Day - 16

Different Levels of Loggingin - 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 <containel_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 (cpu's, Memory) use below Command.

```
# 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 command.

```
# 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) Rollingback Deployment:

If Issue occured after deployment then perform rollback operation using

```
# kubectl rollback undo deployment <deploy_name>
```

9) Persistent Volumes:

If pod uses persistent volume, ensure that they are properly mounted and accessible.

10> RBAC policies :

Verify RBAC policies that are configured correctly to Service account 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.