

KumoScale Provisioner REST API Command Reference

The KumoScale™ Provisioner REST API is the primary public interface to KumoScale software. It allows users to manage the pool of KumoScale storage nodes, and to allocate, connect, monitor, and manage storage volumes.

Inventory Management

The inventory management APIs provide the means for maintaining the pool of KumoScale storage nodes (also known as backends), available for volume allocation. This includes monitoring the pool's health, maintaining its coherency and persistency, and extracting the pool's information. Creating and modifying the pool is done via operators.

Add Backend

Add a KumoScale storage node (backend) to the backend pool.

| Method | URI |
|----------------------------|--|
| POST | /backends |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"mgmtIPs": ["###.##.###.###"], "rack": "Rack2", "region": "Region2", "zone": "Zone2", "hostId": "#####-####-####-#####", " mgmtPort ": 443}' https://###.##.###.###:30100/backends |

Normal response codes: 200

Request

No URI parameters.

Example: *Add Backend*: JSON resquest

```
{
  "mgmtIPs": [ "###.##.###.###" ],
  "rack": "Dev",
  "region": "ISR",
  "zone": "Lab",

  "hostId": "#####-#####-#####-#####",
  "mgmtPort": 443,

}
```

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|---|
| mgmtIPs | String | Mandatory | KS Management IP or DNS name. |
| rack | String | Optional | KS location Rack ID |
| Zone | String | Optional | Accessible Zone ID (e.g., shared switch) |
| region | String | Optional | Accessible Group ID (e.g., shared cage) |
| hostId | String | Optional | If backend (storage node) is running within a host (initiator), the hostId, otherwise should not be given |
| mgmtPort | Int | optional | The backend (storage node) port which will be used to access the backend. Default: 443. |

Response

Example: *Add Backend*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description |
| Status | String | Result code. <ul style="list-style-type: none">• <i>Success</i> - Success• <i>AlreadyExists</i> - Backend already exists with mgmtIPs• <i>NoGroups</i> - Backend has no configured groups• <i>InvalidLocationValue</i> - Rack/Zone/Region has more than 255 chars• <i>NotAvailable</i> - Backend is incompatible, or the persistent ID was not provided• <i>BackendWithHostIdAlreadyExists</i> – This host id is used by another backend.• KS Set Provisioner API result codes |

Update Backend

Update properties of an existing KumoScale storage node (backend) in the pool.

| Method | URI |
|----------------------------|--|
| PUT | /backends/{id} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X PUT -H 'Content-Type: application/json' -d '{"rack": "Rack2", "region": "Region2", "zone": "Zone2", "hostId": "#####-#####-#####-#####" , " mgmtPort ": 444 }' https://##.##.###.####:30100/backends/apCMak3chDoyCPJf |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|-----------------------------------|
| Id | String | Mandatory | KS persistent ID (~ MAC address). |

Example: *Update Backend*: JSON request

| |
|---|
| { "inUse": true, "rack": "Dev", "region": "ISR", "zone": "Lab" "hostId": "#####-#####-#####-#####" , " mgmtPort ": 444 } |
|---|

| Parameter Name | Type | Is Mandatory | Description |
|----------------|---------|--------------|--|
| rack | String | Optional | KS location Rack ID , default is null |
| Zone | String | Optional | Accessible Zone ID (e.g., shared switch) , default is null |
| region | String | Optional | Accessible Group ID (e.g., shared cage) , default is null |
| inUse | Boolean | Optional | Is it used for volume management |
| hostId | String | Optional | If backen, the hostId, otherwise should not be given |
| mgmtPort | Int | optional | The storage node port which will be used to access the storage node. |

Response

Example: *Update Backend*: JSON response

| |
|--|
| { "description": "Success", "status": "Success" } |
|--|

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description |
| Status | String | Result code. <ul style="list-style-type: none">• <i>Success</i> - Success• <i>NotExists</i> – A storage node (backend) with the required id does not exist.• <i>InvalidLocationValue</i> - Rack/Zone/Region has more than 255 chars |

| Parameter Name | Type | Description |
|----------------|------|---|
| | | <ul style="list-style-type: none"><i>BackendWithHostIdAlreadyExists</i> – This host id is used by another storage node (backend).KS Set Provisioner API result codes |

Remove Backend

Remove a KumoScale storage node (backend) from the pool

| Method | URI |
|----------------------------|---|
| DELETE | /backends/{id}?force=false |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE --header "Content-Type: application/json" https://###.###.###.###:30100/backends/##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------------------------------|-------------------------|--|
| Id | String | Mandatory | KS persistent ID (~ MAC address). |
| force | String – Boolean true or false | Optional, default false | If true, force the deletion of the backend even if it has volumes on it and its in Unavailable state |

Response

Example: *Remove Backend*: JSON response

| |
|--|
| <pre>{ "description": "Success", "status": "Success" }</pre> |
|--|

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description |
| Status | String | Result code. <ul style="list-style-type: none"><i>Success</i> - Success<i>NotExists</i> – A storage node (backend) with the required id does not exist.<i>BackendHasVolumes</i> - There are configured volumes on the backend.KS Set Provisioner API result codes |

List Backends

Lists all of the KumoScale storage nodes (backends), currently in the Provisioner’s pool, and their state.

| Method | URI |
|----------------------------|--|
| GET | /backends |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/backends |

Normal response codes: 200

Request

No URI parameters.

Response

Example 1: List Backends: JSON response

| |
|---|
| <pre>{ "persistentID": "82:7b:62:b4:12:c4", "mgmtIPs": ["###.###.###.###", "fe80::c696:5290:b65:c0a5"], "mgmtPort": 443, "rack": "Dev1", </pre> |
|---|

```
    "zone": "Israel",

    "region": "Dev1",

    "inUse": true,

    "state": "Available",

    "totalCapacity": 1099511627776,

    "availableCapacity": 0,

    "numVolumes": 0,

    "lastProbTime": 1633005301913,

    "probeInterval": 60,

    "totalBW": 4525654016,

    "availableBW": 4525654016,

    "totalIOPS": 348600,

    "availableIOPS": 348600,

    "version": "3.22.1b-15757",

    "numberOfSSDs": 2,

    "certified": true,

    "portals": [

        {

            "ip": "###.##.##.###",

            "port": 4420,

            "transport": "TCP_IP"

        }

    ],

    "name": "KS1",

    "authenticationMode": "OPEN_IDC"

}
```

| Parameter Name | Type | Description |
|-------------------|--------------|--|
| persistentID | String | KS persistent ID (~ MAC address). |
| mgmtIPs | List<String> | KS Management IP or DNS name. |
| rack | String | KS location Rack ID. (not reported in case not configured) |
| zone | String | Accessible Zone ID (e.g., shared switch). (not reported in case not configured) |
| region | String | Accessible Group ID (e.g., shared cage). (not reported in case not configured) |
| inUse | Boolean | Is it used for volume management. |
| state | enum | Available / Unavailable. |
| totalCapacity | long | KS current total capacity - The sum of all SSDs capacity. |
| availableCapacity | long | KS sum of all available space. |
| lastProbTime | long | Last probe time. (not reported in case probe was not received yet for this node) |
| probeInterval | int | Probe interval in secs. |
| totalBW | long | KS total potential bandwidth in B/s. |
| availableBW | long | KS available bandwidth in B/s (unused out of potential value). |
| totalIOPS | long | KS total potential IOPS. |
| availableIOPS | long | KS available IOPS (unused out of potential value). |
| portals | List | A list of portals. Each entry includes the following parameters: |

| Parameter Name | Type | Description |
|---------------------|---------|--|
| | | <ul style="list-style-type: none">ip – The ip address of the portal.port – The port of the portal.transport – The transport type of the portal. |
| hostId | String | The host id. (not reported in case no csi driver exists on the storagenode) |
| mgmtPort | int | The backend port which will be used to access the backend. |
| targetDriverVersion | String | The version of the target driver at this backend |
| numVolumes | Int | Total number of volumes. |
| certified | Boolean | true if the backend is certified. The backend is considered certified if all its SSDs are certified. |
| version | String | The backend's version. |
| numberOfSSDs | int | Number of ssds. |
| name | string | The backend's name |
| authenticationMode | Enum | Current backend configured authentication mode (LOCAL , LDAP, OpenIDC) |

Get Backend by ID

Retrieve a KumoScale storage node (backend) and its details from the Provisioner's pool according to its ID.

| Method | URI |
|----------------------------|--|
| GET | /backends |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.##.##.###:30100/backends/##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|-----------------------------------|
| id | String | Mandatory | The persistent ID of the backend. |

Response

Example 2: Get Backend by Id: JSON response

```
{
  "persistentID": "82:7b:62:b4:12:c4",
  "mgmtIPs": [
    "###.##.##.#",
    "fe80::c696:5290:b65:c0a5"
  ],
  "mgmtPort": 443,
  "rack": "Dev1",
  "zone": "Israel",
  "region": "Dev1",
  "inUse": true,
  "state": "Available",
  "totalCapacity": 109951162776,
  "availableCapacity": 0,
  "numVolumes": 0,
  "lastProbTime": 1633005781901,
  "probeInterval": 60,
  "totalBW": 4525654016,
  "availableBW": 4525654016,
  "totalIOPS": 348600,
```

```
    "availableIOPS": 348600,

    "version": "3.22.1b-15757",

    "numberOfSSDs": 2,

    "certified": true,

    "portals": [

        {

            "ip": "192.20.10.1",

            "port": 4420,

            "transport": "TCP_IP"

        }

    ],

    "name": "KS1"

}
```

| Parameter Name | Type | Description |
|-------------------|--------------|--|
| persistentID | String | The persistent ID of the backend. |
| mgmtIPs | List<String> | A list of KumoScale (KS) Management IPs or DNS names. |
| rack | String | KS location Rack ID. (not reported in case not configured) |
| zone | String | Accessible Zone ID (e.g., shared switch). (not reported in case not configured) |
| region | String | Accessible Group ID (e.g., shared cage). (not reported in case not configured) |
| inUse | Boolean | Is it currently in use by the Provisioner. |
| state | enum | Is the backend Available / Unavailable. |
| totalCapacity | long | KS total capacity - The sum of all SSDs capacity. |
| availableCapacity | long | KS sum of all available space. |
| lastProbTime | long | Last probe time. (not reported in case probe was not received yet for this node) |
| probeInterval | int | Probe interval in secs. |
| totalBW | long | KS total potential bandwidth in B/s. |
| availableBW | long | KS available bandwidth in B/s (unused out of potential value). |
| totalIOPS | long | KS total potential IOPS. |
| availableIOPS | long | KS available IOPS (unused out of potential value). |
| portals | List | <ul style="list-style-type: none">ip – The portal ip addressport – The portal porttransport – The portal transport type |
| hostId | String | The host id. (not reported in case no csi driver exist on the storagenode) |
| mgmtPort | int | The backend port which will be used to access the backend. |
| numVolumes | Int | Total number of volumes. |
| certified | Boolean | true if the backend is certified. The backend is considered certified if all its SSDs are certified. |
| version | String | The backend’s version. |
| numberOfSSDs | int | Number of ssds. |
| name | string | The backend’s name |

Get Info

Returns the total of the free capacities on all the KumoScales in the Provisioner’s pool.

| Method | URI |
|----------------------------|---|
| GET | /info |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://##.##.##.###:30100/info |

Normal response codes: 200

Request

No URI parameters.

