



Troubleshooting in Kubernetes

By DevOps Shack

25 Examples With Commands

1. Error: Unable to connect to the cluster

- **Troubleshooting:**
 - Check kubeconfig file for correct cluster information.
 - Verify network connectivity to the cluster.
- **Example Commands:**

```
kubectl config view
kubectl cluster-info
```

2. Error: Pod stuck in Pending state

- **Troubleshooting:**
 - Check events for the pod using `kubectl describe pod`.
 - Inspect the pod's YAML for resource constraints or affinity issues.
- **Example Commands:**

```
kubectl describe pod <pod-name>
kubectl get events --namespace <namespace>
```

3. Error: Insufficient resources to schedule pod

Troubleshooting:

- Check resource requests and limits in the pod specification.
- Verify node resources using `kubectl describe node`.

- **Example Commands:**

```
kubectl describe pod <pod-name>
kubectl describe node <node-name>
```

4. Error: ImagePullBackOff

- **Troubleshooting:**

- Verify the image name and availability.
- Check image pull credentials using `kubectl describe pod`.

- **Example Commands:**

```
kubectl describe pod <pod-name>
kubectl get pods --namespace <namespace> -
o=jsonpath='{.items[*].status.containerStatuses[*].state}'
```

5. Error: CrashLoopBackOff

- **Troubleshooting:**

- Check container logs for details on the crash.
- Inspect pod events using `kubectl describe pod`.

- **Example Commands:**

```
kubectl logs <pod-name> <container-name>
kubectl describe pod <pod-name>
```

6. Error: Unauthorized access

- **Troubleshooting:**

- Verify RBAC permissions for the user.
- Check kubeconfig for correct credentials.

- **Example Commands:**

```
kubectl auth can-i --list
kubectl config view
```

7. Error: ConfigMap not updating in the pod

- **Troubleshooting:**

- Check if the ConfigMap is updated.
- Verify that the pod is configured to use the latest version.

Example Commands:

```
kubectl get configmap <configmap-name> -o yaml  
kubectl describe pod <pod-name>
```

8. Error: Service not reachable

- **Troubleshooting:**
 - Check service endpoints using `kubectl describe service`.
 - Verify network policies and firewall rules.

- **Example Commands:**

```
kubectl describe service <service-name>  
kubectl get networkpolicies
```

9. Error: Node not ready

- **Troubleshooting:**
 - Check node status with `kubectl get nodes`.
 - Review kubelet logs on the node for issues.

- **Example Commands:**

```
kubectl get nodes  
kubectl describe node <node-name>
```

10. Error: PersistentVolumeClaim (PVC) pending

- **Troubleshooting:**
 - Verify available storage in the cluster.
 - Check storage class and provisioner.

- **Example Commands:**

```
kubectl get pvc  
kubectl describe storageclass
```

11. Error: VolumeMounts not working in pod

- **Troubleshooting:**
 - Check pod's YAML for correct volume mounts.
 - Verify if the volume exists and is accessible.

- **Example Commands:**

```
kubectl describe pod <pod-name>
kubectl get pv
```

12. Error: Pod Security Policies (PSP) blocking pod

- **Troubleshooting:**

- Check PSP rules and RBAC for the pod.
- Inspect pod events using `kubectl describe pod`.

- **Example Commands:**

```
kubectl get psp
kubectl describe pod <pod-name>
```

13. Error: ServiceAccount permissions

- **Troubleshooting:**

- Verify ServiceAccount permissions using `kubectl auth can-i`.
- Check RBAC roles and role bindings.

- **Example Commands:**

```
kubectl auth can-i --list --
as=system:serviceaccount:<namespace>:<serviceaccount-name>

kubectl get roles,rolebindings --namespace <namespace>
```

14. Error: NodeSelector not working

- **Troubleshooting:**

- Check pod's YAML for correct node selector.
- Verify that nodes have the required labels.

- **Example Commands:**

```
kubectl describe pod <pod-name>
kubectl get nodes --show-labels
```

15. Error: Ingress not routing traffic

- **Troubleshooting:**

- Check Ingress resource for correct backend services.

- Verify that the Ingress controller is running.
- **Example Commands:**

```
kubectl describe ingress <ingress-name>
kubectl get pods --namespace <ingress-controller-namespace>
```

16. Error: Unable to scale deployment

- **Troubleshooting:**
 - Verify available resources in the cluster.
 - Check replica count in the deployment specification.
- **Example Commands:**

```
kubectl get deployments
kubectl describe deployment <deployment-name>
```

17. Error: Custom Resource Definition (CRD) not creating resources

- **Troubleshooting:**
 - Check CRD definition for correct syntax.
 - Verify controller logs for errors.
- **Example Commands:**

```
kubectl get crd
kubectl describe crd <crd-name>
```

18. Error: Pod in Terminating state

- **Troubleshooting:**
 - Check for stuck finalizers in pod metadata.
 - Force delete pod using `kubectl delete pod --grace-period=0`.
- **Example Commands:**

```
kubectl get pods --all-namespaces --field-selector=status.phase=Terminating
kubectl delete pod <pod-name> --grace-period=0 -force
```

19. Error: Resource quota exceeded

- **Troubleshooting:**
 - Check resource quotas for the namespace.
 - Verify resource usage in the namespace.
- **Example Commands:**

```
kubectl describe quota --namespace <namespace>
kubectl top pods --namespace <namespace>
```

20. Error: Rolling update stuck or not progressing

- **Troubleshooting:**
 - Check rollout status using `kubectl rollout status`.
 - Verify image versions in the deployment.
- **Example Commands:**

```
kubectl rollout status deployment <deployment-name>
```

```
kubectl set image deployment/<deployment-name> <container-name>=<new-image>
```

21. Error: Node draining or cordoning

- **Troubleshooting:**
 - Check node conditions and events.
 - Use `kubectl drain` with caution.
- **Example Commands:**

```
kubectl get nodes
kubectl describe node <node-name>
kubectl drain <node-name> --ignore-daemonsets
```

22. Error: Resource creation timeout

- **Troubleshooting:**
 - Check for issues with the API server.
 - Verify network connectivity to the API server.
- **Example Commands:**

```
kubectl get events --sort-by='.metadata.creationTimestamp'
kubectl describe pod <pod-name>
```

23. Error: Pod stuck in ContainerCreating state

- **Troubleshooting:**
 - Check container runtime logs on the node.
 - Inspect kubelet logs for errors.
- **Example Commands:**

```
kubectl get pods
kubectl describe pod <pod-name>
```

24. Error: Invalid YAML syntax

- **Troubleshooting:**
 - Validate YAML syntax using online tools or linters.
 - Check for indentation and formatting issues.
- **Example Commands:**

```
kubectl apply -f <file.yaml> --dry-run=client
```

25. Error: etcd cluster issues

- **Troubleshooting:**
 - Check etcd logs for errors.
 - Verify etcd cluster health.
- **Example Commands:**

```
kubectl get events --all-namespaces --field-selector=involvedObject.kind=Pod,involvedObject.name=etcd
```

```
kubectl exec -it etcd-pod-name --namespace kube-system -- sh
etcdctl member list
etcdctl cluster-health
```

Remember to replace placeholders like <pod-name>, <namespace>, <deployment-name>, etc., with actual values specific to your environment. Additionally, exercise caution when using force deletion or draining nodes to avoid potential data loss or service disruption.