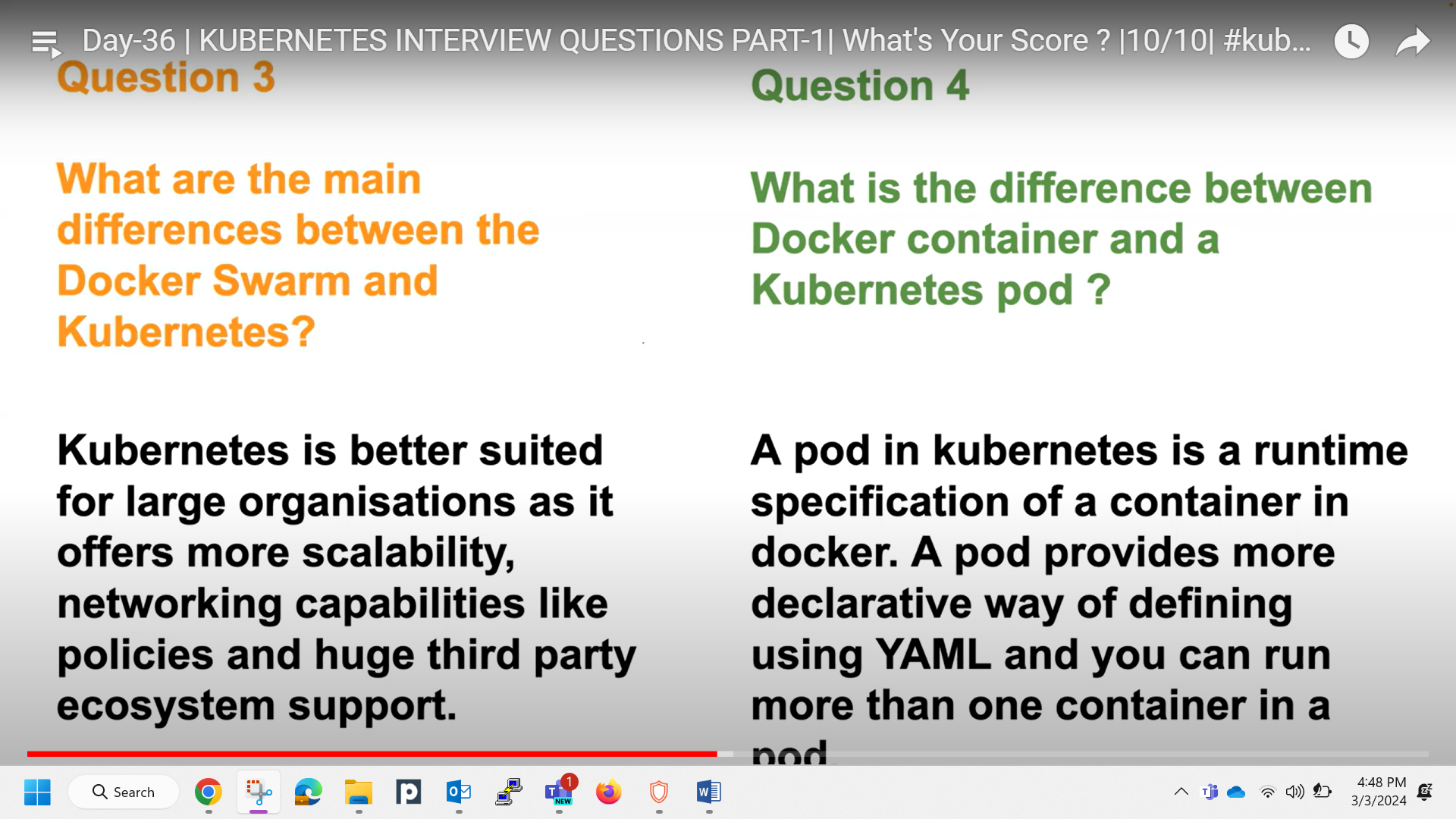
In summary, Quality of Service (QoS) classes in Kubernetes classify pods based on their resource requirements and guarantees. Pods with Guaranteed QoS have dedicated resources reserved for them and are not evicted under normal circumstances. Pods with Burstable QoS can burst beyond their requests but may be subject to eviction if resources become constrained. Pods with BestEffort QoS have no guaranteed resource allocation and are the first to be evicted when resources are scarce.