Response

Example 3: Get Info: JSON response

```
{

    "totalFreeSpace": 1920315949046,

    "version": "1.1",

    "syslogsBackend": "0c:c4:7a:88:09:ee" ,

}
```

```
"authenticationMode": "OPEN_IDC"
}
```

| Parameter Name | Type | Description |
|--------------------|--------|---|
| totalFreeSpace | long | The total free space in bytes (on all KumoScale storage nodes). |
| version | String | Provisioner software version. |
| syslogsBackend | String | The persistentID of the backend the syslog config was based on. |
| authenticationMode | Enum | Current Provisioner configured authentication mode (JWT/OpenIDC) |

Probe

KumoScale makes this call to notify the Provisioner that it is up and running and to update it with its properties.

| Method | URI |
|----------------------------|---|
| POST | /probe |
| Linux Curl Command Example | <pre>curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"availableCapacity": 1920315949046, "usedBW": 1034998978304, " totalBW ": 1920315949046, "usedIOPS": 120903849098, " totalIOPS ": 1920315949046, "mgmtIPs": ["###.##.##.###"], "provisioner": { "inUse": true, "pushInterval": 60, "url": <protocol>://<ip/host>:<port>. "tags": [{ "name": "rack", "value": "Dev" }], "authorizationServerPublicKey":"BIGLONGSTRING", "authorizationServerTokenURL":"http://###.##.##.###:8080/auth/realms/Dev/protocol/openid-connect "backendsClientID":"liorf-backend", "backendsResourceID":"liorf-backend", "backendsClientSecret":"#####-####-####-####-#####" }, " mgmtPort ": 443, "persistentID:" ""ce:10:66:d3:a2:dd", "targetDriverVersion": "3.22.1b-15757", "totalCapacity": 4000315949046, "version": "3.17.13465, "numberOfSSDs": 2, "Certified ": true, "changes": 0, "authenticationMode": "OPEN_IDC"}'</pre> |

Normal response codes: 200

Request

No URI parameters.

Example 1: *Probe*: JSON request

```
{
  "availableCapacity": 1920315949046,
  "usedBW": 100000000000,
  "totalBW": 1920315949046,
  "usedIOPS": 129039487386,
  "totalIOPS": 1920315949046,
  "mgmtIPs": [ "###.##.##.###" ],
  "provisioner": {
```

```
    "inUse": true,

    "url": "https://###.##.##.###:30100",

    "pushInterval": 60,

    "tags": [ { "name": "rack", "value": "Dev" },

               { "name": "storageClassName", "value": "sc1" },

               { "name": "shareSSDBetweenVolumes", "value": "true" }    ],

    "authorizationServerPublicKey":"BIGLONGSTRING",

    "authorizationServerTokenURL":"http://###.##.##.###:8080/auth/realms/Dev/protocol/openid-connect/token",

    "backendsClientID":"liorf-backend",

    "backendsResourceID":"liorf-backend",

    "backendsClientSecret":"#####-####-####-####-#####"

  },

  "totalCapacity": 4000315949046,
  "version": "3.17.13465,

  " numberOfSSDs ": 2,

  " Certified ": true,

  " mgmtPort ": 443,

  "targetDriverVersion": "3.22.1b-15757",

  “persistentID:” “"ce:10:66:d3:a2:dd",

  “changes”: 0,

  "authenticationMode": "OPEN_IDC"}
```

| Parameter Name | Type | Is Mandatory | Description |
|--|--------------|--------------|--|
| availableCapacity | long | Mandatory | The sum of the available space in the appliance. |
| usedBW | long | Mandatory | The sum of the used bandwidth in the appliance in B/s. |
| totalBW | long | Mandatory | KS total bandwidth in B/s. |
| usedIOPS | long | Mandatory | The sum of used IOPS in the appliance. |
| totalIOPS | long | Mandatory | The total potential IOPS. |
| mgmtIPs | List<String> | Mandatory | KS Management IPs or DNS name. |
| totalCapacity | long | Mandatory | KS total capacity - the sum of all SSDs capacity. |
| persistentID | String | Mandatory | A persistent ID string of this engine. |
| provisioner | Provisioner | Mandatory | |
| inUse (provisioner) | Boolean | Mandatory | Is it used for volume management. |
| pushInterval (provisioner) | int | Mandatory | Interval between consequence probe calls in seconds. Range [60 - 86400]. |
| url (provisioner) | String | Mandatory | Provisioner url in the format: <protocol>://<ip/host>:<port>.ing |
| authorizationServerPublicKey (provisioner) | String | Mandatory | public key used to decrypt tokens |
| authorizationServerTokenURL (provisioner) | String | Mandatory | authorization server url for generating tokens |
| backendsResourceID (provisioner) | String | Mandatory | ID of backends resource – the client name as defined in the authorization server of client which reflects backends. |
| backendsClientID (provisioner) | String | Mandatory | ID of a backend client, which has a service account role ADMIN at backends resource |
| backendsClientSecret (provisioner) | String | Mandatory | The backends client secret |

| Parameter Name | Type | Is Mandatory | Description |
|---------------------|-----------|--------------|--|
| tags (provisioner) | List<Tag> | Mandatory | KS location Rack, Zone and Group ID's, Host ID, storageClassName, shareSSDBetweenVolumes. |
| name (tag) | String | Mandatory | |
| value (tag) | String | Mandatory | |
| version | String | Mandatory | |
| numberOfSSDs | int | Mandatory | |
| mgmtPort | int | Mandatory | The backend management port number. |
| Certified | Boolean | Mandatory | True if the appliance is certified. The appliance is considered certified if all its SSDs are certified and the hyper threading is disabled in it. |
| targetDriverVersion | String | Mandatory | The target driver version on the probing backend |
| changes | int | Mandatory | A bitmask, which tells provisioner if it needs to refresh more data. Current bits: PORTALS_CHANGED=1 SYSLOG_CHANGED=2 |
| authenticationMode | Enum | Mandatory | Current backend configured authentication mode (LOCAL , LDAP, OpenIDC) |

Response

Example 2: *Probe*: JSON response

```
{
  "description": "Success",
  "status": "Success"
  "provisioner": {
    , "authorizationServer": {
      "resourceID": "provisioner-gili",
      "clientID": "provisioner-gili",
      "tokenURL": "http://###.##.##.###:31295/auth/realms/Dev/protocol/openid-connect/token", "name": "",
      "clientSecret": "#####-####-####-####-#####",
      "publicKey": "BIGLONGSTRING" }
    , "inUse": false,
    "name": "unitTestProv",
    "pushInterval": 60,
    "url": "https://###.##.##.###30100",
    "tags": [ { "name": "rack", "value": "Rack1" },
              { "name": "zone", "value": "Zone1" },
              { "name": "region", "value": "Region1" } ]
  }
}
```

| Parameter Name | Type | Description |
|---------------------|-------------|--|
| description | String | Result description. |
| status | String | Result code. <i>Success</i> – Success |
| Provisioner | Provisioner | The Provisioner state including tags (similar to what is passed by the engine with SET_PROVISIONER API) |
| AuthorizationServer | Object | This must be provided when using OpenIDC. Otherwise, it may be null. Note that all the fields below are required if OpenIDC is used. |

| | | |
|---|---------------------------------|---|
| (Provisioner) | | |
| publicKey (AuthorizationS erver) | String | Used for decrypting a token. Required for OpenIDC. |
| tokenURL (AuthorizationS erver) | String | The url of the authentication server for generating a token. Required for OpenIDC. |
| clientID (AuthorizationS erver) | String | The storage node client id, used for generating a token. Required for OpenIDC. |
| resourceID (AuthorizationS erver) | String | The storage node resource ID used for extracting roles when opening a token. Required for OpenIDC. |
| clientSecret (AuthorizationS erver) | String | The storage node client secret, used for generating a token. Required for OpenIDC. |
| inUse (Provisioner) | Boolean | This specifies whether the engine is currently used by the Provisioner. |
| Name (Provisioner) | String . | The name of the Provisioner. This is an alpha numeric string not more than 16 characters and may include '-' or '_'. |
| pushInterval (Provisioner) | Int | The number of seconds between probe calls - must be in the range[1 -1440] minutes |
| url (Provisioner) | String | URL of the Provisioner in the format: <protocol>://<ip/host>:<port>. |
| Tags (Provisioner) | List of pairs {name, value } | Mandatory however may be an empty list. A maximum of 100 tags are permitted. <ul style="list-style-type: none">· name may be an alpha numeric string, not more than 16 characters.· value may be any string, not more than 255 chars |

Reset Inventory

Reset the inventory, an internal task is created to track the operation progress, it has an expiration period of halt an hour, which means when the task is finished it will be removed after the expiration period has passed.

| Method | URI |
|----------------------------|---|
| DELETE | /reset_inventory |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE --header "Content-Type: application/json" https://##.##.##.###:30100/reset_inventory |

Normal response codes: 200

Request

No request data

Response

Example 2: *Reset Inventory*: JSON response

```
{
  "description": "Success.",
  "status": "Success",
  "id": 1727
}
```

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| id | int | Command ID (internal usage). |
| status | String | Result code: <ul style="list-style-type: none">• <i>Success</i> – Success.• <i>CommunicationError</i> – KumoScale could not be reached.• KS Reset Inventory API result codes. |

Get Inventory

Get the inventory, an internal task is created to track the operation progress, has an expiration period of half an hour. This means when the task is finished it will be removed after the expiration period has passed.

| Method | URI |
|----------------------------|---|
| GET | /inventory |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/inventory |

Normal response codes: 200

Request

No URI parameters.

Response

Example 2: *Get Inventory*.JSON response

```
{
  "inventorySamples":[
    {
      "timestamp":1576066112110,
      "backends":[
        {
          "hardwareId":"00:0c:29:b9:6b:1d",
          "version":"3.7",
          "state":"AVAILABLE",
          "numSSDs":2
        },
        {
          "hardwareId":"0c:c4:7a:88:09:ee",
          "version":"3.7",
          "state":"AVAILABLE",
          "numSSDs":7
        }
      ]
    },
    {
      "timestamp":1576066112110,
      "backends":[
        {
          "hardwareId":"00:0c:29:b9:6b:1d",
          "version":"3.7",
          "state":"AVAILABLE",
          "numSSDs":2
        },
        {
          "hardwareId":"0c:c4:7a:88:09:ee",
          "version":"3.7",
          "state":"AVAILABLE",
          "numSSDs":7
        }
      ]
    }
  ]
}
```

| Parameter Name | Type | Description |
|--------------------------|------------------------------|---|
| timeStamp | Date | The inventory sampling date |
| changes | List<InventoryChange> | The list of changes in a specific inventory sampling date |
| Change (InventoryChange) | Enum | {NONE,NEW,REMOVED,MODIFIED} – The change type |
| ags (InventoryChange) | List of pairs {name , value} | name can be alpha numeric string value can be any string with the following supported keys: <ul style="list-style-type: none">id – string ,max length 256, not emptyversion - string ,max length 256,not emptystate – string , max length 16,not emptynumSSDs – int >=0 current number of SSDs |

Multi-Tenancy Management

Create Tenant

Create a tenant with capacity and performance budgets for a multi-tenancy environment.

| Method | URI |
|--------|----------|
| POST | /tenants |

| | |
|----------------------------|---|
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"tenantId": "#####-####-####-####-#####", "name": "TEST_TENANT", "capacity": 1000, "totalIOPS": 1000000, "totalBW": 1000000}' https://##.##.##.###:30100/tenants/ |
|----------------------------|---|

Normal response codes: 200

Request

No URI parameters.

