

KumoScale Troubleshooting and Support

This section lists situations you may encounter when using KumoScale, what they mean, and how to resolve them.

- Below are the topics covered on this page
- [Storage node is created but is not available](#)
- [Adding a second master in Appliance mode](#)
- [Collecting logs on the KumoScale Provisioner](#)
- [Paging in commands with large output](#)

Storage Node is Created but not Available

If a storage node is created but is not available, it usually means that the user has provided a KumoScale software secret but does not have a valid license. You will not be able to resolve the problem by applying changes or deleting the node. You will need to contact KumoScale Technical Support to assist you in resolving this issue.

Adding a Second Master in Appliance Mode

When using KumoScale in Appliance mode, if you add a second storage node as a master to the cluster and execute the `kubectl describe` command, you may get the message “**Cannot add a master while there are not enough masters pending.**” Since the number of master nodes needs be an odd number, the master will stay in pending state, until a third storage node is added.

KumoScale Provisioner Service Log Collection

To collect logs on provisioning into the file `provisioner.log`, complete the following:

1. Identify the name of the KumoScale Provisioner service, using either:

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

or

```
kubectl get provisionerservices.provisioner.kumoscale.kioxia.com
```

2. Collect provisioning data from `<ks_prov_pod_name>`, the service returned above, with:

```
kubectl logs -n kumo-services <ks_prov_pod_name> ks-provisioner > provisioner.log
```

For example, to collect KumoScale Provisioner service logs for the Provisioner service identified by `ks-provisioner-deployment-#####-abcde` pod, use:

```
kubectl logs -n kumo-services ks-provisioner-deployment-#####-abcde ks-provisioner > provisioner.log
```

Paging in Commands with Large Output

KumoScale software supports up to 16K volumes and 1K targets per system. This may cause analysis to be very slow due to the large number of items output by KumoScale commands.

The KumoScale REST API supports several options to overcome this problem:

- Filtering** – Return only the information you need by using Get commands to filter the results based on input parameters.
- Paging** – Specify the maximum number of items for the response along with the ID of the last item received, thus reading the response in measured quantities.

