LOGGING AND TROUBLESHOOTING

Logging and Troubleshooting

Cluster Start Sequence

The cluster startup sequence begins with systemd if you built the cluster using **kubeadm**. Other tools may leverage a different method. Use **`systemctl status kubelet.service`** to see the current state and configuration files used to run the kubelet binary.

- Uses **`/etc/systemd/system/kubelet.service.d/10-kubeadm.conf`** file

Inside of the **`config.yaml`** file you will find several settings for the binary, including the **`staticPodPath`** which indicates the directory where kubelet will read every yaml file and start every pod. If you put a yaml file in this directory, it is a way to troubleshoot the scheduler, as the pod is created with any requests to the scheduler.

- Uses **`/var/lib/kubelet/config.yaml`** configuration file
- **`staticPodPath`** is set to **`/etc/kubernetes/manifests/`**

The four default yaml files will start the base pods necessary to run the cluster:

- kubelet creates all pods from *.yaml in directory: kube-apiserver, etcd, kube-controller-manager, kube-scheduler.

Once the watch loops and controllers from kube-controller-manager run using etcd data, the rest of the configured objects will be created.