Example 1: *Create Tenant*: JSON request

```
{
  "name": "TEST_TENANT",
  "capacity": 1000,
  "totalIOPS": 1000000,
  "totalBW": 1000000
}
```

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|-------------------------------------|
| name | String | Mandatory | The tenant name. |
| capacity | long | Mandatory | The tenant capacity budget in GB. |
| totalIOPS | long | Mandatory | The tenant IOPS budget. |
| totalBW | long | Mandatory | The tenant bandwidth budget in B/s. |
| tenantId | String | Optional | The tenant ID. |

Response

Example 2: *Create Tenant*: JSON response

```
{
  "description": "Success",
  "status": "Success",
  "tenantId": "#####-####-####-####-#####"
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>InvalidInput</i> – An input parameter is invalid.<i>TenantAlreadyExists</i> – A tenant with current input name already exists.<i>NoBackends</i> – No backends (storage nodes) are configured.KS Set Provisioner API result codes. |
| tenantId | String | The tenant ID |

Get Tenants

Lists all the configured tenants, their configuration and information.

| Method | URI |
|----------------------------|--|
| GET | /tenants |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://##.##.##.###:30100/tenants |

Normal response codes: 200

Request

No URI parameters.

Response

Example 2: Get Tenants: JSON response

```
[
  {
    "tenantId": "0",
    "name": "default",
    "capacity": 1024,
    "totalIOPS": 0,
    "totalBW": 4,
    "consumedCapacity": 0,
    "consumedIOPS": 0,
    "consumedBW": 0
  },
  {
    "tenantId": "#####-###-###-###-#####",
    "name": "TEST_TENANT",
    "capacity": 1000,
    "totalIOPS": 1000000,
    "totalBW": 1000000,
    "consumedCapacity": 0,
    "consumedIOPS": 0,
    "consumedBW": 0
  }
]
```

| Parameter Name | Type | Description |
|------------------|--------|--|
| tenantId | String | The tenant ID. |
| name | String | The tenant name. |
| capacity | number | The tenant capacity budget in GB. |
| totalIOPS | number | The tenant IOPS budget. |
| totalBW | number | The tenant bandwidth budget in B/s. |
| consumedCapacity | number | The tenant used capacity in GB. |
| consumedIOPS | number | The amount of consumed IOPS. |
| consumedBW | number | The amount of consumed bandwidth in B/s. |

Modify Tenant

Updates the tenant’s budget.

| Method | URI |
|----------------------------|--|
| PUT | /tenants/{tenantId} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X PUT -H 'Content-Type: application/json' -d '{"tenantId": "#####-###-###-###-#####", "capacity": 1000, "totalIOPS": 1000000, "totalBW": 1000000}' https://##.##.##.###:30100/tenants/{tenantId} |

Normal response codes: 200

Request

No URI parameters

Example 1: *Modify Tenant*: JSON request

```
{
  "tenantId": "#####-###-###-###-#####",
  "capacity": 2000,
  "totalIOPS": 2000000,
  "totalBW": 2000000
}
```

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|---|
| tenantId | String | Mandatory | The tenant ID. |
| capacity | number | Mandatory | The updated tenant's capacity budget in GB. |
| totalIOPS | number | Mandatory | The updated tenant's IOPS budget. |
| totalBW | number | Mandatory | The updated tenant's bandwidth budget in B/s. |

Response

Example 2: *Modify Tenant*. JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>InvalidInput</i> – An input parameter is invalid.<i>TenantNotFound</i> – The Tenant ID does not exist.<i>NoBackends</i> – No backends are configured.KS Set Provisioner API result codes. |

Delete Tenant

Deletes a configured tenant.

| Method | URI |
|----------------------------|--|
| DELETE | /tenants/{tenantId} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE -H 'Content-Type: application/json' https://###.##.##.###:30100/tenants/#####-###-###-###-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|----------------|
| tenantId | String | Mandatory | The tenant ID. |

Response

Example 2: *Delete Tenant*. JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| } | | |
|----------------|--------|--|
| Parameter Name | Type | Description |
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>InvalidInput</i> – An input parameter is invalid.<i>TenantNotFound</i> – The Tenant ID does not exist.<i>NoBackends</i> – No backends are configured.<i>TenantHasVolumes</i> – There are volumes configured over the tenant.KS Set Provisioner API result codes. |

Volume Management

Allocates volumes for the user.

Create Volume

Creates a volume in the pool.

| Method | URI |
|----------------------------|---|
| POST | / {tenantId}/volumes |
| Linux Curl Command Example | <pre>curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"capacity": 20, "provisioningType": "THIN", "protocol": "NVMeoF", "reservedSpacePercentage": 10, "storageClass": { "sameRackAllowed": true, "numReplicas": 1, "racks": ["Dev"], "regions": ["ISR"], "zones": ["Lab"], "maxIOPSPerGB": 1, "desiredIOPSPerGB": 1, "maxBWPerGB": 1, "desiredBWPerGB": 1, "blockSize": 1, "hostId": "#####-####-####-####-#####", "allowSpan": true "name": "sc1", "shareSSDBetweenVolumes": true }, "uuid": "#####-####-####-####-#####", "alias": "Vol1"}' https://##.##.##.###:30100/{tenantId}/volumes</pre> |

Normal response codes: 200

Request

No URI parameters.

Example 1: *Create Volume*: JSON request

```
{
  "capacity": 20,
  "provisioningType": "THIN",
  "protocol": "NVMeoF",
  "reservedSpacePercentage": 10,
  "storageClass": {
    "sameRackAllowd": true,
    "numReplicas": 1,
    "racks": [ "Dev" ],
```

```
    "regions": [ "ISR" ],

    "zones": [ "Lab" ] ,

    "maxIOPSPerGB": 1,

    "desiredIOPSPerGB": 1,

    "maxBWPerGB": 1,

    "desiredBWPerGB": 1,

    "blockSize": 512,

    "hostId": "#####-###-###-###-#####",

    "allowSpan": true,

    "name": "sc1",

    "shareSSDBetweenVolumes": true

  },

  "uuid": "#####-###-###-###-#####",

  "alias": "Vol1"

}
```

| Parameter Name | Type | Is Mandatory | Description |
|---------------------------------------|-------------------|--------------|--|
| capacity | long | Mandatory | Requested capacity in GiB granularity. |
| alias | String | Mandatory | |
| uuid | String | Optional | |
| provisioningType | Enum (THICK,THIN) | Optional | The volume provisioning type: thin-provisioned or thick provisioned. The default is thick. |
| reservedSpacePercentage | int | Optional | For thin volumes: The initial allocated capacity in percentage. Default is max (10% of volume capacity, 20GB). Minimal value is max (2% of volume capacity, 20GB). Maximum value is the volume capacity. |
| Protocol | Enum | Optional | {NVMeoF,Local} default is NVMe-oF |
| storageClass | StorageClass | Optional | Description from Storage Class CR. |
| sameRackAllowed (storageClass) | Boolean | Optional | For replicated volumes: Whether or not more than one replica can be allocated on the same rack. Default – false. |
| numReplicas (storageClass) | int | Optional | Default – 1 (a non-resilient volume). |
| racks (storageClass) | List<String> | Optional | A list of racks which the volume(s) may reside on. |
| zones (storageClass) | List<String> | Optional | A list of zones the volume(s) should be accessible from. |
| regions (storageClass) | List<String> | Optional | A list of regions the volumes should be accessible from. |
| maxIOPSPerGB (storageClass) | int | Optional | Upper limit for IOPS/GB. |
| desiredIOPSPerGB (storageClass) | int | Optional | Desired IOPS/GB. |
| maxBWPerGB (storageClass) | int | Optional | Upper limit for bandwidth in B/s per GB. |
| desiredBWPerGB (storageClass) | int | Optional | Desired bandwidth in B/s per GB. |
| blockSize (storageClass) | enum | Optional | Volume block size in bytes – 512 or 4096 (Default). |
| name (storageClass) | String | Optional | The name of the storage class. Mandatory if shareSSDBetweenVolumes is false. |
| shareSSDBetweenVolumes (storageClass) | Boolean | Optional | Whether or not the volume is allowed to be placed on an SSD that already has a volume with the same storageClassName. |
| hostId (storageClass) | String | Optional | The host id. |
| allowSpan (storageClass) | Boolean | Optional | Allow span. |
| tenantId | String | Optional | The tenant ID of the tenant the volume is created for. |

| Parameter Name | Type | Is Mandatory | Description |
|----------------|------|--------------|-------------|
| | | | |

Response

Example 2: *Create Volume*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>AlreadyExists</i> - A volume with requested UUID already exists.<i>InvalidInput</i> - No alias was provided.<i>AlreadyExists</i> - A volume with the requested alias already exists.<i>InvalidReplicas</i> - Invalid number of replicas.<i>NoMatchingBackend</i> - Could not find a compatible backend to allocate at least one of the replicas on.<i>NotEnoughMatchingBackends</i> - Could not create all replicas.<i>NoBackendWithHostIdExists</i> – No backend with the requested host Id exists.<i>TenantNotFound</i> – The Tenant ID does not exist.<i>TenantCapacityExceeded</i> – The tenant has reached its capacity budget.KS Create Volume API result codes. |

Delete Volume

Delete a volume from the KumoScale implementation.

| Method | URI |
|----------------------------|---|
| DELETE | /tenantId/volumes/{id} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE --header "Content-Type: application/json" https://##.##.##.###:30100/tenantId/volumes/#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| id | String | Mandatory | Volume UUID. |
| tenantId | String | Optional | The Tenant ID which the volume belongs to. |

Response

Example 2: *Delete Volume*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|------|-------------|
| | | |

| | | |
|-------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none">• <i>Success</i> – Success.• <i>NotExists</i> - A Backend with the required id does not exist.• <i>TenantNotFound</i> – The tenant doesn't exist.• <i>VolumeIsInUse</i> - Volume is in use by a snapshot.• KS Delete Volume API result codes. |

Get Volumes

Retrieve the location of the KumoScale which the Volume(s) was configured on.

| Method | URI |
|---|--|
| GET | /tenantId/volumes/{id} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/volumes |
| Example with the use of limit and offset parameters | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/{tenantId}/volumes?limit=5&offset=0 |
| Example with the use of limit and marker parameters | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/{tenantId}/volumes?limit=5&marker=5836c3cc-1c7e-4f2e-9698-e39175a41bd7 |
| Example with the use of hostID parameter | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/{tenantId}/volumes?hostID=#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|------------------------------|---|
| id | String | Mandatory (for single query) | Volume UUID. |
| limit | int | optional | Requests a page size of items. Returns a number of items up to a limit value. |
| Offset | int | optional | Used in conjunction with Limit to return a selection of items. Offset determines the starting point of the list. The engine will retrieve Volumes starting from the offset. |
| marker | String | optional | The engine will select Volumes starting from this UUID and greater than this value. |
| tenantId | String | Optional | The volume tenantId |
| backendId | String | Optional | If given, the API will return only volumes which has at least one replica on this backend. |
| hostID | String | Optional | If given, the API will return only volumes which are connected to the host with this hostID. |

Response

Example 4: Get Volumes: JSON response

```
[
{
  "alias": "sanity-node-get-volume-stats-4EC0D022-CB5B17BB",
  "uuid": "#####-####-####-####-#####",
  "capacity": 10737418240,
  "provisioningType": "THICK",
```

```
"numReplicas": 2,

"maxIOPS": 0,

"desiredIOPS": 0,

"maxBW": 0,

"desiredBW": 0,

"blockSize": "4KB",

"maxReplicaDownTime": 0,

"location": [

  {

    "uuid": "#####-###-###-###-#####",

    "backend": {

      "persistentID": "00:0c:29:8c:71:5f"

    },

    "replicaState": "Unknown",

    "currentStateTime": 0,

    "inUse": true

  },

  {

    "uuid": "#####-###-###-###-#####",

    "backend": {

      "persistentID": "00:0c:29:bf:e1:a3"

    },

    "replicaState": "Unknown",

    "currentStateTime": 0,

    "inUse": false

  }

],

"tenantId": "0",

"protocol": "NVMeoF",

"allowSpan": false,

"storageClassName": "default",

"shareSSDBetweenVolumes": true,

"inUse": false

},

{

  "alias": "#####-###-###-###-#####",

  "uuid": "#####-###-###-###-#####",

  "capacity": 42949672960,

  "provisioningType": "THIN",

  "numReplicas": 1,

  "maxIOPS": 0,

  "desiredIOPS": 0,

  "maxBW": 0,

  "desiredBW": 0,

  "blockSize": "4KB",

  "maxReplicaDownTime": 0,

  "location": [

    {

      "uuid": "#####-###-###-###-#####",

      "backend": {

        "persistentID": "00:0c:29:bf:e1:a3"

      },

      "replicaState": "Unknown",

      "currentStateTime": 0,

      "inUse": false

    }

  ]

