

Step 7 for All Nodes. Verify the KumoScale Storage Cluster Configuration

This page explains Step 7 of installing KumoScale for Appliance Mode in a multi-node environment.

•Complete the Set up of the KumoScale Storage Cluster

Step 7. Verify the KumoScale storage cluster configuration.

Step 8. Install internal components.

At any point in the process, you can submit any of the commands listed in this section to get status of services, nodes, or pods. Use one or more of these commands to confirm that the storage cluster is set up with the correct number of masters.

`kubectl get nodes -A -o wide` will show several pieces of information including what is running on all nodes. In the image below, KumoScale software has been installed on one node.

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS G
ATES									
cert-manager	cert-manager-66dc8c1f6b-bdv9p	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
cert-manager	cert-manager-cainjector-5c5d4d8f7-arpl4	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
cert-manager	cert-manager-webhook-6f999c4de-1m4b	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	calico-kube-controllers-6c79551d8-1hmb	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	calico-node-1n6d1	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	coredns-6c77b6459-12ncg	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	coredns-6c77b6459-qqqg	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	csi-kumospace-controller-66b7d7f6c9-51dm	6/6	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	csi-kumospace-node-8dbw	2/2	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	etcd-ks-node1-895856a08352	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	kube-apiserver-ks-node1-895856a08352	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	kube-controller-manager-ks-node1-895856a08352	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	kube-proxy-pdtk	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	kube-scheduler-ks-node1-895856a08352	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kube-system	kube-vip-ks-node1-895856a08352	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kumo-services	elasticsearch-master-0	0/1	Running	0	8d		ks-node1-895856a08352	<none>	<none>
kumo-services	elasticsearch-master-1	0/1	Pending	0	8d		<none>	<none>	<none>
kumo-services	elasticsearch-master-2	0/1	Pending	0	8d		<none>	<none>	<none>
kumo-services	ks-config-operator-controller-manager-55d4f876c-dz7d4	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kumo-services	ks-install-operator-84f893cd4-7d4w	1/1	Running	0	21d		ks-node1-895856a08352	<none>	<none>
kumo-services	ks-provisioner-deployment-85f0770557-f2196	2/2	Running	0	21d		ks-node1-895856a08352	<none>	<none>

`kubectl get svc -A -o wide` will show which services are running. You should see at least the primary storage cluster, the Provisioner Service, the CSI driver, and control operators (`ks-install-operator` and `ks-config-operator`) as shown below.

<pre>(root@ks-node1) kubectl get nodes -A -o wide</pre>									
NAME	STATUS	ROLES	AGE	VERSION	INTERNAL-IP	EXTERNAL-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
ks-node1-895856a08352	Ready	Master	85m	v1.17.5	172.28.1.161	<none>	CentOS Linux 7 (Core)	4.18.0-348	docker://19.3.13
<pre>(root@ks-node1) kubectl get pods -A -o wide</pre>									
NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATE
cert-manager	cert-manager-66dc8c1f6b-4k8d9	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
cert-manager	cert-manager-cainjector-5c5d4d8f7-ky6d2	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
cert-manager	cert-manager-webhook-6f999c4de-qgrb5	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	coredns-6c77b6459-d7ztp	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	coredns-6c77b6459-rf65	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	csi-kumospace-controller-9987917-f6x76	6/6	Running	0	84m		ks-node1-895856a08352	<none>	<none>
kube-system	csi-kumospace-node-8dbw	2/2	Running	1	84m		ks-node1-895856a08352	<none>	<none>
kube-system	etcd-ks-node1-895856a08352	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	kube-apiserver-ks-node1-895856a08352	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	kube-controller-manager-ks-node1-895856a08352	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	kube-proxy-w2f9n	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	kube-scheduler-ks-node1-895856a08352	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	kube-vip-ks-node1-895856a08352	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kube-system	nginx-ingress-nginx	2/2	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kumo-services	ks-config-operator-controller-manager-c55d4f876c-1m4kx	1/1	Running	0	85m		ks-node1-895856a08352	<none>	<none>
kumo-services	ks-install-operator-68b0847f8-4zpf	1/1	Running	0	84m		ks-node1-895856a08352	<none>	<none>
kumo-services	ks-provisioner-deployment-7d4b4999-924jx	2/2	Running	0	84m		ks-node1-895856a08352	<none>	<none>
<pre>(root@ks-node1) kubectl get svc -A -o wide</pre>									
NAMESPACE	NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE	SELECTOR		
cert-manager	cert-manager	ClusterIP	<none>		9402/TCP	85m	app.kubernetes.io/component=controller,app.kubernetes.io/instance=cert-manager,app.kubernetes.io/name=cert-manager		
cert-manager	cert-manager-webhook	ClusterIP	<none>		443/TCP	85m	app.kubernetes.io/component=webhook,app.kubernetes.io/instance=cert-manager,app.kubernetes.io/name=webhook		
kube-system	default	ClusterIP	<none>		443/TCP	85m	<none>		
kube-system	kube-dns	ClusterIP	<none>		53/TCP, 53/TCP, 9153/TCP	85m	app.kubernetes.io/name=kube-dns		
kumo-services	ks-config-operator-webhook-service	ClusterIP	<none>		443/TCP	84m	control-plane=controller-manager		
kumo-services	ks-install-operator	ClusterIP	<none>		8080/TCP, 8080/TCP	84m	app.kubernetes.io/name=ks-install-operator		
kumo-services	ks-install-operator-metrics	ClusterIP	<none>		8083/TCP, 8086/TCP	84m	app.kubernetes.io/name=ks-install-operator		
kumo-services	ks-provisioner-service	ClusterIP	<none>		8080/TCP	84m	app.kubernetes.io/name=ks-provisioner		

`kubectl get pods -A -o wide` will similarly show the pods set up for KumoScale software services and includes the KumoScale Provisioner service, control operators, and the CSI driver. You should also see any other cluster pods you may have set up.

Verifying Individual Services on the KumoScale Storage Cluster

You can see which pods are running with the command:

```
kubectl get pods -n kumo-services
```

It can be useful to individually verify that all storage cluster services (Provisioner Service, control operators, the CSI driver) are running.

Provisioner Service: To verify that the Provisioner Service is running, read its details and issue either of the following commands:

```
kubectl get provisionerservices
kubectl get services -A | grep provisioner
```

CSI driver for Kubernetes orchestration: To verify that the CSI service is running, issue the following command:

```
kubectl get csiservices
```

Control Operators: To verify both `ks-install-operator` and `ks-config-operator` are running, use either:

```
kubectl get services -A | grep ks-install-operator
```

or:

```
kubectl get services -A | grep ks-config-operator
```

Next: [Step 8. Install Internal Components on the KumoScale Storage Cluster](#)
