Kubernetes In 30 Days challenge :-

#Day16 :-

Different levels of logging in Pods (Troubleshooting)

When something is wrong with Application deployed in Pod then we can follow the below procedure to troubleshoot and/or find where the issue is...

① check the POD status and events of POD :

```
# kubectl get pod
# kubectl describe pod <pod_name>
```

② Examine the Logs :-

```
#kubectl logs <pod_name>
#kubectl logs <pod_name> -c <container_name>
```

(Multiple containers)

③ Node level troubleshooting :-

```
#kubectl get nodes
```

check the Node if any issues at Node OS level.

④ Resource constraints :-

check if Pod is hitting resource limits (CPUs, Memory). Use below cmd:

```
#kubectl describe pod <pod_name>
```

⑤ Network issues :-

→ verify that pod's networking is fine and check service, endpoints and Network firewall

→ Perform pod diagnostics after using below cmd :

#kubectl exec -it <pod_name> -- /bin/sh

⑥ check kubernetes Events :-

Use kubectl get events to check cluster wide events for any issues affecting pods

⑦ Pod Health Probes :-

If Health Probes are in place, check what is the status of all probes.

⑧ Rolling back Deployment :-

If Issue occured after deployment then perform Roll back operation using

#kubectl rollback undo deployment <deploy-name>

⑨ Persistant volumes :-

If Pod uses PV, ensure that they are properly mounted and accessible

(10) RBAC policies :-

Verify RBAC policies that are configured correctly to sa associated with pod

(11) check ~~for~~ ongoing maintainance :-

Verify that if any on-going activities at Node level.

This can be a generalised step by step process to troubleshoot a pod. But In Realtime you can get the details of Errors straight-away so, you can directly workon particular error rather sticking to single procedure

This is All about troubleshooting a pod issue in Kubernetes.