}
```

```
    ],
    "tenantId": "0",
    "protocol": "Local",
    "allowSpan": true,
    "storageClassName": "default",
    "shareSSDBetweenVolumes": true,
    "inUse": false,
    "reservedSpace": 21474836480
  }
]
```

| Parameter Name | Type | Description |
|-----------------------------|--------------------------|---|
| uuid | String | Volume UUID. |
| alias | String | Volume alias. |
| capacity | long | Volume capacity in bytes. |
| provisioningType | Enum (THICK,THIN, OTHER) | The type of the volume: THIN/THICK/OTHER. OTHER is used for a snapshot volume |
| reservedSpace | int | For thin volumes: The actual initially allocated capacity in bytes. |
| numReplicas | int | Number of volume replicas. |
| maxIOPS | long | Upper limit for IOPS. |
| desiredIOPS | long | Desired IOPS. |
| maxBW | long | Upper limit for BW. |
| desiredBW | long | Desired BW. |
| blockSize | enum | Volume block size – 512B or 4KB. |
| snapshotID | String | For snapshot volumes only – the source snapshot UUID. |
| writable | Boolean | For snapshot volumes only – true if the volume is writable. |
| reservedSpace | long | For snapshot volumes only – reserved space in bytes |
| inUse | Boolean | Specifies if the volume is in use |
| location | List<VolumeLocation> | |
| uuid (location) | String | Volume replica UUID. |
| backend (location) | Backend | The persistent ID of the backend the replica resides in. |
| replicaState (location) | ReplicaState | Volume replica state – Unknown, Available, Terminating, Missing, or Synchronizing. |
| currentStateTime (location) | long | For replicated volumes - The duration of the current state of the replica, in ms. |
| inUse(location) | Boolean | Specifies if the location(replica) is in use |
| tenantId | String | The Tenant ID which the volume belongs to. |
| protocol | Enum | The volume’s protocol: {NVMeoF,Local}. |
| allowSpan | Boolean | Whether or not span allowed. |
| storageClassName | String | The name of the storage class. (not reported in case not configured) |
| shareSSDBetweenVolumes | Boolean | Whether or not the volume can be placed on an SSD that already has a volume with the same <i>storageClassName</i> . |

Get Volume by Alias

Retrieves the location of the KS which the Volume(s) was configured on by the volume alias.

| Method | URI |
|----------------------------|--|
| GET | /{{tenantId}}/volumes_by_alias/{alias} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/{{tenantId}}/volumes_by_alias/vol1 |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| alias | String | Mandatory | Volume alias. |
| tenantId | String | Optional | The Tenant ID which the volume belongs to. |

Response

See [Get Volumes](#).

Get Volume Extended

Retrieves extended volume information on a specific volume.

| Method | URI |
|----------------------------|--|
| GET | / {tenantId} /volume_extended/{uuid} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://#####:30100/{tenantId}/ volume_extended/{uuid} |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|---------------------|
| uuid | String | Mandatory | Volume uuid. |
| tenantId | String | Optional | The volume tenantId |

Response

Example: Get Volumes Extended: JSON response

```
[
  {
    "alias": "TestVol1",
    "uuid": "#####-###-###-###-#####",
    "capacity": 10737418240,
    "provisioningType": "THICK",
    "numReplicas": 1,
    "maxIOPS": 0,
    "desiredIOPS": 0,
    "maxBW": 0,
    "desiredBW": 0,
    "blockSize": "4KB",
    "maxReplicaDownTime": 0,
    "location": [
      {
        "uuid": "#####-###-###-###-#####",
        "backend": {
          "persistentID": "00:0c:29:79:04:8c"
        },
        "replicaState": "Unknown",
        "currentStateTime": 0,
        "subVolumes": [
          {
            "capacity": 10737418240,
            "ssdID": "VMWare NVME-0000",
            "ssdName": "SSD1"
          }
        ],
        "sessions": [
          {
            "hostID": "#####-###-###-###-#####"
          }
        ]
      }
    ],
    "tenantId": "0",
    "protocol": "NVMeoF",
    "allowSpan": true,
    "shareSSDBetweenVolumes": true
  }
]
```

```
  }
]
```

All parameters from [Get volumes](#) and additionally, for each replica:

| Parameter Name | Type | Description |
|----------------|--------|---|
| Sub-Volumes | List | List of sub-volume, each part includes: capacity, ssdId and name. |
| Capacity | Int | The capacity of the sub-volume (in bytes) |
| ssdID | String | Persistent ID of the SSD |
| ssdName | String | Name of the SSD |
| Session | List | |
| host ID | String | Host UUID (reported only in case there is initiator that connected to the volume) |

Expand Volume

Expand exists volume on all its KS replicas.

| Method | URI |
|----------------------------|---|
| PATCH | /tenantId}/volumes/{id} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X PATCH --header "Content-Type: application/json" -d '{"newCapacity": 0}' https://##.##.##.###:30100/tenantId}/volumes/#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| id | String | Mandatory | Volume UUID. |
| newCapacity | int | Mandatory | New Volume capacity in GB. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

Example 1: *Create Volume*: JSON request

```
{
  "newCapacity": 20
}
```

Response

Example 2: *Expand Volume*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> - Success<i>NotExists</i> - A Backend with the required id does not exist.<i>InvalidSize</i> – Current capacity is greater than or equal to newCapacity.<i>NoFreeSpace</i> – There is not enough free space on at least one of the volume's backend. |

| | | |
|--|--|--|
| | | <ul style="list-style-type: none"><i>TenantNotFound</i> – The Tenant ID does not exist.<i>TenantCapacityExceeded</i> – The tenant has reached its capacity budget.KS Modify Volume API result codes. |
|--|--|--|

Add Replica

Add another replica to a replicated volume.

| Method | URI |
|----------------------------|--|
| POST | /tenantId/replica/{id} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"sameRackAllowed": true, "racks": ["Dev"], "regions": ["ISR"], "zones": ["Lab"] }' https://###.###.###.###:30100/tenantId/replica/#####-####-####-####-##### |

Normal response codes: 200

Request

No URI parameters.

Example 1: Add Replica: JSON request

| |
|--|
| <pre>{ "sameRackAllowed": true, "racks": ["Dev"], "regions": ["ISR"], "zones": ["Lab"] }</pre> |
|--|

| Parameter Name | Type | Is Mandatory | Description |
|-----------------|--------------|--------------|---|
| uuid | String | Mandatory | The volume’s UUID. |
| sameRackAllowed | Boolean | Optional | Can the replicas be allocated on the same rack. Default – false. |
| racks | List<String> | Optional | A list of racks the volume(s) should reside on. |
| zones | List<String> | Optional | A list of zones the volume(s) should be accessible from. |
| regions | List<String> | Optional | A list of regions the volumes should be accessible from. |
| tenantId | String | Optional | The Tenant ID which the volume belongs to. |

Response

Example 2: Add Replica: JSON response

| |
|--|
| <pre>{ "description": "Success", "status": "Success" }</pre> |
|--|

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> - Success<i>NotExists</i> - A Backend with the required id does not exist.<i>NotReplicated</i> – The volume is not replicated.<i>NoMatchingBackend</i> - Could not find a compatible backend to allocate the replica on.<i>InvalidReplicas</i> - Invalid number of replicas.<i>ReplicaTerminating</i> – The replica is terminating.<i>NoFreeSpace</i> - There is not enough free space.<i>TopologyConstraints</i> – Cannot add a replica due to topology constraints.<i>NotInUse</i> - Some of the backends are not in use. |

| Parameter Name | Type | Description |
|----------------|------|---|
| | | <ul style="list-style-type: none">• <i>SameRackNotAllowed</i> - Could not find matching backends on different racks for replication.• <i>NotAvailable</i> - The backend is not available.• <i>NotEnoughMatchingBackends</i> - There were not enough backends which matched the storage class criteria.• <i>TenantNotFound</i> – The Tenant ID does not exist.• <i>TenantCapacityExceeded</i> – The tenant has reached its capacity budget.• KS Create Volume API result codes. |

Delete Replica

Mark a replica to be removed from a Replicated volume.

| Method | URI |
|----------------------------|---|
| PATCH | /tenantId/replica/{id}/{replicaID} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X PATCH --header "Content-Type: application/json" https://###.###.###.###:30100/tenantId/replica/#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|---|
| id | String | Mandatory | Volume UUID. |
| replicaID | String | Mandatory | The volume ID of the replica (physical volume). |
| tenantId | String | Optional | The Tenant ID which the volume belongs to. |

Response

Example 2: *Delete Replica*: JSON response

| |
|--|
| <pre>{ "description": "Success", "status": "Success" }</pre> |
|--|

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | <p>Result code.</p> <ul style="list-style-type: none">• <i>Success</i> - Success• <i>NotExists</i> - A Backend with the required id does not exist.• <i>NotReplicated</i> – The volume is not replicated.• <i>AllReplicasDown</i> – All replicas are down.• <i>ReplicaTerminating</i> – One of the replicas is "Terminating".• <i>TenantNotFound</i> – The Tenant ID does not exist. |

Delete Replica Confirm

Remove a replica (delete volume) from a Replicated volume.

| Method | URI |
|--------|--|
| DELETE | /tenantId/replica/{volumeID}/{replicaID} |

| | |
|-----------------|--|
| Linux Curl | curl -k --cert ./ssdtoolbox.pem -i -X DELETE --header "Content-Type: application/json" |
| Command Example | https://###.###.###.###:30100/{tenantId}/replica/#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|---|
| volumeID | String | Mandatory | Volume UUID. |
| replicaID | String | Mandatory | The volume ID of the replica (physical volume). |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

Response

Example 2: *Delete Replica Confirm*: JSON response

| |
|--|
| <pre>{ "description": "Success", "status": "Success" }</pre> |
|--|

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> - Success<i>NotExists</i> - A Backend with the required id does not exist.<i>NoTerminating</i> - None of the replicas is "Terminating".<i>TenantNotFound</i> - The Tenant ID does not exist.KS Delete Volume API result codes. |

Set Replica State

Sets a volume replica state to either Available or Missing.

| Method | URI |
|-----------------|---|
| PATCH | / {tenantId}/replica/{id}/{replicaID}/{replicaState} |
| Linux Curl | curl -k --cert ./ssdtoolbox.pem -i -X PATCH --header "Content-Type: application/json" |
| Command Example | https://###.###.###.###:30100/{tenantId}/replica/#####-####-####-####-#####/Available |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|---|
| id | String | Mandatory | Volume UUID. |
| replicaID | String | Mandatory | The volume ID of the replica (physical volume). |
| replicaState | String | Mandatory | The replica state: Available, Missing or Synchronizing. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

Response

Example 1: *Set Replica State*: JSON response

| <pre>{ "description": "Success", "status": "Success" }</pre> | | |
|--|--------|---|
| Parameter Name | Type | Description |
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success<i>InvalidInput</i> - Invalid replica state<i>NotExists</i> – A replica with this ID not found.<i>NotReplicated</i> – The volume is not replicated.<i>ReplicaTerminating</i> – The replica is terminating.<i>TenantNotFound</i> – The Tenant ID does not exist. |

Create Snapshot

Create a snapshot over a volume.

| Method | URI |
|----------------------------|---|
| POST | /tenantId/snapshots |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"alias": "TestSnap3", "volumeID": "#####-####-####-####-#####", "snapshotID": "#####-####-####-####-#####", "reservedSpacePercentage": 10}' https://##.##.##.###:30100/tenantId/snapshots |

Normal response codes: 200

Request

No URI parameters.

Example 1: *Create Snapshot*: JSON request

| <pre>{ "alias": "TestSnap3", "volumeID": "#####-####-####-####-#####", "snapshotID": "#####-####-####-####-#####", "reservedSpacePercentage": 10 }</pre> | | | |
|--|--------|--------------|---|
| Parameter Name | Type | Is Mandatory | Description |
| alias | String | Mandatory | Snapshot alias (same rules as volume alias). |
| volumeID | String | Mandatory | Volume UUID of the parent volume for snapshot. |
| snapshotID | String | Optional | The snapshot UUID. |
| reservedSpacePercentage | int | Optional | Percentage of the parent volume used for a log. Default is max (10% of volume capacity, 20GB). Minimal value is max (2% of volume capacity, 20GB). Maximum value is the volume capacity. |

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| | | | |
| tenantId | String | Optional | The Tenant ID which the volume belongs to. |

Response

Example 2: Create Snapshot: JSON response

