

Creating and Managing Storage Nodes in KumoScale using Custom Resource Files

This section describes how to use storage node custom resource files to configure nodes to join as masters or workers.

The following topics are covered:

- [Defining storage nodes](#)
- [Creating storage nodes](#)
- [State and status of storage nodes](#)
- [Parameters that may be defined on storage nodes](#)
- [Example storage nodes](#)
- [Updating storage nodes](#)
- [Deleting storage nodes](#)

Warning: When creating a KumoScale storage node, all existing data on its SSDs will be deleted.

To use custom resource files to create a storage node that will automatically join as a master or be added to the KumoScale Provisioner service:

- [Define your Customized Storage Nodes](#) by specifying values for [storage node parameters](#) in a the storage node CR file.
- Create the storage nodes using kubectl with your storage node CRD or your orchestration interface.

This is described in detail in the next section

Step 1. Define Your Storage Nodes

KumoScale provides a sample storage node CRD file in the **KumoScale_Operator/ks-config-operator/samples** directory that include parameters for system, network, and topology of the storage node. This section explains how to set up your own storage node CRD starting with the sample file, a description of the parameters available, and examples of how they are used.

Note: If you are using KumoScale in Appliance mode, we recommend that all masters of the KumoScale storage cluster, specified by the numberOfMasters in the master CR, be established before you configure nodes for storage.

To create your custom storage node CR:

- Make a copy of **KumoScale_Operator/ks-config-operator/samples/kumoscale_v1_storagenode_cr.yaml** for editing, and save to a separate location. For example, we will save it to the file **deploy/crds/myapp_storagenode_cr.yaml**.
- For each storage node you want to create, update **myapp_storagenode_cr.yaml** with values for the parameters listed in [Storage Node Parameters](#).
- Save your file and go to Step 2.

Step 2. Create Storage Nodes

The create command creates the storage node for the first time and adds it to the KumoScale Provisioner service on the KumoScale storage cluster. If the storage node already exists, you will receive an error.

Note: You will need both a valid license and a secret to create storage nodes. Ensure that you have entered a valid license key before proceeding with this step. If you have the KumoScale software secret but not a valid license, your storage node will be created but will not be registered with the KumoScale Provisioner service and you will not be able to use it for provisioning.

To create the storage node with name = myapp_storagenode_1 defined in the CRD file myapp_storagenode_cr.yaml, enter the following:

```
kubectl create -f myapp_storagenode_cr.yaml
```

To verify that the node was created and added to the KumoScale Provisioner service, enter:

```
kubectl get storagenodes
kubectl describe storagenodes myapp_storagenode_1
```

The storage node will not be added to the KumoScale Provisioner service if one of the parameter values is invalid. In the event of a failure, you will get an error message identifying the parameters causing the failure. Update the relevant parameter in the CRD and use the **apply** command to apply the changes.

```
kubectl apply -f myapp_storagenode_cr.yaml
```

Note: Once the storage node is added to the KumoScale Provisioner service, you will not be able to modify it except for cases noted in [Update Storage Nodes](#).

State and Status of Storage Nodes

There are several **kubect**l commands for storage nodes and services that return STATE and STATUS of the nodes. There is a difference between these two fields. The

- **STATE** reflects the current state of the storage node. It is dynamically reported by the KumoScale Provisioner and reflected by the operator.
- **STATUS** reflects the status of the result of the last user operation done on a storage node CRD. For example; create a new storage node CRD or update an existing one.

Possible values for State and Status are Available, Dead, Pending, and Unavailable.

For example, **kubect**l **get storagenodes** presents the STATE, which is the real STATE of the storage node:

```
Every 1.0s: kubectl get storagenodes
NAME      AGE      STATE      ROLE      VERSION      ALERTS
ks-node1  5h45m    Available  MASTER    3.18-14370    0
ks-node2  5h45m    Available  MASTER    3.18-14370    0
ks-node3  5h45m    Available  MASTER    3.18-14370    0
ks-node4  5h42m    Available  WORKER    3.18-14370    0
ks-node5  5h42m    Available  WORKER    3.18-14370    0
ks-node6  5h42m    Available  WORKER    3.18-14370    0
ks-node7  5h42m    Unavailable WORKER    3.18-14370    0
```