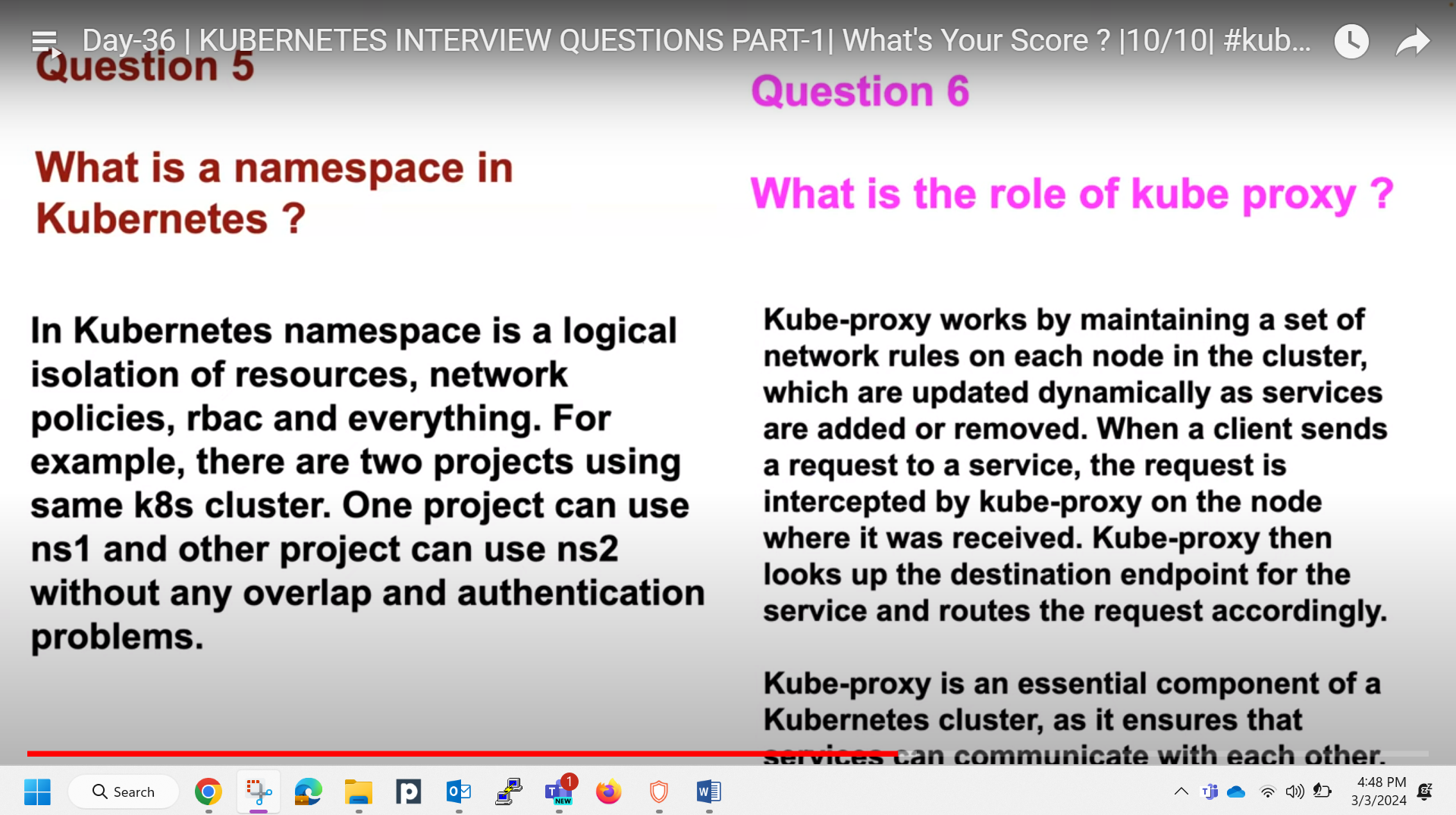
1. Day to day responsibility:  
As part of the DevOps engineer role, we manage Kubernetes clusters for our organization and we also ensure that applications are deployed onto the Kubernetes cluster and there are no issues with the application. So, we have set up monitoring on our Kubernetes cluster. We ensure that whenever there are bugs on the Kubernetes cluster, for example, if the developers are not able to troubleshoot some issue with respect to pods or services, or they are not able to route the traffic inside the Kubernetes cluster, in such cases, as subject matter experts on the Kubernetes clusters, we come into the picture and we solve their problems.