```
{
  "description": "Success",
  "status": "Success",
  "persistentId": "#####-####-####-####-#####"
```

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none">Success – Success.NotExists - Volume is not exists with id.Replicated - Volume is replicated.TenantNotFound - The Tenant ID does not exist.TenantCapacityExceeded - The tenant has reached its capacity budget.KS Create Snapshot API result codes. |
| persistentId | String | The snapshot UUID. |

Delete Snapshot

Delete a snapshot from the KumoScale implementation.

| Method | URI |
|----------------------------|---|
| DELETE | /tenantId/snapshots/{id} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE --header "Content-Type: application/json" https://##.##.##.###:30100/tenantId/snapshots/#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| id | String | Mandatory | Snapshot UUID. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

Response

Example 2: Delete Snapshot: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|------|-------------|
|----------------|------|-------------|

| | | |
|-------------|--------|--|
| | | |
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none">• <i>Success</i> – Success.• <i>NotExists</i> – A snapshot with this ID does not exist.• <i>TenantNotFound</i> - The Tenant ID does not exist.• <i>VolumeIsInUse</i> - Volume is in use by a snapshot• KS Delete Snapshot API result codes. |

Get Snapshots

Retrieves a list of all of the snapshots if the limit parameter is not specified.

Otherwise, retrieves a list with size <=limit snapshots (a page).

| Method | URI |
|---|--|
| GET | /tenantId/snapshots/{id} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/tenantId/snapshots |
| Example with the use of limit and offset parameters | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/tenantId/snapshots?limit=5&offset=0 |
| Example with the use of limit and marker parameters | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/tenantId/snapshots?limit=5&marker=#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|---|---|
| id | String | Mandatory (when retrieving a specific snapshot) | Snapshot UUID. |
| limit | int | optional | Requests a page size of items. Returns a number of items up to the limit. |
| Offset | int | optional | Used in conjunction with Limit to return a selection of items. Offset determines the starting point of the list. The engine will retrieve Volumes starting from the offset. |
| marker | String | optional | The engine will select Snapshots starting from this UUID and greater than this value. |
| tenantId | String | Optional | The Tenant ID to which the volume and snapshot belong. |

Response

Example 5: Get Snapshots: JSON response

```
[
{
  "alias": "TestSnap3",
  "volumeID": "#####-####-####-####-#####",
  "snapshotID": "#####-####-####-####-#####",
  "replicaID": "#####-####-####-####-#####",
  "reservedSpacePercentage": 10,
  "reservedSpace": 10737418240,
  "capacity": 10737418240,
  "timestamp": 1586166064958,
  "tenantId": "0"
}
```

| |
|---|
|] |
|---|

| Parameter Name | Type | Description |
|-------------------------|---------|---|
| volumeID | String | The volume UUID. |
| alias | String | The snapshot alias. |
| ReplicaID | String | The UUID of the volume's replica which the snapshot was taken from. |
| capacity | long | The snapshot capacity in bytes. |
| snapshotID | String | The snapshot UUID. |
| reservedSpace | long | The allocated space in bytes. |
| reservedSpacePercentage | integer | The allocated space by percentage. |
| timestamp | long | The timestamp from when the snapshot was taken. |
| tenantId | String | The ID of the tenant to which the snapshot belongs. |

Get Snapshots by Volume

Get Snapshots according to a Volume UUID.

| Method | URI |
|----------------------------|--|
| GET | / {tenantId}/snapshots_by_vol/{volID} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/{tenantId}/snapshots_by_vol/#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| volID | String | Mandatory | The volume UUID. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

Response

See [Get Snapshots](#).

Get Snapshots by Alias

Get a snapshot by its alias

| Method | URI |
|----------------------------|---|
| GET | / {tenantId}/snapshots_by_alias/{alias} |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -X GET https://###.###.###.###:30100/{tenantId}/snapshots_by_alias/sns1 |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| alias | String | Mandatory | The snapshot alias. |
| tenantId | String | Optional | The Tenant ID to which the snapshot belongs. |

Response

See [Get Snapshots](#).

Create Snapshot Volume

Create a snapshot volume in the pool.

| Method | URI |
|----------------------------|--|
| POST | /tenantId/snapshot_volume |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"alias":"TestSnap3", "volumeID":"#####-####-####-####-#####", "snapshotID":"#####-####-####-####-#####", "reservedSpacePercentage":10, "writable":true, "maxIOPSPerGB":100, "maxBWPerGB":2000, "protocol":"Local", "allowSpan":true, "storageClassName":"sc1"}' https://192.0.2.0:30100/tenantId/snapshot_volumes |

Normal response codes: 200

Request

No URI parameters.

Example 1: *Create Snapshot Volume*: JSON request

```
{  
  "alias": "TestSnap3",  
  "volumeID": "#####-####-####-####-#####",  
  "snapshotID": "#####-####-####-####-#####",  
  "reservedSpacePercentage": 10,  
  "writable": true,  
  "maxIOPSPerGB": 100,  
  "maxBWPerGB": 2000,  
  "protocol": "Local",  
  "allowSpan": true,  
  "storageClassName": "sc1"  
}
```

| Parameter Name | Type | Is Mandatory | Description |
|-------------------------|---------|--------------|---|
| alias | String | Mandatory | The volume alias. |
| volumeID | String | Optional | The volume UUID. |
| snapshotID | String | Mandatory | The snapshot UUID. |
| reservedSpacePercentage | int | Optional | <div>For writable snapshot volumes only:</div> <div>The percentage of the parent volume reserved for the log.</div> <div>Default is max (10% of volume capacity, 20GB).</div> <div>Reserved Space minimal value is max (2% of volume capacity, 20GB), as in Thin Volume.</div> <div>Maximum value is the volume capacity.</div> |
| writable | Boolean | Optional | <div>Indicates if the snapshot volume is ReadOnly or Write Only.</div> <div>(The default is false).</div> |

| Parameter Name | Type | Is Mandatory | Description |
|------------------|---------|--------------|---|
| | | | |
| maxIOPSPerGB | int | Optional | The upper limit for IOPS/GB. |
| maxBWPerGB | int | Optional | The upper limit for bandwidth in B/s per GB. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |
| Protocol | Enum | Optional | The volume's required protocol: {NVMeoF,Local} default is NVMeoF. |
| allowSpan | Boolean | Optional | Allow span (Default – true). |
| storageClassName | String | Optional | The name of the storage class. |

Response

Example 2: *Create Snapshot Volume*: JSON response

```
{
  "description": "Success",
  "status": "Success",
  "persistentId": "#####-####-####-####-#####",
}
```

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>NotExists</i> – A snapshot with this id does not exist.<i>AlreadyExists</i> - A volume with requested UUID already exists or volumes already assigned for this snapshot.<i>TenantNotFound</i> - The Tenant ID does not exist.<i>TenantCapacityExceeded</i> - The tenant has reached its capacity budget.KS Create Snapshot Volume API result codes. |
| persistentId | String | The Snapshot Volume UUID. |

Migrate Volume

Migrates a replica of a volume from one backend to another according to the volume's storage class constraints. This command may be used for online replicated volumes only.

| Method | URI |
|----------------------------|--|
| POST | / {tenantId}/migrate_volume |
| Linux Curl Command Example | <div>curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"repId":"xx></div> |

Normal response codes: 200

Request

No URI parameters.

Example 1: *Migrate Volume*: JSON request

```
{
  "repId": " #####-####-####-####-#####",
}
```

```
"volId":" #####-###-###-###-#####"
}
```

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| repId | String | Mandatory | The ID of the replica you want to migrate. |
| volId | String | Mandatory | The volume UUID the replica belongs to. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

Response

Example 2: *Migrate Volume*: JSON response

```
{
  "description":"Success",
  "status":"Success",
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success<i>ReplicaNotFound</i> - a replica with the repd provided doesn’t exist.<i>VolumeNotFound</i> – the volume is not found.<i>TaskNotFound</i> – the task not found.<i>NotReplicated</i> – the volume is not replicated.<i>NotEnoughReplicas</i> – The volumes do not have enough available replicas (the number it was created with)<i>VolumeNotPublished</i> – The volume is not published (is not online)<i>ReplicaTerminating</i> – The replica is in ‘Terminating’ state<i>HostNotFound</i> – The host was not found.<i>TaskAlreadyRunning</i> – There is an active remote migration task running on this volume.<i>NoBackendToMigrateReplica</i> - No matching backend to migrate the replica to.<i>TenantNotFound</i> - The Tenant ID does not exist.<i>VolumePublishedToMultipleHosts</i> – The volume is published to more than one host. |

Clone Volume

Clone a Volume in the pool.

| Method | URI |
|----------------------------|--|
| POST | /tenantId/clone_volume |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"sourceVolumeId":" #####-###-###-###-#####", "alias":"cloneVolAlias", "capacity":20, "reservedSpacePercentage":10, "volumeId":" #####-###-###-###-#####"}' https://192.0.2.0:30100/tenantId/clone_volume |

Normal response codes: 200

Request

No URI parameters.

Example 1: *Clone Volume*: JSON request

```
{
  "sourceVolumeId": "#####-####-####-####-#####",
  "alias": "cloneVolAlias",
  "volumeId": "#####-####-####-####-#####",
  "reservedSpacePercentage": 10,
  "capacity": 20
}
```

| Parameter Name | Type | Is Mandatory | Description |
|-------------------------|--------|--------------|--|
| sourceVolumeId | String | Mandatory | The source volume UUID. |
| volumeId | String | Optional | The clone UUID. |
| alias | String | Mandatory | The clone alias. |
| reservedSpacePercentage | int | Optional | Percentage of the parent volume for used for log Default is max(10% of volume capacity, 20GB). Reserved Space minimal value is max(2% of volume capacity, 20GB) as in thin volume * In any case – reserved space will not be more than volume capacity. |
| capacity | int | Optional | The capacity of the target volume, possible values are the source volume capacity or bigger. * The parameter units are GB |
| tenantId | String | Optional | The clone volume tenantId |