kubectl **get storagenodes -o wide** presents the STATE of the storage node and the STATUS of the last user operation on this storage node:

```
Every 1.0s: kubectl get storagenodes -o wide
NAME      AGE      STATE      ROLE      NODE NAME      VERSION      ALERTS      CAPACITY      AVAILABLE CAPACITY      IP      STATUS
ks-node1  5h33m    Available  MASTER    ks-node1-000c29cd26f2  3.18-14370    0           76861         74761                    172.28.10.11  Success
ks-node2  5h33m    Available  MASTER    ks-node2-000c299e2b4d  3.18-14370    0           171           98261                    172.28.11.85  Success
ks-node3  5h33m    Available  MASTER    ks-node3-000c29278609  3.18-14370    0           171           98361                    172.28.11.215  Success
ks-node4  5h30m    Available  WORKER    ks-node4-000c2908b9fc  3.18-14370    0           171           102361                   172.28.10.63  Success
ks-node5  5h30m    Available  WORKER    ks-node5-000c290744f1  3.18-14370    0           153661        153561                   172.28.10.177  Success
ks-node6  5h30m    Available  WORKER    ks-node6-000c290f6062  3.18-14370    0           271           204761                   172.28.10.00  Success
ks-node7  5h30m    Unavailable WORKER    ks-node7-000c294bb3b7  3.18-14370    0           25661         25561                    172.28.11.158  Success
```

kubectl **describe storagenode <name>** presents the latest STATUS information on the storage node.

Storage Node Parameters

This section defines all the parameters used with the storage node CRD. Examples of storage nodes using these parameters is provided in [Example Storage Node CRD](#).

Required Parameters

The table below shows system level parameters you must specify for each distinct storage node that you deploy. The **apiVersion** and **kind** parameters which appear at the top of the file should not be modified.

Parameter	Description
name	Unique name for the storage node. It must comply with the Name rules (see KumoScale Field Types)
initMgmtIp	The initial management IP address.
adminSecretName	Name of the secret file created during installation.
groupName	Name of the group. It can remain with the default value.
timeSettings: timeZoneID:	The time zone to use on the storage node. (Appliance mode only; does not apply to Managed mode)
timeSettings: mode:	Mode can be NTP or Manual (Appliance mode only; does not apply to Managed mode) <ul style="list-style-type: none">• NTP: You must also specify a value for ntpServer. It must be reachable by ping otherwise the command will fail. We recommend using this configuration when running a cluster of storage nodes, to synchronize the time across storage nodes.• Manual: Time will be returned in YYYY-MM-DDThh:mm:ss format.
timeSettings: ntpServer:	Specify only when mode:NTP as explained above (Appliance mode only; does not apply to Managed mode)
AuthenticationMode	The authentication mode; select one of Local , Ldap , OPENIDC

See [Example Storage Node CRD](#) for examples using these parameters.

Notes

1. KumoScale requires all nodes to be synchronized as some sites may be supported by external NTP servers, and others by an internal NTP server. To avoid issues related to node timers, validate synchronization across cluster components.
2. Before deploying a storage node, all SSDs must be attached to it.

Network Portal Parameters

You must configure at least one NVMe-oF portal before provisioning storage. Use this section to configure a data interface for the application initiators (hosts) to connect to their targets. See the following sections for details on the configuration you need to support; note that not all configurations are available in Managed mode:

- [Consolidated Data and Management Interfaces](#)
- [Network Management IP Parameters](#)
- [Network Interface Parameters: Maximum Transmission Unit \(MTU\) or Link State \(Appliance Mode only\)](#)
- [Network VLAN and LACP Configuration \(Appliance Mode only\)](#)
- [Network VLAN Configuration \(Appliance mode only\)](#)
- [Storage Node with BGP \(Appliance mode\)](#)
- [Topological Parameters](#)

Consolidated Data and Management Interfaces

Most platforms provide two sets of interfaces for connectivity:

- Data interfaces** are used for data traffic, and storage is accessed via block, file, or object storage protocols.
- Management interfaces** are used by administrators to connect to, and manage, data or storage nodes. Some platforms provide one or more dedicated management interfaces, and these cannot be re-purposed for data traffic. On the other hand, some platforms do not provide any management interfaces, and administrators need to access node management capability via data interfaces.

KumoScale software has qualified certain platforms that do not have a dedicated management interface. In this case, KumoScale software can use a consolidated data and management interface, which performs both functions.

