IBM Spectrum Protect Plus
Version 10.1.3

vSnap Rest API Developer Guide

IBM

Table of Contents

1. General

API requests should be in the form `https://server:8900/api/endpoint`. For example: `https://vsnap-dev-appliance:8900/api/system`.

API requests and responses are in JSON format.

The API does not support session handling at present. Every API request header must specify the username and password using HTTP basic authentication. For example:

```
Authorization: Basic QWxhZGRpbjpPcGVuU2VzYW1l
```

For successful calls, the API returns status code `200`. In case of an error, it returns an appropriate status code e.g. `401` (invalid credentials), `403` (permission denied), or `500` (internal error). For each of these errors, the response body contains an `error` object which has a `type` and a `message`.

Sample error response:

```
{
    "error": {
        "message": "Disk UUID 12345 is not a valid candidate for a pool",
        "type": "InvalidCandidateError"
    }
}
```

2. System

2.1. Get system info

Returns basic system information.

Request

```
GET /api/system
```

Response

```
{
    "fqdn": "sid-vsnap-dev",
    "hostname": "sid-vsnap-dev",
    "id": "69813aef15146b54092deac8564e1249",
    "init_status": "Not Initialized",
    "multipool": false,
    "api_version": "1.0",
    "nfs_version": "1.3.0-0.33.el7_3",
    "nginx_version": "1.10.2-1.el7",
    "os_name": "CentOS Linux",
    "os_version": "7.3.1611",
    "samba_version": "4.4.4-13.el7_3",
    "uwsgi_version": "2.0.14-6.el7",
    "vsnap_version": "1.0.0-19",
    "zfs_version": "0.6.5.10-1.el7.centos"
}
```

2.2. Initialize system

Request

Parameters:

- `action`: Must be set to `init`.

- `async`: Boolean. If set to `true`, the call returns immediately without waiting for init to complete. Init status can then be monitored using `GET /api/system`. Default: `false`.

- `multipool`: Boolean. If set to `true`, allows for the creation of multiple storage pools. Default: `false`.

- `skip_pool`: Boolean. If set to `true`, all other init steps are performed but pool creation is skipped. Default: `false`.

```
POST /api/system
{
    "action": "init",
    "async": true,
    "multipool": false,
    "skip_pool": false
}
```

Response

```
{
    "fqdn": "sid-vsnap-dev",
    "hostname": "sid-vsnap-dev",
    "id": "69813aef15146b54092deac8564e1249",
    "init_status": "Initializing",
    "multipool": false,
    "nfs_version": "1.3.0-0.33.el7_3",
    "nginx_version": "1.10.2-1.el7",
    "os_name": "CentOS Linux",
    "os_version": "7.3.1611",
    "samba_version": "4.4.4-13.el7_3",
    "uwsgi_version": "2.0.14-6.el7",
    "vsnap_version": "1.0.0-19",
    "zfs_version": "0.6.5.10-1.el7.centos"
}
```

3. Users

3.1. Get users

Returns information about vSnap user accounts.

Note: At present vSnap has only one role named `vsnap_admin` which is granted by adding users to the `vsnap` group. The API returns all OS users that have this role i.e. belong to the `vsnap` group.

Request

```
GET /api/user
```

Response

```
{
    "total": 2,
    "users": [
        {
            "gid": "1000",
            "name": "sbhatt",
            "role": "vsnap_admin",
            "uid": "1000"
        },
        {
            "gid": "1001",
            "name": "vsuser",
            "role": "vsnap_admin",
            "uid": "1001"
        }
    ]
}
```

4. Networks

4.1. Get network interfaces

Returns information about network interfaces.

Request

```
GET /api/network
```

Response

```
{
    "networks": [
        {
            "id": "005056b45302",
            "name": "ens192",
            "hw_addr": "005056b45302",
            "ip4_addrs": ["172.20.46.61"],
            "ip6_addrs": ["fe80::250:56ff:feb4:5302"],
            "services": ["mgmt", "nfs", "smb"]
        }
    ],
    "total": 1
}
```

4.2. Set network interface services

Modifies the services associated with a network interface.

Request

Parameters:

- `services`: List (array) of services associated with this interface. Valid values: `all`, `mgmt`, `nfs`, `smb`, `iscsi`. If `all` is specified, other values in the array are ignored and all available services are used.

```
PUT /api/network/<id>
{
    "services": ["nfs", "smb"]
}
```

```
PUT /api/network/<id>
{
    "services": ["all"]
}
```

Response

The response shows details about the modified interface.

```
{
    "id": "005056b45302",
    "name": "ens192",
    "hw_addr": "005056b45302",
    "ip4_addrs": ["172.20.46.61"],
    "ip6_addrs": ["fe80::250:56ff:feb4:5302"],
    "services": ["nfs", "smb", "iscsi"]
}
```

5. Targets

5.1. Get target ports

Returns list of target ports.

Request

```
GET /api/target
```

Response

```
{
    "targets": [
        {
            "type": "iqn",
            "wwn": "iqn.2017-08.com.catalogic.vsnap:69813aef15146b54092deac8564e1249"
        }
    ],
    "total": 1
}
```

6. Disks

6.1. Get unused disks

Returns all unformatted and unpartitioned disks that are available for creating or adding to a pool.

