Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x

F13772-24
May 2024

Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x,

F13772-24

Primary Author: Heidi Hall

Contents

1 Getting Started with the Oracle ZFS Storage Appliance RESTful API

2 Working with the RESTful API

6 Hardware Services

7 Log Commands

8 Network Commands

9 RESTful API Cloud Service

10 RESTful API Problem Service

11 RESTful API SAN Service

12 Service Commands

13 RESTful API Storage Service

14 Storage Encryption

15 System Commands

16 RESTful API User Service

17 RESTful API Role Service

18 Workflow and Script Commands

19 RESTful Clients

1 Getting Started with the Oracle ZFS Storage Appliance RESTful API

Oracle ZFS Storage Appliance provides efficient file and block data services over the network. This guide describes the Oracle ZFS Storage Appliance RESTful Application Programming Interface (API), which can be used to manage the appliance. The RESTful architecture is based on a layered client-server model that lets services be transparently redirected through standard hubs, routers, and other network systems without client configuration.

RESTful API Authentication

The Oracle ZFS Storage Appliance RESTful API uses the same authentication credentials as the browser user interface (BUI) and the command-line interface (CLI). All requests from external clients are individually authenticated using the appliance credentials and are conducted over an HTTPS connection on port 215. The RESTful API supports definable timeouts for HTTPS sessions.

Authentication can take one of the following forms:

- **Basic authentication** – Each request must contain the user login. The authorization string is the concatenation *username*:*password* that is then Base64 encoded.

  Example HTTP Header:

  ```
  Authorization: Basic Tm8gcGVla2luZyE=
  ```

- **User authentication** – BUI or CLI login credentials are used for authentication. In this case, the X-Auth-User header must contain the login name, and the X-Auth-Key header must contain the login password.

  Example HTTP Headers:

  ```
  X-Auth-User:  login-name
  X-Auth-Key:  password-xxx
  ```

  When the RADIUS service is configured, *all* directory users must be authenticated through RADIUS. However, the Oracle ZFS Storage Appliance RESTful API does not support authentication sequences that require multiple prompts and responses, such as a password, a challenge, and a response to the challenge. Note that REST login tokens can be used to bypass this authentication.

  The appliance's BUI and CLI interfaces fully support RADIUS multi-factor authentication for directory users. For more information, see RADIUS Configuration in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

- **Token authentication** – When a token has been authenticated, a token header can be used to continue to run commands until the token expires. After a token expires, authentication must be done again before commands are accepted.

  Example Token Header:

  ```
  X-Auth-Session: qYftpufrrTxlDztkMhllLoyTfSDUSIR
  ```

RESTful API Versions

The RESTful API version for a given release of Oracle ZFS Storage Appliance has a global version number that matches the appliance software version. This version number is returned in the response header of all requests:

```
X-Zfssa-Version:  nas.2013.1.1
```

RESTful API Service Versions

Each RESTful API service has a version number as part of the Uniform Resource Identifier (URI) to access the service. The version has a major and minor number. Requests must supply the major version number, but the minor version number is optional and defaults to a value of 0 if not supplied. The major number must match the major number of the service. The minor number must be less than or equal to the minor number of the service.

For example, the following table shows whether the specified versions could be used in a client request if the client is requesting a service that is running version 2.1.

Request Version	Allowed
1	No: Major version does not match the version the service is running
2	Yes: Major version matches, and the minor version (default 0) is backward compatible
2.1	Yes: Major and minor version values match the version the service is running
2.2	No: Major version matches, but the minor version is newer than the version the service is running

No service API version changes are required for the following property changes. The Oracle ZFS Storage Appliance version number and model must be used to determine which properties are available. These property changes are also reflected in the CLI and BUI and are an indication of the capabilities of that appliance instance.

- New output properties (without removing old properties).
- New input properties added to an existing command, that have default values that make the command behave is it did in an earlier version.

Since a newer version of a backwards-compatible command can return additional properties, clients should be coded to ignore new properties. The minor number is incremented for backwards-compatible changes to the service API.

- Add a new command to an existing service.
- Add new query parameters to service commands.