Parameter	Description	Optional/Required
portal:		
ip	The IP address for the portal. Appliance mode supports ipv4. Managed mode only supports ipv4 for existing address in ipv4.	Required
subnet	The IP address subnet mask.	Required
interface	The interface to host the IP address. A list of available interfaces may be obtained using the Cluster Manager CLI <i>interface-show</i> command.	Required
port	The port for the portal. For valid port numbers see KumoScale Field Types .	Required
name	A unique portal name that must comply with the Name rules (see KumoScale Field Types).	Required
transportType	TCP_IP or RoCEv2 (for RDMA).	Required

See [Example Storage Node CRD](#) for examples using these parameters.

Network Management IP Parameters

Parameter	Description	Optional/Required
mgmtIps:		
interface	Interface name	Required
mode	Can be DHCP (default) or Static (DHCP = Dynamic Host Configuration Protocol)	Static, when IP is static DHCP, otherwise
ipAddress	IP address	Required when mode is Static
mask	Mask	Required when mode is Static
defaultGateway	The storage node default gateway IP	Required when mode is Static
dnsServer	DNS address	Required

See [Example Storage Node CRD](#) for examples using these parameters.

Network Interfaces Parameters: Maximum Transmission Unit (MTU) or Link State (Appliance Mode only)

To set the network interface’s link state or the MTU you need to specify the interface name and desired values.

Note: You can set the MTU for external network interfaces but you cannot set the MTU internal interfaces which are used in HA systems.

Parameter interfaces:	Description	Optional/Required
name:	The interface name. It must comply with the Name rules (see KumoScale Field Types)	Required

Parameter interfaces:	Description	Optional/ Required
mtu	The interface MTU; may be 1500 (default), 4200, or 9000. For network interfaces that use: <ul style="list-style-type: none"> RoCE transport type, the optional MTU values are: 1500, 4200, or 9000. TCP_IP transport type, the optional MTU values may be: 1500 or 9000. 	Optional
adminState	The current administrative (link) state of the interface. The value is either UP or DOWN.	Optional

See [Storage Node with MTU and Link State](#) for an example using these parameters.

Network VLAN and LACP Configuration (Appliance Mode only)

KumoScale software in Appliance mode supports the Link Aggregation Control Protocol (LACP) of data ports to provide network resiliency. This is implemented with data ports teaming for Transmission Control Protocol (TCP) transport and bonding NIC ports for Remote Direct Memory Access (RDMA). Teaming is done by setting team parameters in the storage node CRD.

The following constraints must be followed when configuring LACP:

- Port configuration must be constant during an active session.
- Data ports in an application must all be teamed/bonded or not at all.
- Teamed NIC ports must be identical (e.g. the same maximal speed, product, or vendor).
- The team member interface name or interface id may be specified, but not both.

teams Parameter Name	Description	Optional/Required
name:	Team interface name, It must comply with the Name rules (see KumoScale Field Types). You may specify the name or ID but not both; name is recommended.	Either name or id is required but not both; name is recommended
members: id:	A list of at least two unused network interfaces, with its ID. You may specify the name or id but not both; name is recommended.	Either name or id is required but not both; name is recommended
tags	Supported tag names: tx_hash : value should be a list of comma-separated values from: <ul style="list-style-type: none"> eth: Uses source and destination MAC addresses. vlan: Uses VLAN id. ipv4: Uses source and destination IPv4 addresses. ipv6: Uses source and destination IPv6 addresses. ip: Uses source and destination IPv4 and IPv6 addresses. I3: Uses source and destination IPv4 and IPv6 addresses. tcp: Uses source and destination TCP ports. udp : Uses source and destination UDP ports. sctp: Uses source and destination SCTP ports. I4: Uses source and destination TCP and UDP and SCTP ports. tx_balancer_name : If the user wants to configure a balancer, a balancer name should be provided. The only value supported for now is basic . tx_balancing_interval : An integer specifying the balancing interval in tenths of seconds.	Optional

See [Example Storage Node CRD](#) for examples using these parameters.

Network VLAN Configuration (Appliance mode only)

KumoScale in Appliance mode allows you to configure a VLAN by specifying VLAN information. VLAN tagging may also be applied to a teamed interface, e.g., VLAN over LACP. KumoScale software allows up to sixteen (16) VLAN tags per storage node. Once the VLAN interface is created, a [portal interface](#) may be created over it.