Request

```
GET /api/disk?used_as=unused
```

Response

```
{
    "disks": [
        {
            "model": "Virtual disk",
            "name": "/dev/sdd",
            "size": "4194304",
            "type": "SCSI",
            "uuid": "36000c298311c49a6b3df90fc0e4082f4",
            "vendor": "VMware",
            "used_as": "unused"
        },
        {
            "model": "Virtual disk",
            "name": "/dev/sdc",
            "size": "4194304",
            "type": "SCSI",
            "uuid": "36000c2926edea86a7ca731e360ecb878",
            "vendor": "VMware"
            "used_as": "unused"
        }
    ],
    "total": 2
}
```

6.2. Rescan disks

Rescans the SCSI bus to detect disk additions or removals.

Request

```
POST /api/disk
{
    "action": "rescan"
}
```

Response

```
{}
```

7. Pools

7.1. Create pool

Creates a new pool using a set of candidate disks.

Request

Parameters:

- `name`: Pool name. 256 chars max.
- `disk_list`: List of `uuid` values of disks to be added to the pool. Optional. If not specified, all available eligible disks are used.
- `pool_type`: Possible values: `raid0`, `raid5`, or `raid6`. Required.
- `compression`: Boolean. Optional. Default is `true`.
- `deduplication`: Boolean. Optional. Default is `false`.
- `syncwrite`: Boolean. Optional. Default is `true`.

```
POST /api/pool
{
    "name": "demo1",
    "disk_list": [],
    "pool_type": "raid6",
    "compression": true,
    "deduplication": true,
    "syncwrite": false
}
```

Response

The response shows detailed information about the newly created pool.

```json
{
    "compression": true,
    "compression_ratio": "1.00x",
    "deduplication": true,
    "deduplication_ratio": "1.00x",
    "diskgroup_size": "8",
    "disks": {
        "raid6_group1": [
            "/dev/disk/by-id/scsi-36000c2907913236e99b5ac26e6d344cb",
            "/dev/disk/by-id/scsi-36000c296b2cf9bb3713f26969facf4eb",
            "/dev/disk/by-id/scsi-36000c29c5530671e23bc54f329662140",
            "/dev/disk/by-id/scsi-36000c293febb97ec85d65aeefd519f2a",
            "/dev/disk/by-id/scsi-36000c29654ce70661e2d360c9ae73e83",
            "/dev/disk/by-id/scsi-36000c293478228096f1c50f355f738b8",
            "/dev/disk/by-id/scsi-36000c298311c49a6b3df90fc0e4082f4",
            "/dev/disk/by-id/scsi-36000c29011b0fd065435a3ccf94a8890"
        ],
        "raid6_group2": [
            "/dev/disk/by-id/scsi-36000c29fdd069485029d31123b9a993f",
            "/dev/disk/by-id/scsi-36000c2936c9a9811b73e7840d3eca311",
            "/dev/disk/by-id/scsi-36000c29397c718dedb3474f7308176e1",
            "/dev/disk/by-id/scsi-36000c29aa4ce736c6e7bd1c480a64dc4",
            "/dev/disk/by-id/scsi-36000c29662a1e5715c18e0939be1e49f",
            "/dev/disk/by-id/scsi-36000c2926edea86a7ca731e360ecb878",
            "/dev/disk/by-id/scsi-36000c29cc6f4e6c6184cb3319f55c3ae"
        ]
    },
    "id": "1",
    "name": "demo1",
    "pool_type": "raid6",
    "size_before_compression": "2668643328",
    "size_before_deduplication": "904680960",
    "size_total": "22613000803",
    "size_free": "22613000803",
    "size_used": "155037",
    "status": "ONLINE",
    "syncwrite": false,
    "health": "100",
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

7.2. List pools

Lists all pools.

Request

```
GET /api/pool
```

Response

```
{
    "pools": [
        {
            "id": "1",
            "name": "demo1",
            "pool_type": "raid6",
            "size_total": "22613000803",
            "size_free": "22613000803",
            "size_used": "155037",
            "status": "ONLINE",
            "health": "100",
            "time_created": "1497365131",
            "time_updated": "1497365131"
        }
    ],
    "total": 1
}
```

7.3. Get pool details

Shows detailed information about the specified pool.

Request

```
GET /api/pool/<id>
GET /api/pool?id=<id>
GET /api/pool?name=<name>
```

Response