Response

Example 2: *Clone Volume*: JSON response

```
{
  "description": "Success",
  "status": "Success",
  "persistentId": "#####-####-####-####-#####",
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> - Success<i>NotExists</i> – Source volume doesn’t exists with id<i>AlreadyExists</i> - A volume with requested UUID already exists or volumes already assigned for this snapshot.<i>TenantNotFound</i> – The tenant doesn’t exist.<i>TenantCapacityExceeded</i> – tenant doesn't have enough capacity for cloned volume.KS Clone Volume API result codes.KS Modify Volume API result codes. |
| persistentId | String | The new Cloned Volume UUID. |

Logging to an External Syslog Server

Forward Log

Forward a command or event to the syslogs.

| Method | URI |
|----------------------------|---|
| POST | /forward_log |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST --header "Content-Type: application/json" -d -d '{ "loggingType": "COMMAND", |

| | |
|--|---|
| | <pre>"level":"ERROR", "host":"TestHost", "appName":"TestApp", "message":"TestMessage-forwardLogTest", "parametersList":{ "Test":"TestValue" } }' https://###.###.###.###:30100/forward_log</pre> |
|--|---|

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|----------------------|--------------|--|
| loggingType | enum | Mandatory | An enum with the following possible values {EVENT, COMMAND}. |
| level | enum | Mandatory | The level to record the event in the syslog, values: {FATAL, ERROR, WARN, INFO}. |
| host | String | Mandatory | The host name. |
| appName | String | Mandatory | The name of the application which generated the event. |
| message | String | Mandatory | The message to write to the syslog. |
| parametersList | List<String, String> | Optional | The parameters list of the event/command. |

Response

Example 2: *Forward Log*: JSON response

| |
|---|
| <pre>{ "description":"Success", "status":"Success" }</pre> |
|---|

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>SyslogNotAvailable</i> - The provisioner does not have a working syslog configuration. |

Get Syslogs

Get the provisioner syslogs.

| Method | URI |
|----------------------------|--|
| GET | /syslog |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/syslog |

Normal response codes: 200

Request

No URI parameters.

Response

Example 2: *Get Syslogs*: JSON response

| |
|---|
| <pre>[{ "id":3889, "name":"TestSyslog", "url":"tcp://192.0.2.1:6514", "state":true, "useTls":true, "certFileName":"syslog_cert.crt" }]</pre> |
|---|

| Parameter Name | Type | Description |
|----------------|------|---|
| syslogs | List | <div>The syslog parameters returned:</div> <ul style="list-style-type: none">id - The syslog entity ID.name - The syslog server name.url - The syslog server URL.state - The syslog connection state.useTls - If true - use TLS for the communicating with the syslog server. certFileName - The filename of the file containing the syslog certificate in KS (uploaded using Certificate Upload API) |

License Management

Set License

Set the License. An internal task is created to track the operation progress. It has an expiration period of half an hour, which means when the task is finished it will be removed after the expiration period has passed.

| Method | URI |
|----------------------------|--|
| POST | /LICENSE |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST --header "Content-Type: application/json"-H "Authorization: Basic #####" -d '{"license":"XXXXXXXXXXXXXXXXXXXX"}' https://##.##.##.###:30100/license |

Normal response codes: 200

Request

| Parameter Name | Type | Description |
|----------------|--------|-----------------|
| license | String | The license key |

Response

Example 144: *Set License*: JSON response

| |
|---|
| <div>{ "description":"Success.", "status":"Success", }</div> |
|---|

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | <div>Result code:</div> <ul style="list-style-type: none"><i>Success</i> –successful.<i>LicenseError</i> – If not enough backends succeeded to set the license.<i>InvalidLicense</i> – if the license could not be decrypted.KS Set License API result codes. |

Get License

Get the license information.

| Method | URI |
|----------------------------|--|
| GET | /license |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://##.##.##.###:30100/license |

Normal response codes: 200

Request

No URI parameters.

Response

Example 2: *Get License*: JSON response

| |
|---|
| <div>{ "type":"POC", "expirationDate": "2020-02-26T00:00:00.350+0000", "maxBackends":4</div> |
|---|

```
}

```

| Parameter Name | Type | Description |
|----------------|------|--|
| type | enum | {POC, PRODUCTION}. |
| expirationDate | Date | The expiration date (or null - no expiration date). |
| maxBackends | int | The number of backends the license permits or null if unlimited. |

Host Management

List Hosts

Lists all the application hosts (initiators) configured in the Provisioner and their details.

| Method | URI |
|---|--|
| GET | /hosts |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://##.##.##.###:30100/hosts |
| Example with the use of limit and offset parameters | curl -k --cert ./ssdtoolbox.pem https://##.##.##.###:30100/hosts?limit=5&offset=0 |
| Example with the use of limit and marker parameters | curl -k --cert ./ssdtoolbox.pem https://##.##.##.###:30100/hosts?limit=5&marker=#####-####-####-####-#####&hostName="myHost" |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| limit | int | optional | Requests a page size of items. Returns a number of items up to a limit value. |
| offset | int | optional | Used in conjunction with Limit to return a selection of items. Offset determines the starting point of the list. The Provisioner will retrieve Hosts starting from the offset. |
| marker | String | optional | The Provisioner will select Hosts starting from this ID and greater than this value. |
| hostName | String | optional | The host name. |

Response

Example: List Hosts: JSON response

```
[
  {
    "uuid":"#####-####-####-####-#####",
    "name":"testHostName1",
    "nqn":"testNqn1",
    "clientType":"Agent",
    "state":"Available",
    "version":"1.4",
    "lastProbTime":1579181067908,
    "duration":180,
    "multipath": true ,
    "usedNamespacesIds": [
      1,
      2
    ]
  }
]
```

| Parameter Name | Type | Description |
|----------------|------|-------------|
|----------------|------|-------------|

| | | |
|-------------------|------------|---|
| nqn | String | The host NQN. |
| uuid | String | The host UUID. |
| clientType | Enumerator | {Kubernetes, Agent}. |
| version | String | SW version. |
| state | Enumerator | The host state {available, unavailable}. |
| lastProbTime | Timestamp | The last time the host was available (for an unavailable host). |
| duration | Integer | Probe duration length (seconds). |
| name | String | The host name. |
| multipath | Boolean | Tells if host activates multipath when connecting |
| usedNamespacesIds | Integer | Used namespaces IDs for this host |

Host Probe

A Host makes this call to notify the Provisioner about itself and update its properties.

| Method | URI |
|----------------------------|--|
| POST | /host_probe |
| Linux Curl Command Example | <div><pre>curl -k --cert ./ssdtoolbox.pem -i -X POST -H 'Content-Type: application/json' -d '{"hostId": "6d4a8383-c09f-4f91-836d-5ed192ca6f64", "hostNqn": "nqn.2014-08.org.nvmexpress:NVMf:uuid:#####-####-####-####-#####", "name": "testhostname", "clientType": "Agent", "version": "1.4", "duration": 30, "multipath": true}' https://##.##.##.###:30100/host_probe</pre></div> |

Normal response codes: 200

Request

Example 1: Host *Probe*: JSON request

```
{
  "hostId": "6d4a8383-c09f-4f91-836d-5ed192ca6f64",
  "hostNqn": "nqn.2014-08.org.nvmexpress:NVMf:uuid:#####-####-####-####-#####",
  "name": "testhostname",
  "clientType": "Agent",
  "version": "1.4",
  "duration": 30,
  "multipath": true,
}
```

| Parameter Name | Type | Is Mandatory | Description |
|----------------|---------|--------------|---|
| hostNqn | String | Mandatory | The host NQN. |
| hostId | String | Mandatory | The host UUID. |
| clientType | Enum | Mandatory | {Kubernetes, Agent}. |
| Name | String | Mandatory | The host name. |
| duration | int | Mandatory | Probe duration in seconds. |
| version | String | Mandatory | SW version. |
| multipath | Boolean | Optional | Indicates whether the host activates multipath when connecting. Default value is false. |

Response

Example 2: Host Probe: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success. |

Delete Host

Delete a host from the pool.

| Method | URI |
|----------------------------|--|
| DELETE | /hosts/{uuid} |
| Linux Curl Command Example | <div>curl -k --cert ./ssdtoolbox.pem -i -X DELETE --header "Content-Type: application/json" https://10.10.10.10:30100/hosts/{uuid}</div> |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|----------------|
| UUID | String | Mandatory | The host UUID. |

Response

Example 2: *Delete Host*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> - Success<i>NotExists</i> - A Host with the required uuid does not exist.<i>HostAvailable</i> – The host is available and cannot be deleted. |

Connectivity

Get Targets

Lists all of the targets configured in the Provisioner and their details, according to the connected host ID or a volume ID that was added to it as a namespace (only one of the parameters should be provided).

| Method | URI |
|----------------------------|--|
| GET | /targets |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://10.10.10.10:30100/targets?hostId= #####-####-####-####-##### |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://10.10.10.10:30100/targets?volId=#####-####-####-####-##### |

Normal response codes: 200

Request

Example 1: Get Targets: JSON request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|---|
| hostId | String | optional | The host UUID connected to the target. |
| volId | String | optional | The volume UUID of a volume which was added to the target as a namespace. |

Only one of the input parameters may be provided.

Response

Example 2: Get Targets: JSON response

```
[
{
  "name": "nqn.2022-08.com.kioxia:#####-####-####-####-#####",
  "alias": "linkedin",
  "connectedBackends": [
    {
      "persistentID": "00:0c:29:96:91:b7"
    },
    {
      "persistentID": "00:0c:29:3a:2b:d0"
    }
  ],
  "hostTargetConnectedTo": "linkedin",
  "numNamespaces": 2
}
]
```

| Parameter Name | Type | Description |
|-----------------------|---------|--|
| name | String | The target NQN. |
| alias | String | The target alias |
| hostTargetConnectedTo | String | The host name which is the target connected to |
| numNamespaces | Integer | Number of namespaces |
| connectedBackends | Array | List of connected backends persistent IDs |

Publish

Publish a volume to a specific host.

| Method | URI |
|----------------------------|--|
| POST | / {tenantId} /publish |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST --header "Content-Type: application/json" -d '{ "hostId": "String", "volId": "String" }' https://###.###.###.###:30100/{tenantId}/publish |

Normal response codes: 200

Request

Example 1: *Publish*: JSON request

```
{
  "hostId": "String",
  "volId": "String"
}
```

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|------|--------------|-------------|
|----------------|------|--------------|-------------|

| | | | |
|----------|--------|-----------|--|
| | | | |
| hostId | String | Mandatory | The host UUID. |
| volId | String | Mandatory | The volume UUID. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

Response

Example 2: Publish: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none">Success – Success.HostNotFound – The host does not exist.VolumeNotFound – The volume does not exist.PublishFailed – The operation failed.AlreadyPublished – All replicas already published to the host.TenantNotFound - The Tenant ID does not exist.VolumeNotInHost - The volume is not in the host.KS Create Target API, Create Namespace API, Connect Host API result codes. |

Unpublish

Unpublish a volume from a specific host and stop migration tasks for this volume.

| Method | URI |
|----------------------------|--|
| POST | /tenantId/unpublish |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X POST --header "Content-Type: application/json" -d '{"hostId":"#####-####-####-#####-####-##### ", "volId":"#####-####-####-#####-####-##### "}' https://##.##.##.###:30100/tenantId/unpublish |

Normal response codes: 200

Request

Example 1: Unpublish: JSON request

```
{
  "hostId": "#####-####-####-#####-####-##### ",
  "volId": "#####-####-####-#####-####-##### "
}
```

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|--|
| hostId | String | Mandatory | The host UUID. |
| volId | String | Mandatory | The volume UUID. |
| tenantId | String | Optional | The Tenant ID to which the volume belongs. |

| | | | |
|--|--|--|--|
| | | | |
|--|--|--|--|

Response

Example 2: *Unpublish*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|---|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>HostNotFound</i> – The host does not exist.<i>VolumeNotFound</i> – The volume does not exist.<i>UnpublishFailed</i> – The operation failed.<i>NotAvailable</i> – Unpublishing a target which is not found in KS.<i>TenantNotFound</i> - The Tenant ID does not exist.<i>VolumeNotInHost</i> - The volume is not in the host.KS Remove Target API, Remove Namespace API, Disconnect Host API result codes. |

Task Management

Get Tasks

Lists all of the tasks in the Provisioner and their details, according to the task ID or host ID it relates to, or all tasks. Only one of the parameters may be provided.

Notes: Parameters that are null are not returned in the JSON.

| Method | URI |
|----------------------------|---|
| GET | /tasks |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/tasks?taskId=#####-####-####-####-##### |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/tasks?hostId=#####-####-####-####-##### |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/tasks |

Normal response codes: 200

Request

Example 1: Get Tasks: JSON request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|----------------|
| taskId | String | optional | The task ID. |
| hostId | String | optional | The host UUID. |

Only one of the input parameters may be provided.

Response

Example 2: Get Tasks: JSON response

```
[
```

```
{
  "state": "FINISHED",

  "startDate": "2020-02-19T09:48:01.037",

  "endDate": "2020-02-19T09:49:05.011",

  "progress": 100,

  "type": "RemoteMigration",

  "statusDescription": "Success",

  "stoppable": false,

  "taskId": "#####-####-####-####-##### ",

  "taskConfiguration": [
    {
      "name": "volId",

      "value": "#####-####-####-####-##### "
    }
  ],

  "hostId": "#####-####-####-####-##### ",

  "taskStatus": "Success"
}
```

]

| Parameter Name | Type | Description |
|-------------------|---------|--|
| taskId | String | The task ID. |
| state | Enum | {IDLE,RUNNING,FINISHED,FAILED,ABORTED}. |
| taskConfiguration | tags | A list of <Tag> the task was configured with. |
| type | Enum | {Remote Migration, Install License, Update Inventory, Fetch Inventory, Reset Inventory}. |
| hostId | String | The host UUID of the host related to the task. |
| progress | number | Progress by percentage. |
| taskStatus | String | The error code string. |
| statusDescription | String | The error code description. |
| startDate | Date | The task starting time in yyyy-MM-dd'T'HH:mm:ss.SSS format. |
| endDate | Date | The task end time in yyyy-MM-dd'T'HH:mm:ss.SSS format. |
| stoppable | Boolean | Is the task stoppable. |

Update Task

Updates properties of a task according to its task ID or its host ID.

| Method | URI |
|----------------------------|--|
| PUT | /tasks |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X PUT -H 'Content-Type: application/json' -d '{ "state": "FINISHED", "progress": 100, "statusDescription": "Success", "taskId": "#####-####-####-####-##### ", "hostId": "#####-####-####-####-##### ", "taskStatus": "Success", }' https://###.###.###.###:30100/tasks |

Normal response codes: 200

Request

No URI parameters

Example 1: *Update Task*: JSON request