The major number is incremented with incompatible changes to the service API.

- Removing command query parameters.
- Removing a command from an existing service.

Major releases of appliance software may include incompatible version changes. There may or may not be older versions of a given service during a major update. Each command response must contain an HTTP header with the current version of the appliance API for a given module:

```
X-Zfssa-Nas-Api:  1.1
```

Common RESTful Operations

The following table shows the common RESTful operations for a given resource.

Table 1-1   Common RESTful Operations

Request	Path	Description
GET	resources	List all resources
GET	resources/*name*	Get a JSON object describing the selected resource
POST	resources	Create a new resource
PUT	resources/*name*	Modify the selected resource
DELETE	resources/*name*	Delete the selected resource

HTTP Response Body

All response data is encoded in JSON format as defined by RFC 4627. Unless otherwise specified, commands against a single resource return a single JSON results object with the resource name as a property. Each command section documents which property names are returned in this JSON result object.

Unless otherwise stated, the create (`POST`) and modify (`PUT`) commands return the properties of the created or modified resource. The contents should match the values returned by the `GET` request.

Example Body:

```
{
    "resource_name": {
        "href": "path/to/this/resource",
        "property_01": "value_01",
        "property_02": "value_01
    }
}
```

Some GET commands return a list of resources.

```
{
    "resource_list_name": [
        {
            "href": "path/to/resource_01",
            "property_01": "value_01"
        }, {
            "href": "path/to/resource_02",
            "property_02", "value_02"
        }
    ]
}
```

Note:

Throughout this document, commands show JSON return results that have been formatted by adding returns and spaces to make it more readable. The actual output does not contain this formatting.

HTTP Response Headers

All Oracle ZFS Storage Appliance service commands that send data use the JSON data format and require the following header values:

```
Accept:  application/json
Content-Type:  application/json
```

Response Headers include the following information:

```
Date: Tue, 23 Jul 2013 13:07:37 GMT X-Zfs-Sa-Appliance-Api: 1.0 Content-Type:
application/json Content-Length: 357
```

For list results, the content length may not be known before data is sent back. If the content length is not supplied, the client must read the response body until EOF to read all the returned data.

Appliance Errors

Errors return an HTTP status code that indicates the error along with the following fault response payload.

JSON Fault Response:

```
{
    fault: {
        message: 'ERR_INVALID_ARG',
        details: 'Error Details...',
        code: 500
    }
}
```

Table 1-2   Common Error Codes

Error	Code	Description
ERR_INVALID_ARG	400	Invalid input argument
ERR_UNKNOWN_ARG	400	Extra unhandled input argument
ERR_MISSING_ARG	400	Required input argument missing
ERR_UNAUTHORIZED	401	This user is not authorized to execute command
ERR_DENIED	403	Operation denied
ERR_STATE_CHANGED		Conflict in system state
ERR_NOT_FOUND	404	The requested item was not found

Table 1-2   (Cont.) Common Error Codes

Error	Code	Description
ERR_OBJECT_EXISTS	409	Request creates an object that already exists
ERR_CONFIRM_REQUIRED	409	Request requires the `?confirm=true` query parameter to complete
ERR_OVER_LIMIT	413	Input request too large to handle
ERR_UNSUPPORTED_MEDIA	415	Requested media type is not supported by request
ERR_NOT_IMPLEMENTED	501	Operation not implemented
ERR_BUSY	503	Service not available due to limited resources

Security Protocols and Ciphers Settings

The protocol version and associated cipher commands manage the SSL/TLS protocol versions and ciphers for accessing Oracle ZFS Storage Appliance.

By default, SSL/TLS protocol versions TLSv1.1, TLSv1.2 and their associated ciphers are enabled. You can enable TLSv1.0 by sending a PUT request to the HTTPS service to set the `tls_version` property.