```json
{
    "compression": true,
    "compression_ratio": "1.00x",
    "deduplication": true,
    "deduplication_ratio": "1.00x",
    "diskgroup_size": "8",
    "disks": {
        "raid6_group1": [
            "/dev/disk/by-id/scsi-36000c2907913236e99b5ac26e6d344cb",
            "/dev/disk/by-id/scsi-36000c296b2cf9bb3713f26969facf4eb",
            "/dev/disk/by-id/scsi-36000c29c5530671e23bc54f329662140",
            "/dev/disk/by-id/scsi-36000c293febb97ec85d65aeefd519f2a",
            "/dev/disk/by-id/scsi-36000c29654ce70661e2d360c9ae73e83",
            "/dev/disk/by-id/scsi-36000c293478228096f1c50f355f738b8",
            "/dev/disk/by-id/scsi-36000c298311c49a6b3df90fc0e4082f4",
            "/dev/disk/by-id/scsi-36000c29011b0fd065435a3ccf94a8890"
        ],
        "raid6_group2": [
            "/dev/disk/by-id/scsi-36000c29fdd069485029d31123b9a993f",
            "/dev/disk/by-id/scsi-36000c2936c9a9811b73e7840d3eca311",
            "/dev/disk/by-id/scsi-36000c29397c718dedb3474f7308176e1",
            "/dev/disk/by-id/scsi-36000c29aa4ce736c6e7bd1c480a64dc4",
            "/dev/disk/by-id/scsi-36000c29662a1e5715c18e0939be1e49f",
            "/dev/disk/by-id/scsi-36000c2926edea86a7ca731e360ecb878",
            "/dev/disk/by-id/scsi-36000c29cc6f4e6c6184cb3319f55c3ae"
        ]
    },
    "id": "1",
    "name": "demo1",
    "pool_type": "raid6",
    "size_before_compression": "2668643328",
    "size_before_deduplication": "904680960",
    "size_total": "22613000803",
    "size_free": "22613000803",
    "size_used": "155037",
    "status": "ONLINE",
    "syncwrite": false,
    "health": "100",
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

7.4. Update pool

The following operations are available for modifying a pool.

7.4.1. Modify pool properties

Updates a pool properties like its name and enables or disables compression and deduplication.

Request

Parameters:

- `name`: Pool name. If not specified, the existing name is retained.
- `compression`: Boolean. If not specified, the existing setting is retained.
- `deduplication`: Boolean. If not specified, the existing setting is retained.
- `syncwrite`: Boolean. If not specified, the existing setting is retained.

```
PUT /api/pool/<id>
{
    "name": "newName",
    "compression": false,
    "deduplication": true,
    "syncwrite": true
}
```

The **Response** returns the pool details, similar to the response from `GET /api/pool/<id>`.

7.4.2. Expand pool capacity

Adds disks to expand the pool capacity.

Request

Parameters:

- `action`: Must be set to `expand`.
- `disk_list`: List of `uuid` values of disks to be added to the pool. Optional. If not specified, all available eligible disks are used.

```
POST /api/pool/<id>
{
    "action": "expand",
    "disk_list": []
}
```

The **Response** returns the pool details, similar to the response from `GET /api/pool/<id>`.

7.4.3. Add or remove log disk

Adds or removes separate disks for use as the ZFS Intent Log. Optional. By default, the ZIL is allocated from blocks within the main pool. However, it might be possible to get better performance using separate ZIL devices such as a dedicated SSD. If multiple disks are specified, the ZIL is mirrored across them for redundancy.

Request

Parameters:

- `action`: Must be set to `addlog` or `removelog`.
- `disk_list`: Required if `action` is `addlog`. Specify list of `uuid` values of disks to add. If multiple disks are specified, the log is mirrored across them. This parameter is ignored when `action` is `removelog`.

```
POST /api/pool/<id>
{
    "action": "addlog",
    "disk_list": ["36000c29cc6f4e6c6184cb3319f55c3ae",
"36000c29cc6f4e6c6184cb3319f55c3af"]
}
```

```
POST /api/pool/<id>
{
    "action": "removelog"
}
```

The **Response** returns the pool details, similar to the response from `GET /api/pool/<id>`.

7.4.4. Add or remove cache disk

Adds or removes a disk for use as the cache. Optional. For read-heavy workloads where the working set size is much larger than what can be cached in main memory, using a cache device allow much more of this working set to be served from low latency media.

Request

Parameters:

- `action`: Must be set to `addcache` or `removecache`.
- `disk`: Required if `action` is `addcache`. Specify the `uuid` value of the the disk to add. This parameter is ignored when `action` is `removecache`.

```
POST /api/pool/<id>
{
    "action": "addcache",
    "disk": "36000c29cc6f4e6c6184cb3319f55c3ae"
}
```

```
POST /api/pool/<id>
{
    "action": "removecache"
}
```

The **Response** returns the pool details, similar to the response from `GET /api/pool/<id>`.

7.4.5. Add or remove spare disk

Adds or removes hot spare disks. Optional. These devices are not actively used in the pool, but when an active device fails, it is automatically replaced by a hot spare if available.

Request

Parameters:

- `action`: Must be set to `addspare` or `removespare`.

- `disk_list`: Specify list of `uuid` values of disks to add or remove from the spares list.

```
POST /api/pool/<id>
{
    "action": "addspare",
    "disk_list": ["36000c29cc6f4e6c6184cb3319f55c3ae",
"36000c29cc6f4e6c6184cb3319f55c3af"]
}
```

```
POST /api/pool/<id>
{
    "action": "removespare",
    "disk_list": ["36000c29cc6f4e6c6184cb3319f55c3ae"]
}
```

The **Response** returns the pool details, similar to the response from `GET /api/pool/<id>`.

7.5. Delete pool

Deletes the specified pool. Deleting a pool also deletes all other objects associated with it i.e. volumes, snapshots, clones, shares.

Request

```
DELETE /api/pool/<id>
```

Response

```
{}
```

8. Volumes

8.1. Create volume

Creates a new volume under a specific pool.