The following constraints must be followed when configuring VLAN:

- Port configuration must be constant during an active session.
- VLAN configuration may not be changed while there is an active session.

vians Parameter Name	Description	Optional/Required
interface	Name of the physical interface	Required
tag	The VLAN tag maximum: 4094 (default), minimum:1	Required

See [Example Storage Node CRD](#) for examples using these parameters.

Network Border Gateway Protocol (BGP) Configuration (Appliance Mode only)

KumoScale in Appliance mode includes support for the Border Gateway Protocol (BGP) implemented via integration of the Free Range Routing (FRR) network routing software. It delivers multipath networking for NVMe-oF storage over TCP/IP networks. A Clos network topology is often used by data center operators to build high performance, scalable, cost effective, and robust networks. Such networks use IP routing as the primary packet forwarding mechanism, and BGP is the most popular routing protocol used in this type of environment. Storage system support for the BGP protocol enables storage resources to participate as a first-class citizen in a Clos network, allowing resilient, high bandwidth connectivity between initiators and storage targets. Traditional storage interconnect uses layer 2 technologies, such as port channels, to connect into IP networks. By participating instead at layer 3 (i.e. IP routing), KumoScale storage systems enter the modern data center network as a native cloud service. Running BGP as the routing protocol allows KumoScale storage systems to provide reliable and dynamically reroutable L3 level multipath network connectivity between initiators and KumoScale storage targets.

BGP-based multipath routing enables a network to support resilient connectivity, higher capacity and performance through load balancing, improvement of the timeliness of response to network path changes, and enhancement of system resilience and security in the face of failures and attacks.

KumoScale software supports two types of BGP configurations: unnumbered and numbered.

- BGP unnumbered: KumoScale software storage nodes are configured with Free Range Routing (FRR) such that storage nodes can be deployed with copy-paste FRR configuration files that simply specify the nearest Tor neighbors. KumoScale implements target-side virtual IPs to provision volumes to these initiators while allowing network paths to change beneath storage initiators and targets.
- BGP numbered: KumoScale software storage nodes are configured with static IPs; with each NIC mapped to a virtual IP. Note that you will need to add a masquerade rule to the IP tables of the initiator machine. The reason for this is that with numbered BGP, the source IP is the BGP peer IP which is internal; masquerade will make packets look like they were sent from the external IP. For example, if the BGP peers on the initiator machine are `###.##.##.1/31` on eth4 and `###.##.##.3/31` on eth5, the rules would be one of the following. For:

Interface level general:

```
iptables -t nat -A POSTROUTING -o eth4 -j MASQUERADE
iptables -t nat -A POSTROUTING -o eth5 -j MASQUERADE
```

IP level precise:

```
iptables -t nat -A POSTROUTING -s ###.##.##.1 -j MASQUERADE
iptables -t nat -A POSTROUTING -s ###.##.##.3 -j MASQUERADE
```

IP range less precise, shorter:

```
iptables -t nat -A POSTROUTING -s ###.##.##.0/24 -j MASQUERADE
```

The user can configure a BGP portal by specifying the information below in the storage node CRD. Example configurations are provided in [Storage Node with BGP Unnumbered](#) and [Storage Node with BGP Numbered](#).

Parameter	Description	Optional/Required
bgpportals:		
asn	The portal Autonomous System Number (ASN). The value should be an integer in the range 1–23455, 23457–64495, 64512-65534, 131072–4294967294.	Required
ip	The IP address for the portal.	Required
members: interface:	The net interface name for the member of the BGP portal. This should adhere to name field requirements specified in KumoScale Field Types .	Required
members: ip	IP address for the member.	Required for numbered configurations
members: mask	Mask for the member.	Required for numbered configurations
members: neighborASN	The neighbor ASN.	Required for numbered configurations
members: neighborIP	The neighbor IP.	Required for numbered configurations
mode	The portal mode. Equal to either Unnumbered or Numbered.	Required
name	Name for the portal. This should adhere to name field requirements specified in see KumoScale Field Types	Required
port	Port number for the portal. An integer between 4420 (default) and 65535.	Required
subnet	The subnet for the portal IP.	Required
transportType	The transport type of the portal. Only TCP_IP is currently supported.	Optional

Topological Parameters

Topological parameters are used for volume placements and are specified as pairs. You can specify one or all of rack, zone, and region.

topology Parameter Name	Description	Optional/

		Required
name	Can be one of topology.kubernetes.io/rack, topology.kubernetes.io/zone, topology.kubernetes.io/region, or kubernetes.io/hostname.	Optional
value	Rack, zone, or region depending on value of name .	Optional

See [Example Storage Node CRD](#) for examples using these parameters.