```
{
  "state": "FINISHED",
  "progress": 100,
  "statusDescription": "Success",
  "taskId": "#####-####-####-####-##### ",
  "hostId": "#####-####-####-####-##### ",
  "taskStatus": "Success",
}
```

| Parameter Name | Type | Is Mandatory | Description |
|-------------------|-----------|--------------|-----------------------------|
| taskId | String | Mandatory | The task ID. |
| state | String | Optional | The task state. |
| progress | Number | Optional | The progress by percentage. |
| taskStatus | String | Optional | The error code string. |
| statusDescription | String | Optional | The error code description. |
| type | Enum | Optional | The task type. |
| taskConfiguration | List<Tag> | Optional | The task configuration. |
| hostId | String | Mandatory | The host UUID. |

Response

Example 2: *Update Task*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>TaskNotFound</i> – The task was not found.<i>UpdateTaskFailed</i> – Failed to update the task.<i>TaskEnded</i> – The task already ended. |

Remove Task

Remove a task in the pool by task id and host id if given.

| Method | URI |
|----------------------------|--|
| DELETE | /tasks |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE -H 'Content-Type: application/json' https://###.###.###.###:30100/tasks?taskId=#####-####-####-####-##### |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE -H 'Content-Type: application/json' https://###.###.###.###:30100/tasks?taskId=#####-####-####-####-##### hostId=#####-####-####-####-##### |

Normal response codes: 200

Request

URI parameters:

| Parameter Name | Type | Is Mandatory | Description |
|----------------|--------|--------------|----------------|
| taskId | String | Mandatory | Task ID. |
| hostId | String | optional | The host UUID. |

Response

Example 2: *Remove Task*: JSON response

```
{
  "description": "Success",
  "status": "Success"
}
```

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none"><i>Success</i> – Success.<i>NoTask</i> – Task with task ID does not exist.<i>OngoingTask</i> – Cannot remove ongoing tasks.<i>TaskBelongsToOtherHost</i> – The task belongs to another host. |

Telemetry API’s

Get Metrics

Shows all Prometheus metrics collected from the provisioner.

| Method | URI |
|--|----------|
| GET | /metrics |
| Linux Curl Command Example curl -k --cert ./ssdtoolbox.pem https://###.###.###.###:30100/metrics | |

Normal response codes: 200

Request

No URI parameters.

Response

Example: Get Metrics response:

```
# HELP ks_node_state ks node state per Storage Node
# TYPE ks_node_state gauge

ks_node_state{node="00:0c:29:4b:b3:b7",name="ks-node7-000c294bb3b7",rack="null",zone="null",region="null",} 0.0
ks_node_state{node="00:0c:29:b8:b0:05",name="ks-node8-000c29b8b005",rack="null",zone="null",region="null",} 0.0
ks_node_state{node="00:0c:29:cd:26:f2",name="ks-node1-000c29cd26f2",rack="null",zone="null",region="null",} 0.0
ks_node_state{node="00:0c:29:08:b9:fc",name="ks-node4-000c2908b9fc",rack="null",zone="null",region="null",} 0.0
ks_node_state{node="00:0c:29:9e:2b:4d",name="ks-node2-000c299e2b4d",rack="null",zone="null",region="null",} 0.0
ks_node_state{node="00:0c:29:0f:60:62",name="ks-node6-000c290f6062",rack="null",zone="null",region="null",} 0.0
ks_node_state{node="00:0c:29:07:44:f1",name="ks-node5-000c290744f1",rack="null",zone="null",region="null",} 0.0
ks_node_state{node="00:0c:29:27:86:09",name="ks-node3-000c29278609",rack="null",zone="null",region="null",} 0.0

# HELP ks_vol_capacity_bytes ks vol capacity bytes per volume
# TYPE ks_vol_capacity_bytes gauge

ks_vol_capacity_bytes{uuid="#####-####-####-####-##### ",alias="pvc-7f07b3c8-be9c-4dbf-b1a6-1f2d72a9fe4b",n
ks_vol_capacity_bytes{uuid="#####-####-####-####-##### ",alias="pvc-6304471d-5f39-4c02-a1f8-7b5cac6b7b10",n
ks_vol_capacity_bytes{uuid="#####-####-####-####-##### ",alias="pvc-7562688a-fa9d-4d77-9678-d39f41203e19",n
ks_vol_capacity_bytes{uuid="#####-####-####-####-##### ",alias="pvc-b5f77a21-604d-4380-80cf-843b761a8bc2",n
ks_vol_capacity_bytes{uuid="#####-####-####-####-##### ",alias="pvc-09a7bbe0-dcde-413a-b048-ef8c790882a4",n
ks_vol_capacity_bytes{uuid="#####-####-####-####-##### ",alias="pvc-bbe1e338-20cd-4320-8a6d-6acfa4623b16",n

# HELP ks_node_used_capacity_bytes ks node used capacity bytes per Storage Node
```

```
# TYPE ks_node_used_capacity_bytes gauge

ks_node_used_capacity_bytes{node="00:0c:29:0f:60:62",name="ks-node6-000c290f6062",} 2.01326592E8

ks_node_used_capacity_bytes{node="00:0c:29:27:86:09",name="ks-node3-000c29278609",} 6.4726499328E10

ks_node_used_capacity_bytes{node="00:0c:29:b8:b0:05",name="ks-node8-000c29b8b005",} 1.34217728E8

ks_node_used_capacity_bytes{node="00:0c:29:08:b9:fc",name="ks-node4-000c2908b9fc",} 1.34217728E8

ks_node_used_capacity_bytes{node="00:0c:29:4b:b3:b7",name="ks-node7-000c294bb3b7",} 1.34217728E8

ks_node_used_capacity_bytes{node="00:0c:29:9e:2b:4d",name="ks-node2-000c299e2b4d",} 6.5800241152E10

ks_node_used_capacity_bytes{node="00:0c:29:cd:26:f2",name="ks-node1-000c29cd26f2",} 4.3167776768E10

ks_node_used_capacity_bytes{node="00:0c:29:07:44:f1",name="ks-node5-000c290744f1",} 2.01326592E8

# HELP ks_node_used_iops ks node used iops per Storage Node

# TYPE ks_node_used_iops gauge

ks_node_used_iops{node="00:0c:29:0f:60:62",name="ks-node6-000c290f6062",} 0.0

ks_node_used_iops{node="00:0c:29:27:86:09",name="ks-node3-000c29278609",} 13670.0

ks_node_used_iops{node="00:0c:29:b8:b0:05",name="ks-node8-000c29b8b005",} 0.0

ks_node_used_iops{node="00:0c:29:08:b9:fc",name="ks-node4-000c2908b9fc",} 0.0

ks_node_used_iops{node="00:0c:29:4b:b3:b7",name="ks-node7-000c294bb3b7",} 0.0

ks_node_used_iops{node="00:0c:29:9e:2b:4d",name="ks-node2-000c299e2b4d",} 14010.0

ks_node_used_iops{node="00:0c:29:cd:26:f2",name="ks-node1-000c29cd26f2",} 6835.0

ks_node_used_iops{node="00:0c:29:07:44:f1",name="ks-node5-000c290744f1",} 0.0

# HELP ks_node_used_bw_bytes_per_sec ks node used bw bytes per sec per Storage Node

# TYPE ks_node_used_bw_bytes_per_sec gauge

ks_node_used_bw_bytes_per_sec{node="00:0c:29:0f:60:62",name="ks-node6-000c290f6062",} 0.0

ks_node_used_bw_bytes_per_sec{node="00:0c:29:27:86:09",name="ks-node3-000c29278609",} 1.7747392E8

ks_node_used_bw_bytes_per_sec{node="00:0c:29:b8:b0:05",name="ks-node8-000c29b8b005",} 0.0

ks_node_used_bw_bytes_per_sec{node="00:0c:29:08:b9:fc",name="ks-node4-000c2908b9fc",} 0.0

ks_node_used_bw_bytes_per_sec{node="00:0c:29:4b:b3:b7",name="ks-node7-000c294bb3b7",} 0.0

ks_node_used_bw_bytes_per_sec{node="00:0c:29:9e:2b:4d",name="ks-node2-000c299e2b4d",} 1.81893504E8

ks_node_used_bw_bytes_per_sec{node="00:0c:29:cd:26:f2",name="ks-node1-000c29cd26f2",} 8.873696E7

ks_node_used_bw_bytes_per_sec{node="00:0c:29:07:44:f1",name="ks-node5-000c290744f1",} 0.0

# HELP ks_node_free_iops ks node free iops per Storage Node

# TYPE ks_node_free_iops gauge

ks_node_free_iops{node="00:0c:29:0f:60:62",name="ks-node6-000c290f6062",} 522900.0

ks_node_free_iops{node="00:0c:29:27:86:09",name="ks-node3-000c29278609",} 334930.0

ks_node_free_iops{node="00:0c:29:b8:b0:05",name="ks-node8-000c29b8b005",} 348600.0

ks_node_free_iops{node="00:0c:29:08:b9:fc",name="ks-node4-000c2908b9fc",} 348600.0

ks_node_free_iops{node="00:0c:29:4b:b3:b7",name="ks-node7-000c294bb3b7",} 348600.0

ks_node_free_iops{node="00:0c:29:9e:2b:4d",name="ks-node2-000c299e2b4d",} 334590.0

ks_node_free_iops{node="00:0c:29:cd:26:f2",name="ks-node1-000c29cd26f2",} 341765.0

ks_node_free_iops{node="00:0c:29:07:44:f1",name="ks-node5-000c290744f1",} 522900.0

# HELP ks_node_free_bw_bytes_per_sec ks node free bw bytes per sec per Storage Node

# TYPE ks_node_free_bw_bytes_per_sec gauge

ks_node_free_bw_bytes_per_sec{node="00:0c:29:0f:60:62",name="ks-node6-000c290f6062",} 6.788481024E9

ks_node_free_bw_bytes_per_sec{node="00:0c:29:27:86:09",name="ks-node3-000c29278609",} 4.348180096E9

ks_node_free_bw_bytes_per_sec{node="00:0c:29:b8:b0:05",name="ks-node8-000c29b8b005",} 4.525654016E9

ks_node_free_bw_bytes_per_sec{node="00:0c:29:08:b9:fc",name="ks-node4-000c2908b9fc",} 4.525654016E9

ks_node_free_bw_bytes_per_sec{node="00:0c:29:4b:b3:b7",name="ks-node7-000c294bb3b7",} 4.525654016E9

ks_node_free_bw_bytes_per_sec{node="00:0c:29:9e:2b:4d",name="ks-node2-000c299e2b4d",} 4.343760512E9

ks_node_free_bw_bytes_per_sec{node="00:0c:29:cd:26:f2",name="ks-node1-000c29cd26f2",} 4.436917056E9

ks_node_free_bw_bytes_per_sec{node="00:0c:29:07:44:f1",name="ks-node5-000c290744f1",} 6.788481024E9