Request

Parameters (if creating a new volume):

- `name`: Volume name.
- `volume_type`: `filesystem` or `lun` or `filebackedlun`.
- Additional parameters depend on `volume_type`.
- If `volume_type` is `filesystem` or `lun`: + `pool_id`: ID of the parent pool. + `size_total`: The size limit (in bytes). Required for `lun`. Optional for `filesystem`. + `size_block`: The physical block size (in bytes). Applicable to `lun` volumes only. Optional. Default: 65536.
- If `volume_type` is `filebackedlun`: + `parent_id`: ID of the parent volume (of type `filesystem`) where the file is located. + `parent_path`: Path to the backing file, relative to the root of the parent `filesystem`.

Parameters (if cloning a volume from a snapshot):

- `name`: Volume name.
- `snapshot_id`: ID of the snapshot from which to clone the volume.

```
POST /api/volume
{
    "name": "my_new_vol",
    "volume_type": "lun",
    "pool_id": "1",
    "size_total": "1073741824",
    "size_block": "512"
}
```

```
POST /api/volume
{
    "name": "my_clone_vol",
    "snapshot_id": "5"
}
```

```
POST /api/volume
{
    "name": "my_file_lun",
    "volume_type": "filebackedlun",
    "parent_id": "1",
    "parent_path": "my_files/file1.img"
}
```

Response

The response shows detailed information about the newly created volume.

```
{
    "id": "1",
    "name": "my_new_vol",
    "volume_type": "lun",
    "is_clone": false,
    "serial": "68d65662-0bbf-4fd6-90f0-85a5d62df761",
    "pool_id": "1",
    "pool_name": "demo1",
    "size_total": "1073741824",
    "size_free": "1073741824",
    "size_used": "35856",
    "parent_id": null,
    "parent_path": null,
    "snapshot_id": null,
    "share_id": null,
    "host_ids": [],
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

Sample response for volume of type `filebackedlun`:

```
{
    "id": "2",
    "name": "my_file_lun",
    "volume_type": "filebackedlun",
    "is_clone": false,
    "serial": "68d65662-0bbf-4fd6-90f0-85a5d62df761",
    "pool_id": "1",
    "pool_name": "demo1",
    "size_total": null,
    "size_free": null,
    "size_used": null,
    "parent_id": "1",
    "parent_path": "my_files/file1.img",
    "snapshot_id": null,
    "share_id": null,
    "host_ids": [],
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

8.2. List volumes

Request

```
GET /api/volume
```

Response

```json
{
    "volumes": [
        {
            "id": "1",
            "volume_type": "filesystem",
            "is_clone": false,
            "name": "myfs1",
            "pool_id": "1",
            "size_total": null,
            "size_free": "1073741824",
            "size_used": "35856",
            "time_created": "1497365131",
            "time_updated": "1497365131"
        },
        {
            "id": "2",
            "volume_type": "lun",
            "is_clone": false,
            "name": "myvol2",
            "pool_id": "1",
            "size_total": "22613000803",
            "size_free": "22613000803",
            "size_used": "35856",
            "time_created": "1497365131",
            "time_updated": "1497365131"
        }
    ],
    "total": 2
}
```

8.3. Get volume details

Show detailed information about the specified volume.

Request

```
GET /api/volume/<id>
GET /api/volume?id=<id>
GET /api/volume?name=<name>
```

Response

```
{
    "id": "1",
    "name": "my_new_vol",
    "volume_type": "lun",
    "is_clone": false,
    "serial": "68d65662-0bbf-4fd6-90f0-85a5d62df761",
    "pool_id": "1",
    "pool_name": "demo1",
    "size_total": "1073741824",
    "size_free": "1073741824",
    "size_used": "35856",
    "parent_id": null,
    "parent_path": null,
    "snapshot_id": null,
    "share_id": null,
    "host_ids": [],
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

8.4. Update volume

TODO

8.5. Delete volume

Deletes the specified volume. Deleting a volume also deletes any associated share (if filesystem) or detaches it from any associated host (if lun). It also deletes all snapshots of the volume. However, deletion will throw an error if any of the snapshots have active clones.

Request

```
DELETE /api/volume/<id>
```

Response

```
{}
```

8.6. Create snapshot of volume

Create a new snapshot of the specified volume.

Request

Parameters:

- `name`: Snapshot name. Optional. If not specified, a unique name is generated internally.

```
POST /api/volume/<id>/snapshot
{
    "name": "ecx_1007_1234567890_myvol1"
}
```

Response

The response shows detailed information about the newly created snapshot.

```
{
    "id": "1",
    "name": "ecx_1007_1234567890_myvol1",
    "parent_id": "1",
    "parent_name": "myvol1",
    "pool_id": "1",
    "pool_name": "demo1",
    "has_clones": true,
    "size_used": "35856",
    "version_id": "6021379531292140614",
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

8.7. List snapshots of volume

Lists all snapshots of the specified volume.

Request

```
GET /api/volume/<id>/snapshot
```

Response

```json
{
    "snapshots": [
        {
            "id": "1",
            "name": "ecx_1007_1234567890_myvol1",
            "parent_id": "1",
            "time_created": "1497365131",
            "time_updated": "1497365131"
        },
        {
            "id": "2",
            "name": "ecx_1007_9876543210_myvol1",
            "parent_id": "1",
            "time_created": "1497369523",
            "time_updated": "1497369523"
        }
    ],
    "total": 2
}
```

8.8. Create share of volume

Applicable only to volumes of type `filesystem`

Exports the specified filesystem as an NFS or SMB share. Creating a share immediately makes the filesystem accessible to external clients. Each filesystem can only have one share associated with it at any given time.

