

Maintaining and Using KumoScale for Appliance Mode

This section explains how to update, use, and remove KumoScale software for Appliance mode.

This page includes the following information:

- [Upgrading the KumoScale Software Cluster](#)
- Updating KumoScale Components: [License](#) and [Syslog](#)
- [Upgrading the SSD Firmware](#)
- [Installing NVMe Patches on Application Initiators](#)

Upgrading the KumoScale Software Cluster

This section describes the requirements and process for upgrading an existing cluster from KumoScale software version 3.22^[1]. The process will upgrade the following components in the order below:

- Storage Nodes:** The management engine, target module, CLI, and local registry catalogue will be upgraded with all new containers.
- KumoScale software and Kubernetes internal services:** All services automatically installed with the KumoScale software, such as the operators, CSI, and Provisioner will be upgraded.

The upgrade process maintains configuration and user data between versions. If you have a high-availability KumoScale Installation, the upgrade will happen without service disruption. It does NOT support the following:

- Upgrades to services that were manually installed by the user. For example, Prometheus, Fluentd, and Loki, must be manually upgraded through the CRD.
- Upgrades to the Private Kubernetes cluster for the KumoScale storage cluster.
- Upgrades to the storage node's Operating System (OS) and kernel versions.
- Upgrades to external components and agents deployed using the OpenStack Cinder driver, Nova Connector, Ansible Client, Ansible modules, and Customer-developed CSI deployments in Kubernetes.
- Adding a new storage node or removing an existing one during the cluster upgrade.

A syslog event is generated at the beginning and end of all upgrades so that you can review the results of a successful or failed upgrade.

Note: If you need to upgrade a specific component independently, you must contact KIOXIA technical support for guidance. The process varies from the one described and supported in this document.

Requirements for Upgrading

To upgrade an existing cluster, you need to confirm the following:

- The cluster is healthy. In particular:
 - All storage nodes are up and running.
 - The time is synchronized across all nodes in the cluster.
 - The KumoScale Provisioner service has a valid license.
 - The KumoScale secret file is deployed and active.
 - There is at least one initialized storage node.
- All KumoScale storage nodes and components are running the same version.
- No node or replica is in a synchronizing state.
- No storage node has a replica of a degraded resilient volume. Such a node cannot be rebooted; thus, you must wait to upgrade until this condition is resolved.

Upgrade Process

Once you have confirmed all requirements for upgrading have been met, you are ready to complete the following steps to perform the upgrade.

- Follow the instructions for preparing files for the installation process as documented in [Prepare the Installation File and Environment](#).
- Using boot menu parameters as explained in [Determine Installation Settings](#), specify the value **upgrade** for command line parameter **kx_mode**:

```
kx_mode=upgrade
```

Do NOT pass any other **kx_** parameter other than **kx_debug**; passing any other **kx** parameter will cause the upgrade to fail.

- Start the installation as instructed in [Install the KumoScale Software](#).
- Once the installation is completed, modify the software upgrade secret yaml file as follows:

- Insert the URL from step 1.
- If username and password are required for URL access, encode them into base 64 and add to the file under data.

For example, using the secret yaml file called softwareupgrade-secret.yaml:

```
apiVersion: v1
kind: Secret
metadata:
  name: softwareupgrade-secret
  namespace: default
type: Opaque
stringData:
  url: ftp://<server_IP_address>:/ftp/upgrade_nvmf_<version>-<subversion>.tar.gpg
  data:
    username: dGVzdHVzZXI=
    password: QUJDREYmZQ=
```

5. Create the KumoScale software upgrade secret CR with the command:

```
kubectl create -f softwareupgrade-secret.yaml
```

You should receive the message secret/softwareupgrade-secret created.

You can also verify the secret was created with `/usr/bin/kubectl get secret` to see the name, type, and age of the secret.

6. Modify the software upgrade CR YAML file with values for:
- a. **upgradeSecretName:** The name of the secret file from step 2 containing the ftp, http, or https connection details.
 - b. **waitPeriod:** The time to wait between storage nodes upgrades in minutes. The default (and recommended) value is 5 minutes. The value can range between 1 and 1440. 1440 minutes is equivalent to 1 day.
 - c. **adminState:** This should be equal to either RUN or STOP to indicate whether the Admin user can resume (RUN) or stop (STOP) the upgrade process. The default value is RUN. Admin users can change **adminState** (from STOP to RUN or from RUN to STOP) as long as the state of the upgrade is equal to Pending, Upgrading, or Stopped.
- Below is an example of the upgrade CR file **kumoscale_v1_softwareupgrade_cr.yaml**:

```
apiVersion: kumoscale.kioxia.com/v1
kind: SoftwareUpgrade
metadata:
  name: softwareupgrade
spec:
  upgradeSecretName: softwareupgrade-secret
  waitPeriod: 5
  adminState: RUN
```

7. Start the upgrade process by creating the software upgrade CR. For example, using **kumoscale_v1_softwareupgrade_cr.yaml**:

```
kubectl create -f kumoscale_v1_softwareupgrade_cr.yaml
```

The nodes will be rebooted once in a cycle.

8. Verify the upgrade was successful by checking the version returned by either of the following commands.

```
kubectl get softwareupgrade -o wide

kubectl describe softwareupgrade <name>
```

Updating KumoScale Components

This section explains how to perform updates on various components of KumoScale software. As mentioned in [Logging into the KumoScale Storage Cluster from a Known Remote Host](#), these user operations need to be done externally from a remote host via kubectl commands.

Updating the License

Edit the license CR with the new license key and execute the following:

```
kubectl apply -f kumoscale.kioxia.com_v1_license_cr.yaml
```

Updating a Syslog

Edit the Syslog CR with the new version information and execute the following:

```
kubectl apply -f kumoscale.kioxia.com_syslogs.yaml
```

Upgrading the SSD Firmware

Note: Selecting a missing SSD invalidates the process

The SSD firmware upgrade is done using SSD Custom Resources. See the [User Guide](#) for complete instructions on how to upgrade firmware.

NVMe Patches on Applications Initiators

This section describes how to setup a patch for NVMe module application initiators with an example (kernel 4.18.0-193.el8). This patch resolves timing issues seen when an initiator tries to connect to a target.

To verify the patch is installed, run the following on the initiator machine

```
# modinfo nvme-core|grep description
```

If the result is nvme host KIOXIA patch (compiled 2020-12-01_12-13-59), then the patch is installed, otherwise the initiator runs with original kernel module.

To install the NVMe initiator patch:

```
# rpm -i nvme-centos82-patch-2020_12_01_12_13_59-1.x86_64.rpm
```

To uninstall the NVMe initiator patch:

```
# rpm -e nvme-centos82-patch-2020_12_01_12_13_59-1.x86_64
```

[\[1\]](#) To upgrade KumoScale cluster running software versions older than 3.18, please contact your KIOXIA support team.