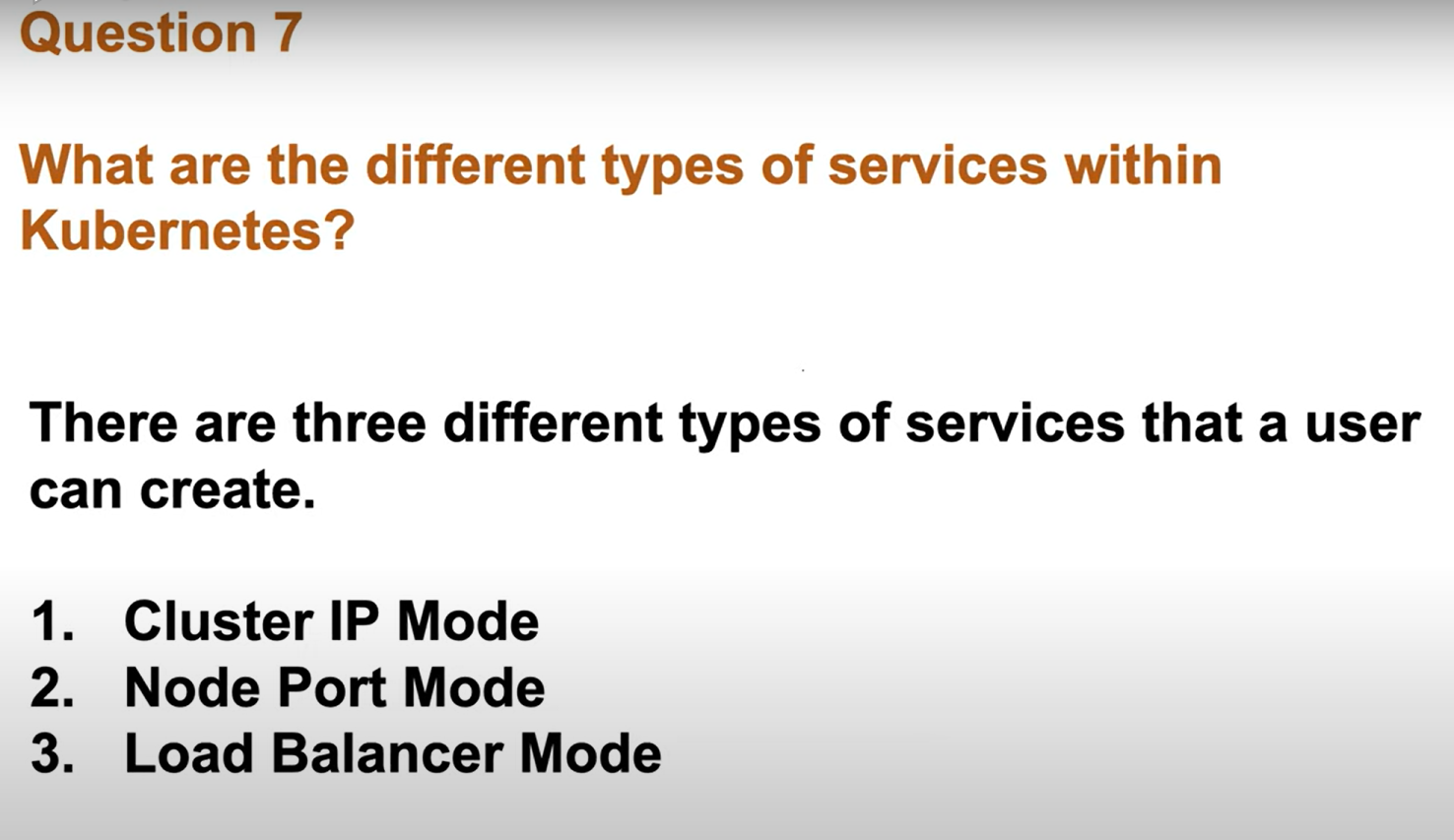
Apart from that, we also do a lot of maintenance activities. For example, we have Kubernetes clusters with three master nodes and ten worker nodes, so we have to do some continuous maintenance activities on these worker nodes. Probably upgrading the versions of these worker nodes or installing some default mandatory packages, ensuring that these worker nodes are not exposed to security vulnerabilities. So, all of these things are our day-to-day activities on Kubernetes.