Example Request:

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{ "tls_version":  ["TLSv1.0", "TLSv1.1", "TLSv1.2"] }
```

Example Result (output is omitted for brevity):

```
HTTP/1.1 202 Accepted
Content-Length: 1265
X-Zfssa-Service-Api: 1.1
X-Zfssa-Api-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v1/services/https",
    "<status>": "online",
    "tls_version": "TLSv1 TLSv1.1 TLSv1.2",
    "ciphers": "SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-SHA:
    ...
    3DES-EDE-CBC-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-SHA:DH-RSA-DES-CBC3-SHA:
    DH-DSS-DES-CBC3-SHA:DES-CBC3-SHA"
  }
}
```

To enable TLSv1.0 only, set the `ciphers` property to the list of ciphers available for TLSv1.0 only.

Example Request (output is omitted for brevity):

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{
  "tls_version":  ["TLSv1.0"] ,
  "ciphers" : ["SRP-DSS-AES-256-CBC-SHA", "SRP-RSA-AES-256-CBC-SHA", "SRP-
AES-256-CBC-SHA",
  ...
  "EDH-RSA-DES-CBC3-SHA", "EDH-DSS-DES-CBC3-SHA", "DH-RSA-DES-CBC3-SHA", "DH-DSS-
DES-CBC3-SHA",
  "DES-CBC3-SHA"]
}
```

Example Result (output is omitted for brevity):

```
HTTP/1.1 202 Accepted
Content-Length: 809
X-Zfssa-Service-Api: 1.1
X-Zfssa-Api-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v1/services/https",
    "<status>": "online", "tls_version": "TLSv1",
    "ciphers": "SRP-DSS-AES-256-CBC-SHA:SRP-RSA-AES-256-CBC-SHA:SRP-AES-256-CBC-
SHA:
    ...
    3DES-EDE-CBC-SHA:SRP-3DES-EDE-CBC-SHA:EDH-RSA-DES-CBC3-SHA:EDH-DSS-DES-CBC3-
SHA:DH-
    RSA-DES-CBC3-SHA:DH-DSS-DES-CBC3-SHA:DES-CBC3-SHA"
  }
}
```

Note:

To avoid being blocked from using the RESTful API or the BUI, keep the default settings for the `tls_version` and `ciphers` properties unless otherwise needed or as instructed by Oracle Support.

Session Timeout

The HTTPS service controls the session timeout, which specifies the number of minutes until the browser automatically logs out of the session after user inactivity or if the user navigates away from the BUI. The default value is 15 minutes. This replaces the session timeout property previously located in the user preferences area of the software.

To set the value to other than 15 minutes, send a PUT request to the HTTPS service, and set the `session_timeout` property to a different value.

Example Request:

```
PUT /api/service/v1/services/https HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{ "session_timeout":  5 }
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Length: 1265
X-Zfssa-Service-Api: 1.1
X-Zfssa-Api-Version: 1.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v1/services/https",
    "<status>": "online",
    "tls_version": "default",
    "permit_root_login": true,
    "session_timeout": 5,
    "hsts_enable": false,
    "hsts_max_age": 63072000
  }
}
```

Password Complexity

The password RESTful API enables a user who has the `changeProperties` authorization to set password complexity rules for all local users. For information about user authorizations, see RESTful API Role Service.

Password requirements are enforced when local users change their passwords. Existing passwords are not affected by password rule changes.

Use the following command to show the properties that can be changed.

Example Request:

```
GET /api/setting/v2/password HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Date: Fri, 14 May 2021 17:07:39 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Content-Length: 196

{
  "complexity": {
    "href": "/api/setting/v2/password",
    "passlength": 8,
    "min_letters": 0,
    "min_upper": 0,
    "min_lower": 0,
    "min_digit": 0,
    "min_punctuation": 0,
    "max_repeats": 0,
    "namecheck": true
  }
}
```

For descriptions of these properties, see Password Complexity Properties in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

The following example changes the password rules to require at least one each of upper case letter, lower case letter, number, and punctuation character. The `min_letters` value must be updated to account for the new `min_upper` and `min_lower` values.