Example Storage Node CRD

This section includes examples of storage node CRD and the purpose of specific settings. For security reasons, valid IP addresses, subnets, host IDs, UUIDs, licenses, and tokens are **not** provided in KumoScale documentation.

Storage Node using NTP, TCP_IP Portal, Rack and Zone Topology

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: ks-node1-example
spec:
  initMgmtIp: 192.0.20.0
  adminSecretName: kumoscale-secret
  groupName: group1
  timeSettings:
    timeZoneID: Asia/Jerusalem
    mode: NTP
    ntpServer: 172.###.###.###
  network:
    portals:
      - ip: 192.0.20.1
        name: portal1
        subnet: 255.255.0.0
        interface: kx0
        port: 4420
        transportType: TCP_IP
  topology:
    - name: topology.kubernetes.io/rack
      value: "RACK1"
    - name: topology.kubernetes.io/zone
      value: "LAB"
```

Storage Node with MTU and Link State (Appliance mode only)

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: storagenode-test
spec:
  initMgmtIp: 192.0.2.0
  adminSecretName: kumoscale-secret
  groupName: group1
  timeSettings:
    timeZoneID: Asia/Jerusalem
    mode: NTP
    ntpServer: time.google.com
  network:
    interfaces:
      - name: kx2
        mtu: b9000
        adminState: UP
    portals:
      - ip: 192.0.2.1
        name: portal20
        subnet: 255.255.0.0
        interface: kx2
        port: 4420
        transportType: TCP_IP
  topology:
    - name: topology.kubernetes.io/rack
      value: "RACK1"
    - name: topology.kubernetes.io/zone
      value: "LAB"
```

Storage Node with Management IP on Team (Appliance Mode only)

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: ks-node2-example
spec:
  initMgmtIp: 192.0.2.0
  adminSecretName: kumoscale-secret
  groupName: group1
  timeSettings:
    timeZoneID: Asia/Jerusalem
    mode: Manual
    ntpServer: None
  network:
    portals:
      - ip: 192.0.2.1
        name: portal1
        subnet: 255.255.0.0
        interface: kx0
        port: 4420
        transportType: TCP_IP
    mgmtIps:
      - interface: team1
        ipAddress: 30.###.###.###
```

```
      mask: 255.255.255.0
      mode: STATIC
    teams:
      - members:
          - name: kx1
          - name: kx2
        name: team1
  topology:
    - name: topology.kubernetes.io/rack
      value: "RACK1"
    - name: topology.kubernetes.io/zone
      value: "LAB"
```

Storage Node with Management IP on VLAN (Appliance Mode only)

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: ks-node2-example
spec:
  initMgmtIp: 172.##.##.###
  adminSecretName: kumoscale-secret
  groupName: group1
  timeSettings:
    timeZoneID: Asia/Jerusalem
    mode: Manual
    ntpServer: None
  network:
    portals:
      - ip: 172.##.##.###
        name: portal1
        subnet: 255.255.0.0
        interface: kx0
        port: 4420
        transportType: TCP_IP
    mgmtIps:
      - interface: kx1.9
        ipAddress: 30.##.##.###
        mask: 255.255.255.0
        mode: STATIC
    vlans:
      - interface: kx1
        tag: 9
  topology:
    - name: topology.kubernetes.io/rack
      value: "RACK1"
    - name: topology.kubernetes.io/zone
      value: "LAB"
```

Storage Node with Portal on Physical Interface

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: ks-node3-example
spec:
  initMgmtIp: 10.###.##.##
  adminSecretName: kumoscale-secret
  groupName: p1
  timeSettings:
    timeZoneID: Asia/Kolkata
    mode: NTP
    ntpServer: 172.##.##.##
  network:
    portals:
      - ip: 192.##.##.##
        name: portal1
        subnet: 255.255.255.0
        interface: kx4
        port: 4420
        transportType: RoCEv2
  topology:
    - name : kubernetes.io/hostname
      value : "ks-node2-example"
```

Storage Node with Portal on VLAN (Appliance mode only)

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: ks-node4-example
spec:
  initMgmtIp: 10.###.##.##
  adminSecretName: kumoscale-secret
  groupName: p1
  timeSettings:
    timeZoneID: Asia/Kolkata
    mode: NTP
    ntpServer: 172.##.##.##
  network:
    teams:
    vlans:
      - interface: kx4
        tag: 9
    portals:
      - ip: 30.##.##.##
        name: portal1
        subnet: 255.255.255.0
        interface: kx4.9
        port: 4420
        transportType: TCP_IP
  topology:
```

```
- name : kubernetes.io/hostname
  value : "ks-node2-example"
```