Request

Parameters:

- `share_type`: Possible values: `nfs` or `smb`.
- `share_options` for `nfs` shares:
  - `read_only`: Boolean. Optional. Default is `false` (i.e. shares are read-write)
  - `allowed_hosts`: List of hosts allowed to access the share. Optional. If not specified, the share is not accessible from external hosts. Hosts can be specified as hostname (e.g. `host.domain.com`), IP address (e.g. `192.168.1.128`), or subnet (e.g. `192.168.1.0/24`). Specify `all` to allow access to all hosts.
- `share_options` for `smb` shares:
  - No share options currently supported for `smb`.

```
POST /api/volume/<id>/share
{
    "share_type": "nfs",
    "share_options": {
        "read_only": false,
        "allowed_hosts": ["all"]
    }
}
```

Response

The response shows detailed information about the newly created share. The `name` attribute in the response is the NFS path or SMB name that clients can use to access the share.

```
{
    "id": "1",
    "name": "/data/vol1",
    "vol_id": "1",
    "share_type": "nfs",
    "share_options": {
        "read_only": false,
        "allowed_hosts": [
            "172.20.0.0/16",
            "172.19.1.6"
        ]
    },
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

8.9. Delete share of volume

Applicable only to volumes of type `filesystem`

Deletes the NFS or SMB share associated with the specified volume, if a share exists. Deleting the share immediately makes the filesystem inaccessible to external clients.

Request

```
DELETE /api/volume/<id>/share
```

Response

```
{}
```

8.10. Map volume to host

Applicable only to volumes of type `lun` and `filebackedlun`

Attaches the specified lun to a host. This immediately makes the lun accessible to client initiators associated with the host.

Request

Parameters:

- `host_id`: ID of the host to attach the lun to.

```
POST /api/volume/<id>/map
{
    "host_id": "1"
}
```

Response

The response shows the mapped lun information.

```
{
    "vol_id": "1",
    "lun": "0"
}
```

9. Shares

9.1. List shares

List all shares.

Request

```
GET /api/share
```

Response

```
{
    "shares": [
        {
            "id": "1",
            "name": "/data/vol1",
            "vol_id": "1",
            "share_type": "nfs",
            "time_created": "1497365131",
            "time_updated": "1497365131"
        },
        {
            "id": "2",
            "name": "pool1_fs2",
            "vol_id": "2",
            "share_type": "smb",
            "time_created": "1497365131",
            "time_updated": "1497365131"
        }
    ],
    "total": 2
}
```

9.2. Get share details

Show detailed information about the specified share.

Request

```
GET /api/share/<id>
GET /api/share?id=<id>
```

Response

```
{
    "id": "1",
    "name": "/data/vol1",
    "share_type": "nfs",
    "vol_id": "1",
    "share_options": {
        "read_only": false,
        "allowed_hosts": [
            "172.20.0.0/16",
            "172.19.1.6"
        ]
    }
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

9.3. Update share

Applicable only to volumes of type `filesystem`

Request

The request body for updating a share is similar to the one for creating it except for the HTTP method being `PUT` instead of `POST`. The entire body returned by *Get share details* must be sent back in the `PUT` request with the appropriate changes.

Parameters:

- `share_type`: Possible values: `nfs` or `smb`.
- `share_options` for `nfs` shares:
    - `read_only`: Boolean. Optional. Default is `false` (i.e. shares are read-write)
    - `allowed_hosts`: List of hosts allowed to access the share. Optional. If not specified, the share is not accessible from external hosts. Hosts can be specified as hostname (e.g. `host.domain.com`), IP address (e.g. `192.168.1.128`), or subnet (e.g. `192.168.1.0/24`). Specify `all` to allow access to all hosts.
- `share_options` for `smb` shares:
    - No share options currently supported for `smb`.

```
PUT /api/volume/<id>/share
{
    "share_type": "nfs",
    "share_options": {
        "read_only": false,
        "allowed_hosts": [
            "172.20.0.0/16",
            "172.19.1.6"
        ]
    }
}
```

Response

```
{
    "id": "1",
    "name": "/data/vol1",
    "vol_id": "1",
    "share_type": "nfs",
    "share_options": {
        "read_only": false,
        "allowed_hosts": [
            "172.20.0.0/16",
            "172.19.1.6"
        ]
    },
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

9.4. Delete share

Deletes the specified share. Deleting a share immediately unmounts the filesystem and makes it inaccessible to external clients.

Request

```
DELETE /api/share/<id>
```

Response

```
{}
```

10. Hosts

10.1. Create host

Creates a host representing a storage client.

Request

Parameters:

- `name`: Name of the host. Required.

- `iqns`: List of one or more iSCSI initiators associated with the host. Optional.

```
POST /api/host
{
    "name": "sid-oracle-47",
    "iqns": ["iqn.1988-12.com.oracle:devoracle47"]
}
```

Response

The response shows detailed information about the host.