Example Request:

```
PUT /api/setting/v2/password HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{"min_letters": 2, "min_upper": 1, "min_lower": 1, "min_digit": 1,
"min_punctuation": 1}
```

Example Result:

```
HTTP/1.1 202 Accepted
Date: Fri, 14 May 2021 17:38:40 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Content-Length: 196

{
  "complexity": {
    "href": "/api/setting/v2/password",
    "passlength": 8,
    "min_letters": 2,
    "min_upper": 1,
    "min_lower": 1,
    "min_digit": 1,
    "min_punctuation": 1,
    "max_repeats": 0,
    "namecheck": true
  }
}
```

RESTful API Service Version 2.0

This section describes differences between RESTful API service version 2 and RESTful API service version 1:

- Scriptable Values
- Consistent Values

Both RESTful API version 2 and RESTful API version 1 are available simultaneously, and the remainder of this guide shows version 1 examples. Use the service version portion of the request URI (`v1` or `v2`) to select the REST API version that you want to use.

Scriptable Values

RESTful API version 2 operations always return scriptable values. A scriptable value has the same stable form for each type of property.

RESTful API version 1 operations often return a scriptable value, but not always. For example, RESTful API version 1 sometimes returns datetime strings in full Javascript date format and sometimes returns datetime strings in the ISO 8601 datetime format. RESTful API version 2 always returns datetime strings in the ISO 8601 datetime format.

In the following example, the `GET /api/system/v1/updates` operation returns datetime strings in full Javascript date format, and the `GET /api/system/v2/updates` operation returns datetime strings in the ISO 8601 datetime format:

```
GET /api/system/v1/updates
{
 "updates": [{
   "status": "previous",
   "href": "/api/system/v1/updates/ak-nas@2013.06.05.4.0,1-1.7",
   "release_date": "Fri May 01 2015 20:13:00 GMT+0000 (UTC)",
   "install_date": "Tue Nov 15 2016 01:01:07 GMT+0000 (UTC)",
   "version": "2013.06.05.4.0,1-1.7",
   "date": "Fri May 01 2015 20:13:00 GMT+0000 (UTC)"
 }]
}
GET /api/system/v2/updates
{
 "updates": [{
   "status": "previous",
   "href": "/api/system/v2/updates/ak-nas@2013.06.05.4.0,1-1.7",
   "release_date": "2015-05-01T20:13:00Z",
   "install_date": "2016-11-15T01:01:07Z",
   "version": "2013.06.05.4.0,1-1.7",
   "date": "2015-05-01T20:13:00Z"
 }]
}
```

Consistent Values

RESTful API version 1 operations sometimes return different values for the same property, depending on how the property is accessed. RESTful API version 2 operations return consistent values, independent of how the property is accessed.

In the following example, when all replication actions are listed, the value of the `max_bandwidth` property returns as `-1`:

```
GET /api/storage/v1/replication/actions
{
 "actions": [{
   "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
   ...
   "max_bandwidth": -1,
   ...
 }]
}
```

When only one replication action is specified, the value of the `max_bandwidth` property returns as `0`, even though the underlying value is unchanged:

```
GET /api/storage/v1/replication/actions/71b1b8b9-9c57-c969-aab9-f96d5f4e5d54
{
 "action": {
   "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
   ...
   "max_bandwidth": 0,
   ...
 }
}
```

RESTful API version 2 operations always return the same value for a particular property, regardless of how that property value is accessed:

```
GET /api/storage/v2/replication/actions
{
 "actions": [{
   "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
   ...
   "max_bandwidth": -1,
   ...
 }]
}
GET /api/storage/v2/replication/actions/71b1b8b9-9c57-c969-aab9-f96d5f4e5d54
{
 "action": {
   "id": "71b1b8b9-9c57-c969-aab9-f96d5f4e5d54",
   ...
   "max_bandwidth": -1,
   ...
 }
}
```

Query Parameters

Some requests can take optional query parameters that select which data are returned or are operated on. This section documents query parameters that can be used by more than one type of resource. See the documentation for each resource for any query parameters that are specific to that resource and for specialized uses of the query parameters that are described in this section.