Storage Node with Portal on VLAN on Team (LACP) (Appliance Mode only)

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: ks-node2-example
spec:
  initMgmtIp: ##.###.##.##
  adminSecretName: kumoscale-secret
  groupName: p1
  timeSettings:
    timeZoneID: Asia/Kolkata
    mode: NTP
    ntpServer: 172.##.###.##
  network:
    teams:
      - name: team1
        members:
          - name: kx1
          - name: kx2
        tags:
          - name: tx_hash
            value: eth,ipv4,ipv6
          - name: tx_balancer_name
            value: basic
          - name: tx_balancing_interval
            value: "70"
    vlans:
      - interface: team1
        tag: 9
    portals:
      - ip: 30.##.###.##
        name: portal1
        subnet: 255.255.255.0
        interface: team1.9
        port: 4420
        transportType: TCP_IP
  topology:
    - name : kubernetes.io/hostname
      value : "ks-node2-example"
```

Storage Node with BGP Unnumbered (Appliance mode)

```
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: bgp-unnumbered-example
spec:
  initMgmtIp: 172.##.###.##
  adminSecretName: kumoscale-secret
  groupName: group1
  timeSettings:
    timeZoneID: Asia/Jerusalem
    mode: Manual
    ntpServer: None
  network:
    bgp_portals:
      - asn: 64005
        ip: 10.##.###.##
        members:
          - interface: kx1
          - interface: ens161
        mode: Unnumbered
        name: bgp_portal1
        port: 7980
        subnet: 255.255.0.0
        transportType: TCP_IP
```

Storage Node with BGP Numbered (Appliance mode)

Remember that you will also need to provide masquerade rules for the IP tables of the initiator (host) machine.

```
kumoscale_v1_storagenode_cr_numbered.yaml:
Storage Node numbered
apiVersion: kumoscale.kioxia.com/v1
kind: StorageNode
metadata:
  name: demo
spec:
  initMgmtIp: 172.##.###.##
  adminSecretName: kumoscale-secret
  groupName: group1
  timeSettings:
    timeZoneID: Asia/Jerusalem
  mode: Manual
  ntpServer: None

  network:
    bgp_portals:
      - asn: 64005
        ip: 10.##.###.##
        members:
          - interface: kx1
            ip: 10.##.###.##
            mask: 255.255.0.0
            neighborASN: 63005
            neighborIP: 10.##.###.##
          - interface: ens161
            ip: 10.##.###.##
            mask: 255.255.0.0
            neighborASN: 62005
```

```
neighborIP: 10.##.##.##
mode: Numbered
name: bgp_portal1
port: 7980
subnet: 255.255.0.0
transportType: TCP_IP
```

Updating Storage Nodes

You can change the configuration of a storage node by using **kubectl** to edit the CRD file that defined the particular storage node and then apply the change. For example, to modify the storage node called **myapp_storagenode_1** defined by **myapp_storagenode_cr.yaml**:

1. Edit the CRD file **myapp_storagenode_cr.yaml**:

```
kubectl edit -f myapp_storagenode_cr.yaml
```

2. Change the settings and save the file. Then enter the command:

```
kubectl apply -f myapp_storagenode_cr.yaml
```

3. Run the following command to verify the node has the desired configuration:

```
kubectl describe storagenodes myapp_storagenode_1
```

Deleting Storage Nodes

You can use **kubectl** to delete a storage node only when it is not currently undergoing an update and the status of the node is either:

- Dead, or
- Available and it has no volumes attached.

The system will remove any Syslog configured on the server.

For example, to delete the storagenode **myapp_storagenode_1** defined by **myapp_storagenode_cr.yaml**, enter:

```
kubectl delete -f myapp_storagenode_cr.yaml
```

Next, Managed Mode: [Authentication](#)

Next, Appliance Mode: [Role-Based Access Control \(RBAC\)](#)