```
{
    "id": "1",
    "name": "sid-oracle-47",
    "iqns": ["iqn.1988-12.com.oracle:devoracle47"],
    "mapped_luns": [],
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

10.2. List hosts

Request

```
GET /api/host
```

Response

```
{
    "hosts": [
        {
            "id": "1",
            "name": "sid-oracle-47",
            "iqns": ["iqn.1988-12.com.oracle:devoracle47"],
            "time_created": "1497365131",
            "time_updated": "1497365131"
        },
        {
            "id": "2",
            "name": "sid-oracle-45",
            "iqns": ["iqn.1988-12.com.oracle:devoracle45"],
            "time_created": "1497365131",
            "time_updated": "1497365131"
        }
    ],
    "total": "2"
}
```

10.3. Get host details

Show detailed information about the specified host.

Request

```
GET /api/host/<id>
GET /api/host?id=<id>
GET /api/host?name=<name>
```

Response

```
{
    "id": "1",
    "name": "sid-oracle-47",
    "iqns": ["iqn.1988-12.com.oracle:devoracle47"],
    "mapped_luns": [
        {
            "lun": "0",
            "vol_id": "1",
        },
        {
            "lun": "1",
            "vol_id": "2"
        }
    ],
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

10.4. Update host

Updates a host's properties such as its name or initiator list.

Request

Parameters:

- `name`: New name for the host. Optional.

- `iqns`: New list of initiators for the host. Optional. Note that the existing initiators will be replaced with the list specified here.

```
PUT /api/host/<id>
{
    "name": "new-host-name"
}
```

```
PUT /api/host/<id>
{
    "name": "new-host-name",
    "iqns": [
        "iqn.1988-12.com.oracle:devoracle46",
        "iqn.1988-12.com.oracle:devoracle47",
    ]
}
```

```
PUT /api/host/<id>
{
    "iqns": []
}
```

Response

```
{
    "id": "1",
    "name": "new-host-name",
    "iqns": [
        "iqn.1988-12.com.oracle:devoracle46",
        "iqn.1988-12.com.oracle:devoracle47"
    ],
    "mapped_luns": [],
    "time_created": "1497365131",
    "time_updated": "1497365131"
}
```

10.5. Map volume to host

Attaches a volume to the specified host.

Request

Parameters:

- `vol_id`: ID of the volume to attach.

```
POST /api/host/<id>
{
    "action": "map",
    "vol_id": "2"
}
```

Response

The response shows the mapped lun information.

```
{
    "vol_id": "2",
    "lun": "1"
}
```

10.6. Unmap volume from host

Detaches a volume from the specified host.

Request

Parameters:

- vol_id: ID of the volume to detach.

```
POST /api/host/<id>
{
    "action": "unmap",
    "vol_id": "2"
}
```

Response

```
{}
```

10.7. Delete host

Deletes the specified host. Any volumes that remain to the host are unmapped.

Request

```
DELETE /api/host/<id>
```

Response

```
{}
```

11. Snapshots

11.1. List snapshots

List all snapshots of all volumes.

Request

```
GET /api/snapshot
```

Response

```
{
    "snapshots": [
        {
            "id": "1",
            "name": "ecx_1007_1234567890_myvol1",
            "parent_id": "1",
            "time_created": "1497365131",
            "time_updated": "1497365131"
        },
        {
            "id": "2",
            "name": "ecx_1007_9876543210_myvol1",
            "parent_id": "1",
            "time_created": "1497369523",
            "time_updated": "1497369523"
        }
    ],
    "total": 2
}
```

11.2. Get snapshot details

Show detailed information about the specified snapshot.

Request

```
GET /api/snapshot/<id>
GET /api/snapshot?id=<id>
GET /api/snapshot?name=<name>
```

Response

```
{
    "id": "1",
    "name": "ecx_1007_1234567890_myvol1",
    "parent_id": "1",
    "parent_name": "myvol1",
    "pool_id": "1",
    "pool_name": "demo1",
    "has_clones": true,
    "size_used": "35856",
    "version_id": "6021379531292140614",
    "time_created": "1497365131",
    "time_updated": "1497365131",
}
```

11.3. Delete snapshot

Deletes the specified snapshot. Deletion will throw an error if the snapshot has one or more active clones.

Request

```
DELETE /api/snapshot/<id>
```

Response

```
{}
```

12. Partners

12.1. Add partner system

Establishes a partnership with a remote vSnap system.

Request

Parameters:

- `remote_addr`: The management address of the remote vSnap system.

- `remote_user`: Username for the remote vSnap system.

- `remote_pass`: Password for the remote vSnap system.

- `local_addr`: Optional. The management address of the local vSnap system.

- `local_api_port`: Optional. The API port of the local vSnap system.

- `remote_api_port`: Optional. The API port of the remote vSnap system.

- `local_ssh_port`: Optional: The SSH port of the local vSnap system.

- `remote_ssh_port`: Optional: The SSH port of the remote vSnap system.