Table 1-3 Common Query Parameters

Parameter	Description
props=true	List property metadata for a resource; the default value is false
start=*index*	Specify the oldest data or objects to return after the specified time or object ID
end=*index*	Specify the newest data or objects to return before the specified time or object ID
limit=*n*	Return no more than *n* list elements
depth=*n*	Specify the level of detail for the returned data
match_*property-name*=*value*	Return list objects that have the specified property at the specified value

Query Parameter: props

The props query parameter shows property metadata values. When you use props=true with operations that would otherwise change data or create new data, properties and metadata are shown and the operation is *not* performed. This enables you to show current data values that can help you to modify or create the resource.

Table 1-4 Property Metadata Values

Property	Description
name	Property name
label	Description of property
immutable	Flag that indicates that the property cannot be modified
type	Property type such as String, Integer, Boolean, or ChooseOne
choices	For enumerated properties, an array of available values

Query Parameter: start

The start query parameter can be an object index number or a time.

- Specify an object index number to return a list that includes the object selected by that index and the oldest objects that were created after that specified object was created.
- Specify a UTC time to return a list of the oldest objects or data that were created on or after the specified time. Some resources do not support time values for the start query parameter.

Time values must be expressed in UTC time in the format shown in the following table.

Service Version	Time Value Format	Time Value Example
v1 paths	%Y%m%dT%H:%M:%SZ	20200723T14:11:49
v2 paths	%Y-%m-%dT%H:%M:%SZ	2020-07-23T14:11:49Z

See the following sections for examples that use the start query parameter:

- Get Dataset Data
- List Cloud Backups

Query Parameter: end

The end query parameter can be an object index number or a time.

- Specify an object index number to return a list that includes the object selected by that index and the newest objects that were created before that specified object was created.
- Specify a UTC time to return a list of the newest objects or data that were created on or before the specified time.

  Time values must be expressed in UTC time in the format %Y-%m-%dT%H:%M:%SZ.

For an example of using the end query parameter, see List Cloud Backups.

Query Parameter: limit

The `limit` query parameter specifies the maximum number of elements to return.

When neither `start` nor `end` is also specified, `limit=`*n* returns the *n* most recent elements.

Query Parameter: depth

The `depth` query parameter specifies the level of detail for a returned list of resources, where a higher depth value returns more detail as shown in the following table.

Depth Value	Information Returned in List
`depth=0`	Properties of nodes and names of children
`depth=1`	Properties of nodes, names and properties of children, and names of grandchildren
`depth=2`	Properties of nodes, names and properties of children, and `depth=0` output for grandchildren

> **Note:**
>
> The `depth` query parameter is not supported for listing logs using `/api/log/v{1|2}`, or for listing pools, projects, filesystems, or LUNs using `/api/storage/v{1|2}`.

Example request for a list of users using query parameter `depth`:

```
GET /api/user/v1/users?depth=2 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password-xxx
```

Example response:

Additional users are omitted for brevity.

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1558
X-Zfssa-Access-Api: 1.0

{
  "users": [
    {
      "name": "root",
...
    },
    {
      "name": "firstlast",
      "properties": {
        "logname": "firstlast",
        "type": "directory",
        "uid": uid,
        "fullname": "First Last",
        "require_annotation": false,
        "roles": [
          "basic"
        ],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard"
      },
      "children": [
        {
          "name": "exceptions",
          "properties": {},
          "children": [],
          "list": [
            {
              "name": "auth-000",
              "properties": {
                "scope": "ad",
                "name": "*",
                "allow_domain": true,
                "allow_workgroup": false
              },
              "children": [],
              "list": []
            },
            {
              "name": "auth-001",
              "properties": {
                "scope": "alert",
                "allow_configure": true,
                "allow_post": true
              },
              "children": [],
              "list": []
            }
          ]
        },
        {
          "name": "preferences",
          "properties": {
            "locale": "C",
            "login_screen": "configuration/preferences",
            "session_timeout": 15,
            "cli_idle_timeout": "infinite",
            "advanced_analytics": false
          },
          "children": [
            {
              "name": "keys",
              "properties": {},
              "children": [],
              "list": []
            },
            {
              "name": "tokens",
              "properties": {},
              "children": [],
              "list": []
            }
          ],
          "list": []
        }
      ],
      "list": [],
      "href": "/api/user/v1/users/firstlast"
    },
    {
...
    }
  ]
}
```

Query Parameter: match

The `match`*`property-name=value`* query parameter returns a list of resources that have the specified property name at the specified value.

The following example returns the list of users for which the value of the `kiosk_mode` property is `true`:

```
match_kiosk_mode=true
```

The following example returns the list of users for which the value of the `roles` property contains `super` and the value of the `require_annotation` property is `true`:

```
match_roles='*super*'&match_require_annotation=true
```

> **Note:**
>
> The `match_`*`property-name=value`* query parameter is not supported for listing logs using `/api/log/v{1|2}`, or for listing pools, projects, filesystems, or LUNs using `/api/storage/v{1|2}`.

2

Working with the RESTful API

The access service is the entry point for all RESTful API services on Oracle ZFS Storage Appliance. The service is used to authenticate user credentials and to list the available RESTful API services, including their versions and access points.

Accessing the Service

To access the service, use one of the following URLs:

- http://*hostname*:215/api/access/v2
- http://*hostname*:215/api/access/v1

To access other services, log in using the access service to get the location and versions of the available services and then use the returned URI to access those services. Service locations can change based on the current appliance configuration or release level.

Table 2-1 Access Service Commands

Request	Path	Description
GET	/api/access/v{1|2}	List RESTful API service access points
POST	/api/access/v{1|2}	Create a non-persistent login token
DELETE	/api/access/v{1|2}	Log out and delete a non-persistent login token

List Services

The list services command lists the available service access URIs. If a login session is not desired, list services can be used with appropriate credentials to list the available service access URIs. This command lists all the RESTful API services and versions available on Oracle ZFS Storage Appliance.

Example Request:

```
GET /api/access/v1 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: admin1
X-Auth-Key: password
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 190
X-Zfssa-Access-Api: 1.0

{
  "services": [
    {
      "version": "1.1",
      "name": "hardware",
      "uri": "https://hostname:215/api/hardware/v1"
    },
    {
      "version": "1.0",
      "name": "san",
      "uri": "https://hostname:215/api/san/v1"
    },
    {
      "version": "1.0",
      "name": "network",
      "uri": "https://hostname:215/api/network/v1"
    }
  ]
}
```

List Service Operations

This command returns the list of operations (methods) that are available for the specified service. If applicable, this command returns information about the resources of the specified service. In the following example, the cluster hardware component has resources that can be further examined and configured.

Example Request. Notice that this request uses an authentication token. See Authentication Tokens:

```
GET /api/hardware/v1 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: puPnHZKgSrUmXqYzOwFCrGcLOGwPODj
```

Example Result. For brevity, most of this output is omitted:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 204
X-Zfssa-Access-Api: 1.0

{
  "service": {
    "methods": [
      {
        "path": "",
        "href": "/api/hardware/v1",
        "request": "GET",
        "description": "List the hardware service commands."
      },
      {
        "path": "/cluster",
        "href": "/api/hardware/v1/cluster",
        "request": "GET",
        "description": "Get cluster properties and cluster resource list"
      },
      {
        "path": "/cluster/resources/<resource:path>",
        "href": "/api/hardware/v1/cluster/resources/<resource:path>",
        "request": "GET",
        "description": "Get properties for the specified cluster resource"
      },
      {
        "path": "/cluster/resources/<resource:path>",
        "href": "/api/hardware/v1/cluster/resources/<resource:path>",
        "request": "PUT",
        "description": "Modify the specified cluster resource"
      },
      {
        "path": "/chassis",
        "href": "/api/hardware/v1/chassis",
        "request": "GET",
        "description": "List hardware chassis"
      }
    ],
    "version": "1.1",
    "name": "hardware",
    "uri": "https://hostname:215/api/hardware/v1"
  }
}
```

Authentication Tokens

A non-persistent login token is obtained from the access service by sending a `POST` request. This non-persistent login token can be used by all other services as an identity credential. The non-persistent login token is invalidated after a timeout period set by the user's session timeout property. The default is usually 15 minutes. A `DELETE` request can be used to logout and invalidate the non-persistent login token.