# HELP ks_node_free_capacity_bytes ks node free capacity bytes per Storage Node

# TYPE ks_node_free_capacity_bytes gauge

ks_node_free_capacity_bytes{node="00:0c:29:0f:60:62",name="ks-node6-000c290f6062",} 2.19882192896E12

ks_node_free_capacity_bytes{node="00:0c:29:27:86:09",name="ks-node3-000c29278609",} 1.034785128448E12

ks_node_free_capacity_bytes{node="00:0c:29:b8:b0:05",name="ks-node8-000c29b8b005",} 1.099377410048E12

ks_node_free_capacity_bytes{node="00:0c:29:08:b9:fc",name="ks-node4-000c2908b9fc",} 1.099377410048E12

ks_node_free_capacity_bytes{node="00:0c:29:4b:b3:b7",name="ks-node7-000c294bb3b7",} 2.74743689216E11
```

| | |
|---|-------------------|
| ks_node_free_capacity_bytes{node="00:0c:29:9e:2b:4d",name="ks-node2-000c299e2b4d",} | 1.033711386624E12 |
| ks_node_free_capacity_bytes{node="00:0c:29:cd:26:f2",name="ks-node1-000c29cd26f2",} | 7.81465944064E11 |
| ks_node_free_capacity_bytes{node="00:0c:29:07:44:f1",name="ks-node5-000c290744f1",} | 1.649066115072E12 |

Volume Parameters:

| Metric Name | Type | Sampled value | Source |
|--|-------|--|-------------|
| ks_vol_capacity_bytes | Gauge | Capacity (in bytes) | Get Volumes |
| <div>Labels:</div> <div><ul style="list-style-type: none">uuid - UUID of the parent volume in case of replicated Volume or volume in case of simple volume.alias - Volume's aliasnumReplicas - Number of replicastenantID - tenant ID (0 for default tenant)storageClassName - Storage Class name or "unknown" if not providedprovisioningType - Thin, thick, "Snapshot/Clone"protocol - NVMeoF or LocalrepUUIDX - For replicated volume – UUID of the X replica (blank for simple)nodeIDX - For replicated volume – node ID (Persistent ID) of the X replica. For simple volume, Node ID of the volumenodeNameX - - For replicated volume – the node name of the X replica. For simple volume, Node name of the volume.</div> | | | |
| <div>ks_connected_volumes_state</div> <div><div></div></div> <div>explanation:</div> <div>The state considered “Available” if at least one replica has “Available” ReplicaState.</div> <div>* Only reported for NVMe-oF volumes which are published</div> | Gauge | <div>State:</div> <div>0= Available</div> <div>1= UnAvailable</div> | Get Volumes |
| <div>Labels:</div> <div><ul style="list-style-type: none">uuid - UUID of the parent volume in case of replicated Volume or volume in case of simple volume.alias - Volume's alias.hostId - Connected host UUID.hostname – the host name.nqn - Nqn of connected host.version - Software version of the host's agent.</div> | | | |
| <div>ks_connected_replica_state</div> <div>* Only reported for NVMe-oF volumes which are published</div> | Gauge | <div>State:</div> <div>0=Available</div> <div>1=Terminating</div> <div>2=Missing</div> <div>3=Unknown</div> <div>4=Synchronizing</div> | Get Volumes |
| <div>Labels:</div> <div><ul style="list-style-type: none">uuid - UUID of the parent volume.alias- Alias of the parent volume.repUUID- UUID of the replica.hostId-Connected host UUID.hostname – the host name.Nqn- Nqn of connected host.</div> | | | |

Storage Node Parameters:

| Metric Name | Type | Sampled value | Source |
|-----------------------------|-------|-----------------------|---------------|
| ks_node_free_capacity_bytes | Gauge | Free Capacity (Bytes) | List Backends |

| | | | |
|--|-------|--|---------------|
| Labels: <ul style="list-style-type: none">node- KS persistence ID.name - KS name. | | | |
| ks_node_used_capacity_bytes | Gauge | Used Capacity (bytes) | List Backends |
| Labels: <ul style="list-style-type: none">node- KS persistence ID.name - KS name. | | | |
| ks_node_free_iops | Gauge | Free IOps (IO per second) | List Backends |
| Labels: <ul style="list-style-type: none">node- KS persistence ID.name - KS name. | | | |
| ks_node_used_iops | Gauge | Used IOps (IO per sec) | List Backends |
| Labels: <ul style="list-style-type: none">node- KS persistence ID.name - KS name. | | | |
| ks_node_free_bw_bytes_per_sec | Gauge | Free bandwidth (bytes per sec) | List Backends |
| Labels: <ul style="list-style-type: none">node- KS persistence ID.name - KS name. | | | |
| ks_node_used_bw_bytes_per_sec | Gauge | Used bandwidth (bytes per sec) | List Backends |
| Labels: <ul style="list-style-type: none">node- KS persistence ID.name - KS name. | | | |
| ks_node_state | Gauge | State: 1=Available 2=Unavailable | List Backends |
| Labels: <ul style="list-style-type: none">node - KS persistence ID.name - KS name.rack- KS location Rack ID.zone- KS location zone ID.region- KS location region ID. | | | |

Authorization Server Config API’s

Set Authorization Server

Enable to configure authorization server for OpenIDC authentication mode.

| | |
|--------------------------------|--|
| Method | URI |
| POST | /auth_server |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -X POST -H 'Content-Type: application/json' -d '{\"authorizationServerPublicKey\":\"BIGLONGSTRING\", \"authorizationServerTokenURL\":\"http://###.###.###.###:8080/auth/realms/Dev/protocol/openid-connect/token\", \"backendsClientID\":\"liorf-backend\", \"backendsResourceID\":\"liorf-backend\", \"backendsClientSecret\":\"secretstring\", \"provisionerClientID\":\"liorf-prov\", \"provisionerResourceID\":\"liorf-prov\", \"provisionerClientSecret\":\"secretstring\"}' https://##.##.##.###:30100/auth_server |
| Linux Curl Command Example for | curl -k --cert ./ssdtoolbox.pem -X POST -H 'Content-Type: application/json' -d '{\"authorizationServerJwkSetURL\":\"https://dc-dev.koalab.il/adfs/discovery/keys\", \"authorizationServerTokenURL\":\"https://dc-dev.koalab.il/adfs/oauth2/token\", \"provisionerClientID\":\"provisioner\", |

| | |
|--|--|
| Active Directory Federation Service (ADFS) | "provisionerResourceID":"provisioner", "provisionerClientSecret":"secretstring", "provisionerClientScope":"kumoscale/openid", "backendsClientID":"kumoscale", "backendsResourceID":"kumoscale", "backendsClientSecret":"secretstring", "backendsClientScope":"provisioner/openid"}' https://###.###.###:30100/auth_server |
|--|--|

Normal response codes: 200

Request

No URI parameters.

Example 1: *Set Authorization Server*: JSON request

```
{  
  
  "authorizationServerPublicKey": "BIGLONGSTRING",  
  "authorizationServerTokenURL": "http://###.###.###:8080/auth/realms/Dev/protocol/openid-connect/token",  
  
  "backendsClientID": "liorf-backend",  
  
  "backendsResourceID": "liorf-backend",  
  
  "backendsClientSecret": "secretstring",  
  
  "provisionerClientID": "liorf-prov",  
  
  "provisionerResourceID": "liorf-prov",  
  
  "provisionerClientSecret": "secretstring"  
  
}
```

| Parameter Name | Type | Is Mandatory | Description |
|------------------------------|--------|--------------|---|
| provisionerResourceID | String | Mandatory | ID of the provisioner resource – the client name as defined in the authorization server of client which reflects provisioner |
| ProvisionerClientID | String | Mandatory | ID of the provisioner client, which has a service account role ADMIN at provisioner resource |
| provisionerClientSecret | String | Mandatory | The provisioner client secret |
| authorizationServerPublicKey | String | Mandatory | public key used to decrypt tokens |
| authorizationServerTokenURL | String | Mandatory | authorization server url for generating tokens |
| backendsResourceID | String | Mandatory | ID of backends resource – the client name as defined in the authorization server of client which reflects backends. |
| backendsClientID | String | Mandatory | ID of a backend client, which has a service account role ADMIN at backends resource |
| backendsClientSecret | String | Mandatory | The backends client secret |
| authorizationServerJwkSetURL | String | Optional | A URL to retrieve the current keys which are used by the authorization server to sign tokens. Exactly one of this field or the public key must exist. |
| provisionerClientScope | String | Optional | The scope to use while generating a token for the Provisioner client; the scope for the storage node scope. |
| backendsClientScope | String | Optional | The scope to use while generating a token for the storage node client; the scope for the Provisioner. |

Response

Example 2: *Set Authorization Server*: JSON response

```
{  
  
  "description": "Success",  
  
  "status": "Success",  
  
}
```

| } | | |
|----------------|--------|--|
| Parameter Name | Type | Description |
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none">• <i>Success</i> - Success• <i>CreateProvisionerClientTokenFailed</i> – Could not create a provisioner client token• <i>CreateBackendsClientTokenFailed</i> - Could not create a backends client token• <i>DecryptProvisionerClientTokenFailed</i> – Could not decrypt the provisioner client token• <i>DecryptBackendsClientTokenFailed</i> - Could not decrypt the backends client token• <i>ProvisionerClientUnauthorized</i> – The provisioner client token has no ADMIN role• <i>BackendsClientUnauthorized</i> – The backends client token has no ADMIN role• <i>NotAvailable</i> - Not all backends are available• <i>NoChange</i> – No configuration change |

Delete Authorization Server

Enable to configure authorization server back to JWT authentication mode.

| | |
|----------------------------|--|
| Method | URI |
| DELETE | /auth_server |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem -i -X DELETE --header "Content-Type: application/json" https://###.##.##.###:30100/auth_server |

Normal response codes: 200

Request

No URI parameters.

Response

Example 2: *Delete Authorization Server*: JSON response

| |
|---|
| { "description":"Success", "status":"Success", } |
|---|

| Parameter Name | Type | Description |
|----------------|--------|--|
| description | String | Result description. |
| status | String | Result code. <ul style="list-style-type: none">• <i>Success</i> - Success• <i>InvalidAuthMode</i> – The current authentication mode is not OpenIDC• <i>NotAvailable</i> - Not all backends are available |

Get Authorization Server

Get the authorization server configuration.

| | |
|----------------------------|---|
| Method | URI |
| GET | /auth_server |
| Linux Curl Command Example | curl -k --cert ./ssdtoolbox.pem https://###.##.##.###:30100/auth_server |

Normal response codes: 200

Request

No URI parameters.

Response

Example 2: *Get Authorization Server*: JSON response

| |
|---|
| { |
|---|

```
"authorizationServerPublicKey":"BIGLONGSTRING",
"authorizationServerTokenURL":"http://###.##.##.###:8080/auth/realms/Dev/protocol/openid-connect/token",

"backendsClientID":"liorf-backend",

"backendsResourceID":"liorf-backend",

"backendsClientSecret":"secretstring",

"provisionerClientID":"liorf-prov",

"provisionerResourceID":"liorf-prov",

"provisionerClientSecret":"secretstring"

}
```

| Parameter Name | Type | Description |
|------------------------------|--------|--|
| provisionerResourceID | String | ID of the provisioner resource – the client name as defined in the authorization server of client which reflects provisioner |
| ProvisionerClientID | String | ID of the provisioner client, which has a service account role ADMIN at provisioner resource |
| provisionerClientSecret | String | The provisioner client secret |
| authorizationServerPublicKey | String | Public key used to decrypt tokens |
| authorizationServerTokenURL | String | Authorization server url for generating tokens |
| backendsResourceID | String | ID of backends resource – the client name as defined in the authorization server of client which reflects backends. |
| backendsClientID | String | ID of a backend client, which has a service account role ADMIN at backends resource |
| backendsClientSecret | String | The storage nodes client secret |
| authorizationServerJwkSetURL | String | A URL to retrieve the current keys which are used by the authorization server to sign tokens. |
| provisionerClientScope | String | The scope to use while generating a token for the Provisioner client; the storage node scope. |
| backendsClientScope | String | The scope to use while generating a token for the storage node client; the Provisioner scope. |

Next: [REST API Client](#)