```
POST /api/partner
{
    "remote_addr": "172.20.46.xx",
    "remote_user": "vsuser",
    "remote_pass": "catal0gic",
    "local_addr": null,
    "local_api_port": null,
    "remote_api_port": null,
    "local_ssh_port": null,
    "remote_ssh_port": null
}
```

Response

```
{
    "id": "5058a20323b943ed925fd92f88f1b926",
    "address": "172.20.46.6",
    "api_port": "8900",
    "ssh_port": "22",
    "time_created": "1513856343",
    "time_updated": "1513856344",
}
```

12.2. Get list of partners

Returns a list of partner vSnap systems.

Request

```
GET /api/partner
```

Response

```
{
    "partners": [
        {
            "address": "172.20.46.6",
            "id": "5058a20323b943ed925fd92f88f1b926",
            "api_port": "8900",
            "ssh_port": "22",
            "time_created": "1513856343",
            "time_updated": "1513856344"
        },
        {
            "address": "172.20.46.7",
            "id": "c87c9cee768f4cabb04abd5e215be851",
            "api_port": "8900",
            "ssh_port": "22",
            "time_created": "1513856352",
            "time_updated": "1513856352"
        }
    ],
    "total": 2
}
```

12.3. Get partner details

Get details for a specific partner system.

Request

```
GET /api/partner/<partner_id>
```

Response

```
{
    "id": "5058a20323b943ed925fd92f88f1b926",
    "address": "172.20.46.6",
    "api_port": "8900",
    "ssh_port": "22",
    "time_created": "1513856343",
    "time_updated": "1513856344",
}
```

12.4. Remove partner system

Remove partnership with a remote vSnap system.

Request

```
DELETE /api/partner/<partner_id>
```

Response

```
{}
```

13. Relationships

13.1. Create relationship for volume

Create replication relationship for a volume.

Request

Parameters:

- `partner_id`: The ID of the partner server where the replica volume should be created.
- `remote_pool_id`: The ID of the pool on the partner server where the replica volume should be created. Optional.
- `remote_vol_name`: The name of the replica volume that should be created on the partner server. Optional. If not specified, a default name is chosen automatically.
- `reuse`: If true, reestablish relationship with an existing replica volume on the partner server if a volume already exists. Default: false.

```
POST /api/volume/<id>/relationship
{
    "partner_id": "5058a20323b943ed925fd92f88f1b926",
    "remote_pool_id": "1",
    "remote_vol_name": "my_replica_volume",
    "reuse": false
}
```

Response

```
{
    "id": "ad05bf4db630bfc7ecb706563788f087",
    "local_pool_id": "1",
    "local_role": "primary",
    "local_vol_id": "11",
    "local_vol_name": "my_primary_volume",
    "partner_id": "c87c9cee768f4cabb04abd5e215be851",
    "remote_pool_id": "1",
    "remote_vol_id": "4",
    "remote_vol_name": "my_replica_volume",
    "last_sync_status": null,
    "last_attempt_snap_id": null,
    "last_success_snap_id": null,
    "time_created": "1513887884",
    "time_updated": "1513887884"
}
```

13.2. Get list of relationships

Get list of all relationships.

Request

Get all relationships for a volume ID:

```
GET /api/volume/<id>/relationship
```

Get all relationships for a volume name:

GET /api/relationship?vol_name=my_source_vol ---

Get all relationships of all volumes:

```
GET /api/relationship
```

Response

```
{
    "relationships": [
        {
            "id": "ad05bf4db630bfc7ecb706563788f087",
            "local_pool_id": "1",
            "local_role": "primary",
            "local_vol_id": "11",
            "local_vol_name": "my_primary_volume",
            "partner_id": "c87c9cee768f4cabb04abd5e215be851",
            "remote_pool_id": "1",
            "remote_vol_id": "4",
            "remote_vol_name": "my_replica_volume",
            "last_sync_status": "COMPLETED",
            "last_attempt_snap_id": "22",
            "last_success_snap_id": "22",
            "time_created": "1513887884",
            "time_updated": "1513887884"
        },
        {
            "id": "49e435aecdc5d2455d8b20a3be9da67f",
            "local_pool_id": "1",
            "local_role": "primary",
            "local_vol_id": "5",
            "local_vol_name": "my_primary_volume2",
            "partner_id": "c87c9cee768f4cabb04abd5e215be851",
            "remote_pool_id": "1",
            "remote_vol_id": "5",
            "remote_vol_name": "my_replica_volume2",
            "last_sync_status": "FAILED",
            "last_attempt_snap_id": "21",
            "last_success_snap_id": "20",
            "time_created": "1513888068",
            "time_updated": "1513888068"
        }
    ],
    "total": 2
}
```

13.3. Get relationship details

Get detailed information about a particular relationship.

Request

```
GET /api/relationship/<id>
GET /api/relationship?id=<id>
```

Response if API is called on primary server:

```
{
    "id": "ad05bf4db630bfc7ecb706563788f087",
    "local_pool_id": "1",
    "local_role": "primary",
    "local_vol_id": "11",
    "local_vol_name": "my_primary_volume",
    "partner_id": "c87c9cee768f4cabb04abd5e215be851",
    "remote_pool_id": "1",
    "remote_vol_id": "4",
    "remote_vol_name": "my_replica_volume",
    "last_sync_status": "COMPLETED",
    "last_attempt_snap_id": "22",
    "last_success_snap_id": "22",
    "time_created": "1513887884",
    "time_updated": "1513887884"
}
```

Response if same API is called on target server:

```
{
    "id": "ad05bf4db630bfc7ecb706563788f087",
    "local_pool_id": "1",
    "local_role": "replica",
    "local_vol_id": "4",
    "local_vol_name": "my_replica_volume",
    "partner_id": "5058a20323b943ed925fd92f88f1b926",
    "remote_pool_id": "1",
    "remote_vol_id": "11",
    "remote_vol_name": "my_primary_volume",
    "last_sync_status": null,
    "last_attempt_snap_id": null,
    "last_success_snap_id": null,
    "time_created": "1513887884",
    "time_updated": "1513887884"
}
```

13.4. Remove relationship for volume

Delete a replication relationship. Deleting a relationship automatically deletes the replica volume. The primary volume is not deleted. Optionally, the parameter `preserve_replica` can be set to `1` or `true` to indicate that the relationship should be deleted without deleting the replica volume.

Request

```
DELETE /api/relationship/<id>
DELETE /api/relationship/<id>?preserve_replica=true
```

Response

```
{}
```

14. Sessions

14.1. Start replication session

Start replication for a volume.

Request

Parameters:

- `snap_id`: The ID of the local snapshot to sync. Optional. If not specified, a new snapshot is created and synced.

- `remote_snap_name`: The name of the snapshot that will be created on the remote replica volume. Optional. If not specified, a default name is chosen automatically.

```
POST /api/relationship/<id>/session
{
    "snap_id": "123",
    "remote_snap_name": "my_replica_snap"
}
```

Response

The returned session information contains the following attributes. Note that some of the values may be `null` if the information is not available.

- `id`: Unique ID for the session.

- `rel_id`: The relationship for which the session was initiated.
- `local_snap_id`: The local snapshot ID that is being synced to the remote partner.
- `local_snap_name`: The local snapshot name that is being synced to the remote partner.
- `remote_snap_id`: The remote snapshot ID on the partner server that will be created during sync.
- `remote_snap_name`: The remote snapshot ID on the partner server that will be created during sync.
- `repl_address`: The address of the partner server that is being used for data transfer. This may or may not be the same as the management address. The session will automatically select a suitable network interface that has the `repl` service enabled on the partner server.
- `size_sent`: Size of data that has been sent to the partner server during this session.
- `status`: Session status. Possible values:
    - `PENDING`: Session has been queued but data transfer has not started yet.
    - `ACTIVE`: Session is actively transferring data.
    - `COMPLETED`: Session has completed successfully.
    - `FAILED`: Session has failed.
- `time_queued`: Epoch timestamp for when the session was first created i.e `PENDING` status.
- `time_started`: Epoch timestamp for when the session became `ACTIVE` and started sending data.
- `time_ended`: Epoch timestamp for when the session either completed or failed.
- `message`: String containing brief message about the session. In case of `FAILED` status, this message will contain an error description.

```
{
    "id": "2",
    "rel_id": "46b9beeb33a4a67c2ffa3c517a11e6c8",
    "local_snap_id": "36",
    "remote_snap_id": "37",
    "local_snap_name": "xxx",
    "remote_snap_name": "xxx",
    "repl_address": "172.20.46.6",
    "size_sent": null,
    "status": "PENDING",
    "time_queued": "1514562940",
    "time_started": null,
    "time_ended": null,
    "message": null
}
```

14.2. Get list of sessions

Get a list of all replication sessions.

Request

```
GET /api/session
```

Response

```
{
    "sessions": [
        {
            "id": "2",
            "rel_id": "46b9beeb33a4a67c2ffa3c517a11e6c8",
            "local_snap_id": "36",
            "remote_snap_id": "37",
            "local_snap_name": "xxx",
            "remote_snap_name": "xxx",
            "repl_address": "172.20.46.6",
            "size_sent": null,
            "status": "PENDING",
            "time_queued": "1514562940",
            "time_started": null,
            "time_ended": null,
            "message": null
        }
    ],
    "total": 1
}
```

14.3. Get session details

Get details of a specific replication session.

Request

```
GET /api/session/<id>
```

Response

```
{
    "id": "2",
    "rel_id": "46b9beeb33a4a67c2ffa3c517a11e6c8",
    "local_snap_id": "36",
    "remote_snap_id": "37",
    "local_snap_name": "xxx",
    "remote_snap_name": "xxx",
    "repl_address": "172.20.46.6",
    "size_sent": null,
    "status": "PENDING",
    "time_queued": "1514562940",
    "time_started": null,
    "time_ended": null,
    "message": null
}
```

14.4. Cancel session

Abort an active/pending session.

Request

```
POST /api/session/<id>
{
    "action": "cancel"
}
```

Response

The response returns a session object with the `status` set to `FAILED` and the `message` indicates that the session was canceled.

```
{
    "id": "2",
    "rel_id": "46b9beeb33a4a67c2ffa3c517a11e6c8",
    "local_snap_id": "36",
    "remote_snap_id": "37",
    "local_snap_name": "xxx",
    "remote_snap_name": "xxx",
    "repl_address": "172.20.46.6",
    "size_sent": null,
    "status": "FAILED",
    "time_queued": "1514562940",
    "time_started": null,
    "time_ended": null,
    "message": "Canceled"
}
```

IBM ®

Printed in USA