This non-persistent login token is equivalent to the previous authentication session ID. It is supported in both RESTful API version 2 and RESTful API version 1. It is specific to the cluster node on which the ID was created and is not synchronized between the cluster peers.

A user can also create persistent tokens to access RESTful API. Creating persistent tokens is only supported in RESTful API version 2 and later. Persistent tokens are synchronized between the cluster peers and thus can be created on one cluster node and used to communicate with the other cluster node. See RESTful API User Service.

Create Non-persistent Login Token

A `POST` request requests a new non-persistent login token. On success, an HTTP status of 201 is returned along with a JSON object that has a single property, `access`, that contains a list of available RESTful API services. An optional property, `name`, is available to set the token’s name.

Example Create Request:

```
POST /api/access/v2 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password-xxx
```

A successful request returns HTTP status 201 (Created), as well as a non-persistent login token through the X-Auth-Session HTTP header. The response body contains a list of services accessible via this login.

Response Header:

```
HTTP/1.1 201 Created
X-Auth-Session: puPnHZKgSrUmXqYzOwFCrGcLOGwPODj
X-Auth-Name: REST-YG02oRod
Content-Type: application/json
```

```
Content-Length: 378
X-Zfssa-Access-Api: 1.0

{
    "access": {
        "services":[{
            ...
        }]
    }
}
```

Logout and Delete Non-persistent Login Token

An empty `DELETE` sends a request to log out and invalidate the non-persistent login token.

Example Logout Request:

```
DELETE /api/access/v2 HTTP/1.1
X-Auth-Session: puPnHZKgSrUmXqYzOwFCrGcLOGwPODj
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Access-Api: 1.0
```

3

Managing Certificates

The RESTful API enables you to manage certificate signing requests (CSRs), system or trusted user certificates, and certificate authority (CA) certificates.

In the following table, *request* is the `uuid` of a CSR, and *certificate* is the `uuid` of a system or trusted user certificate or a CA certificate.

Table 3-1 Certificate Operations

Request	Append to Path /api/setting/v{1|2}	Description
GET	/certificates/system/template	Return a template CSR.
POST	/certificates/system	Create a CSR. Upload a new system certificate.
GET	/certificates/system/*request*	List the properties of the specified CSR. Return the CSR in PEM format.
GET	/certificates/system	List the properties of all system certificates and requests.
PUT	/certificates/system	Set the value of the default system certificate.
GET	/certificates/system/*certificate*	List the properties of the specified system certificate. Return the certificate in PEM format.
DELETE	/certificates/system/*certificate*	Destroy the specified system certificate.
GET	/certificates/trusted	List the properties of all trusted certificates.
POST	/certificates/trusted	Upload a new trusted certificate.
GET	/certificates/trusted/*certificate*	List the properties of the specified trusted certificate. Return the certificate in PEM format.
PUT	/certificates/trusted/*certificate*	Set the value of the `services` property of the specified trusted certificate.
DELETE	/certificates/trusted/*certificate*	Destroy the specified trusted certificate.

List Certificates

The following request lists the properties of all system certificates on the host and lists the value of the `default` property.

Example Request:

```
GET /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

The first certificate in the following example is an automatically-generated conventional certificate (based on the domain or IP address). The second certificate is an automatically-generated certificate based on the Appliance Serial Number (ASN) UUID.

At the end of this result, the value of the `default` property shows that the system default certificate is selected automatically.

```
HTTP/1.1 200 OK
Date: Sat, 08 May 2021 00:37:21 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Content-Length: 1975

