**Issue:** Node is not ready.

**Explanation**: It means that node is unable to accept the pods or workloads.

**cause**:

1. Insufficient memory, cpu or disk space

2. node might be having problem in communication with other nodes or control plane

3. condition such as disk pressure, memory pressure or network issue can make the node not ready.

4. issue with Kubelet or docker

**Resolution:**

**1.** check node status **kubectl get nodes**

2.Inspect node conditions **kubectl describe node nodename**

3. Check resource utilization **kubectl top node nodename**

4. Inspect kubelet logs **journalctl -u kubelet**

5. Restart docker or Kubelet **systemctl restart docker/kubelet**

6. Make sure the node can talk to the rest of K8 cluster by pinging the main k8 server **ping apiserverip**

7. If the node is running low on resources then adjust the resource limit

**Issue**: Pods repeatedly crashed and enter crashloopbackoff state

**Explanation**: Whenever a pod fails k8 tries it to restart but it again crashed immediately after the restart so K8 slows down the restart process so to avoid the pod from wasting resources. That is crashloopbackoff state where k8 gives pod time to restart.

**cause**:

1. resource constraint 2. application code issue 3. volume mount issue 4. Image related issue.

5. Health check failures

**Solution:**

1.Investigate the pod status and logs kubectl get pods, kubectl logs podname, kubectl describe pods

2, check the container logs

3. check for the problems in the applications code or settings

4. check If the pods has enough memory and cpu

5. check if the image used is correct and available.

6. deleting and restarting the pods can provide a temporary fix

**Issue**: Services are not accessible or showing 503 error.

**Explanation**: services giving 503 error means that they are inaccessible.

**Cause**: This error can be due to misconfigured service selector, endpoint mismatch or underlying pod issue.

**Resolution**:

1.We can kubectl get svc and kubectl describe svc to check the service configurations, endpoints and selectors.

2. We’ll ensure the endpoints specified in the service configuration correctly maps with the healthy pods, Inconsistent mapping can lead to 503 error.

3. Inspect the pod health status using kubectl get pods and kubectl describe pods to confirm they are operational.

In a previous project I encountered the same issue where incorrect service selectors caused service disruption. By aligning selectors with pod labels, we resolved the issue.

**Issue**: pods are unable to pull container images

**Explanation**: Incorrect image name, registry authentication issues or network problems may cause.

**Solution**: check image name and registry permission, ensure proper image pull secrets and investigate the network connectivity to the image registry

**Issue**: Pods are stuck in pending state due to unfulfilled pvc

**Explaination**:

1. This situation arises due to insufficient storage

2. absence of pv that matches the specifc pvc

3. mismatch of specified storage class.

**Solution**:

1. check available pv in cluster **kubectl get pv** to ensure that resources are available to satisfy the PVC request.

2. Inspect the storage class configuration to confirm if PVC corrects storage class is being used.