Debugging in Kubernetes

Relevant Documentation

- Troubleshoot Applications
- Application Introspection and Debugging
- Monitoring, Logging, and Debugging

Exam Tips

- Use `kubectl get pods` to check the status of all Pods in a Namespace. Use the `--all-namespaces` flag if you don't know what Namespace to look in.
- Use `kubectl describe` to get detailed information about Kubernetes objects.
- Use `kubectl logs` to retrieve container logs.
- Check cluster-level logs if you still cannot locate any relevant information.

Lesson Reference

Log in to the **control plane node**.

Create a broken Pod.

```
vi broken-pod.yml
```

```
apiVersion: v1
kind: Pod
metadata:
  name: broken-pod
spec:
  containers:
  - name: nginx
    image: nginx:1.20.q
    livenessProbe:
      httpGet:
        path: /
        port: 81
      initialDelaySeconds: 3
      periodSeconds: 3
```

```
kubectl apply -f broken-pod.yml
```

Get a list of Pods in the `default` namespace.

```
kubectl get pods

kubectl get pods -n default
```

Get a list of Pods in all Namespaces.

```
kubectl get pods --all-namespaces
```

Get more details about the Pod.

```
kubectl describe pod broken-pod
```

Edit the yaml to fix the Pod's image version.

```
vi broken-pod.yml
```

```
    image: nginx:1.20.1
```

Delete and re-create the Pod.

```
kubectl delete pod broken-pod --force

kubectl apply -f broken-pod.yml
```

Check the Pod status again. You should notice that it begins restarting repeatedly. This is because the liveness probe is not configured correctly.

```
kubectl get pod broken-pod
```

Check the Pod's container logs.

```
kubectl logs broken-pod
```

Check the Pod's YAML manifest.

```
kubectl get pod broken-pod -o yaml
```

Edit the yaml to fix the liveness probe.

```
vi broken-pod.yml
```

```
    livenessProbe:
      httpGet:
        path: /
        port: 80
```

Delete and re-create the Pod.

```
kubectl delete pod broken-pod --force

kubectl apply -f broken-pod.yml
```

Check the Pod status again.

```
kubectl get pod broken-pod
```

The Pod should now be working correctly.

Check the kube-apiserver logs. Note that the log file name contains a random hash. You will need to browse the file system within `/var/log/containers/` to find the file.

```
sudo cat /var/log/containers/kube-apiserver-k8s-control_kube-system_kube-apiserver-<hash>.log
```

Check the kubelet logs.

```
sudo journalctl -u kubelet
```