Bavithran
@bavicnative

Kubernetes Troubleshooting Handbook

10 Common Issues & How to Fix Them – A Practical Guide for DevOps Engineers


Bavithran
@bavicnative

1. Pod Stuck in ContainerCreating State

Pod remains in ContainerCreating status indefinitely.

Possible Reasons:

- Image pull taking too long or failing.
- Volume mount issues.
- Networking/DNS problems.

Troubleshooting Steps:

1. Check pod events:
   kubectl describe pod <pod-name>

2. Verify image pull errors:
   Look for **ImagePullBackOff** or similar errors.

3. Ensure image is accessible (check private registry credentials).

4. Verify volume mounts:
   Use **kubectl get pv,pvc** and inspect binding.

5. Network check:
   Run a test pod and use **nslookup** to confirm DNS.

2. CrashLoopBackOff Error

Problem: Pod starts, crashes, and restarts repeatedly.

Possible Reasons:

- Application error (wrong config, port, DB issue).
- Readiness/liveness probe failures.
- Missing dependencies.

Troubleshooting Steps:

1. View logs:
   kubectl logs <pod-name> --previous

2. Check pod events and status:
   kubectl describe pod <pod-name>

3. Validate probes in YAML.

4. Run the container locally with the same command.

5. Ensure secrets/configs are mounted correctly.

3. Node Not Ready

Problem: One or more nodes are in NotReady state.

Possible Reasons:

- kubelet down.
- Out of disk/memory.
- Network unreachable.

Troubleshooting Steps:

1. Check node status:
   kubectl get nodes

2. SSH into node:
   sudo systemctl status kubelet

3. Check disk:
   df -h

4. Restart kubelet if needed:
   sudo systemctl restart kubelet

5. Verify firewall/network.

4. DNS Resolution Failure Inside Pods

Problem: Application inside pod cannot resolve external/internal domains.

Possible Reasons:

- CoreDNS not working.
- Network policy blocking access.

Troubleshooting Steps:

1. Exec into pod and test DNS:
   nslookup google.com

2. Check CoreDNS status:
   kubectl get pods -n kube-system

3. View CoreDNS logs:
   kubectl logs <coredns-pod> -n kube-system

4. Inspect **resolv.conf** inside pod.

5. Review network policies.

Bavithran
@bavicnative

5. Services Not Accessible via ClusterIP

Problem: Service is created, but cannot access the pod behind it.

Possible Reasons:

- Pod labels don’t match selector.
- Port mismatch.

Troubleshooting Steps:

1. Check pod labels and service selector.

2. Use **kubectl port-forward** to verify pod is responding.

3. Use **kubectl describe svc <name>** to inspect target ports.

4. Run busybox or curl pod and test service.

5. Validate endpoints:
   kubectl get endpoints <svc-name>

6. PVC Stuck in Pending State

Problem: PersistentVolumeClaim is not bound to a volume.

Possible Reasons:

- No matching PV.
- Storage class issue.

Troubleshooting Steps:

1. Check PVC:
   kubectl get pvc

2. Inspect events:
   kubectl describe pvc <name>

3. Validate storage class:
   kubectl get sc

4. Create PV manually if needed.

5. Check cloud provider storage integration.

7. kubectl exec/logs Hanging

Problem: kubectl exec or logs hangs or takes long time.

Possible Reasons:

- kubelet-to-pod communication broken.
- Network latency.
- Pod not fully initialized.

Troubleshooting Steps:

1. Check node and pod IP.

2. Try **kubectl describe** – if fast, it's exec/logs issue.

3. Check for high CPU/memory usage on node.

4. Try restarting kubelet or use **-it** for interactive sessions.

5. Verify container runtime (e.g., containerd/docker).

8. Ingress Not Routing Traffic

Problem: Ingress is created but doesn't route to the backend service.

Possible Reasons:

- Wrong host/path rules.
- Backend service not reachable.
- Ingress controller not running.

Troubleshooting Steps:

1. Check Ingress rules:
   kubectl describe ingress <name>

2. Validate DNS mapping for hostname.

3. Check Ingress controller logs.

4. Use curl from internal pod to test backend.

5. Look at Ingress resource events.

9. HPA Not Scaling Pods

Problem: Horizontal Pod Autoscaler (HPA) not scaling up/down.

Possible Reasons:

- Metrics server not installed or unhealthy.
- CPU/memory usage below threshold.

Troubleshooting Steps:

1. Check HPA status:
   kubectl get hpa

2. Inspect metrics:
   kubectl top pod

3. Check metrics-server:
   kubectl get pods -n kube-system

4. View HPA YAML and ensure thresholds are valid.

5. Load test to generate usage.


Bavithran
@bavicnative

10. Deployment Update Not Rolling Out

Problem: kubectl apply runs, but pod doesn't update.

Possible Reasons:

- No actual change in spec.
- Wrong image tag (same as previous).

Troubleshooting Steps:

1. Check rollout status:
   kubectl rollout status deploy <name>

2. View rollout history:
   kubectl rollout history deploy <name>

3. Check image tag – ensure it changed.

4. Force rollout with annotation change:

kubectl patch deploy <name> -p '{"spec":{"template":{"metadata":{"annotations":{"force":"yes"}}}}}'

5. Confirm new pods launched.

Found this useful?

Follow

Let's connect

💬 Found it useful? Drop your thoughts below and share it with your fellow DevOps engineers!