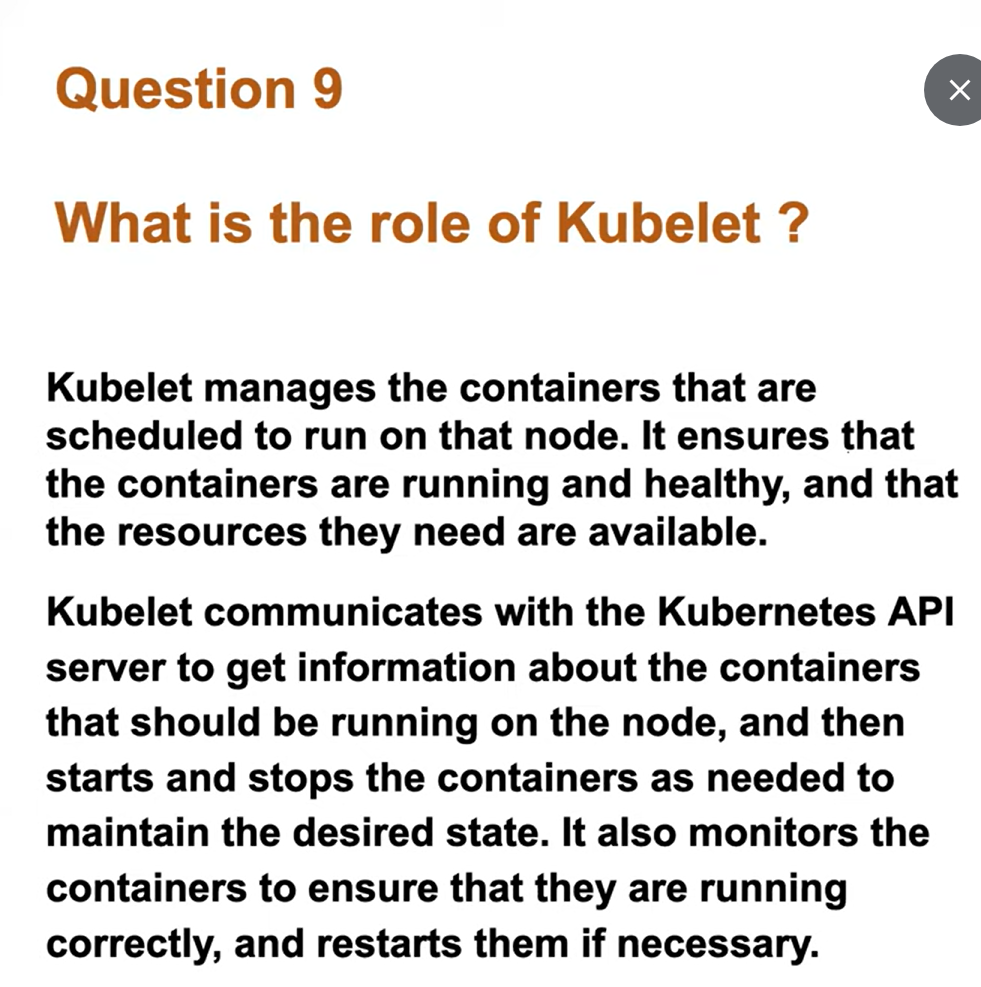
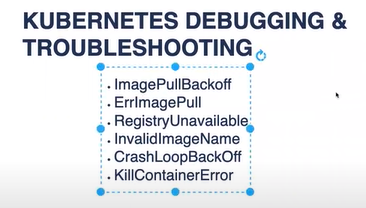
Apart from that, we also serve as subject matter experts on Kubernetes. So, if anyone in the organization has any issues with Kubernetes, they create Jira items for us or they create tickets for us, and we will help them in solving or making them understand the concept of Kubernetes.









  
  
   
diff b/w deployment, daemon set, statefulset