{
    "certificates": [
        {
            "uuid": "system-cert1-uuid",
            "type": "cert",
            "data": {
                "subject": [
                    {
                        "countryName": "US"
                    },
                    {
                        "stateOrProvinceName": "CA"
                    },
                    {
                        "localityName": "Exampleton"
                    },
                    {
                        "organizationName": "Example Corp, Inc"
                    },
                    {
                        "commonName": "alice.example.com"
                    }
                ],
                "issuer": [
                    {
                        "countryName": "US"
                    },
                    {
                        "stateOrProvinceName": "AK"
                    },
                    {
                        "localityName": "Trustville"
                    },
                    {
                        "organizationName": "Totally Trustworthy Certificates,
Inc"
                    },
                    {
                        "commonName": "Most Trusted Certificate"
                    }
                ],
                "serialNumber": "64",
                "validity": {
                    "notBefore": "20210520T21:08:27",
                    "notAfter": "20220520T21:08:27"
                },
                "extensions": {
                    "basicConstraints": {
                        "value": [
                            {
                                "CA": false
                            }
                        ]
                    },
                    "subjectKeyIdentifier": {
                        "value": "subjectKeyIdentifier"
                    },
                    "authorityKeyIdentifier": {
                        "value": [
                            {
                                "keyid": "authorityKeyIdentifier"
                            }
                        ]
                    },
                    "subjectAltName": {
                        "value": [
                            {
                                "DNS": "alice.example.com"
                            },
                            {
                                "IP": "alice.example.com-ipaddr"
                            }
                        ]
                    }
                }
            },
            "sha1fingerprint": "sha1fingerprint",
            "href": "/api/setting/v2/certificates/system/system-cert1-uuid"
        },
        {
            "uuid": "system-cert2-uuid",
            "type": "cert",
            "asn": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275",
            "data": {
                "subject": [
                    {
                        "commonName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
                    }
                ],
                "issuer": [
                    {
                        "commonName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
                    }
                ],
                "serialNumber": "59:8A:73:7B:00:00:00:07",
                "validity": {
                    "notBefore": "20060215T18:00:00",
                    "notAfter": "20380119T03:14:07"
                },
                "extensions": {
                    "nsComment": {
                        "value": "Automatically generated"
                    },
                    "subjectAltName": {
                        "critical": true,
                        "value": [
                            {
                                "DirName": "8bf7f9bc-8b3a-4064-e59f-bf09e3dba275"
                            }
                        ]
                    }
                }
            },
            "sha1fingerprint": "sha1fingerprint",
            "href": "/api/setting/v2/certificates/system/system-cert2-uuid"
        }
    ],
    "default": "auto"
}
```

The following request lists the properties of the specified *trusted-cert1-uuid* trusted certificate.

Example Request:

```
GET /api/setting/v2/certificates/trusted/trusted-cert1-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Date: Sat, 08 May 2021 00:37:57 GMT
Content-Length: 984
Content-Type: application/json; charset=utf-8
X-Zfssa-Setting-Api: 2.0
X-Zfssa-Api-Version: 2.0

{
    "certificate": {
        "uuid": "trusted-cert1-uuid",
        "type": "cert_ca",
        "data": {
            "subject": [
                {
                    "countryName": "US"
                },
                {
                    "stateOrProvinceName": "AK"
                },
                {
                    "localityName": "Trustville"
                },
                {
                    "organizationName": "Totally Trustworthy Certificates, Inc"
                },
                {
                    "commonName": "Most Trusted Certificate"
                }
            ],
            "issuer": [
                {
                    "countryName": "US"
                },
                {
                    "stateOrProvinceName": "AK"
                },
                {
                    "localityName": "Trustville"
                },
                {
                    "organizationName": "Totally Trustworthy Certificates, Inc"
                },
                {
                    "commonName": "Most Trusted Certificate"
                }
            ],
            "serialNumber": "83:F7:79:02:5F:44:4D:60",
            "validity": {
                "notBefore": "20210316T17:28:37",
                "notAfter": "20210415T17:28:37"
            },
            "extensions": {
                "subjectKeyIdentifier": {
                    "value": "subjectKeyIdentifier"
                },
                "authorityKeyIdentifier": {
                    "value": [
                        {
                            "keyid": "authorityKeyIdentifier"
                        }
                    ]
                },
                "basicConstraints": {
                    "value": [
                        {
                            "CA": true
                        }
                    ]
                }
            }
        },
        "sha1fingerprint": "sha1fingerprint",
        "services": [
            "ldap",
            "cloud"
        ],
        "href": "/api/setting/v2/certificates/trusted/trusted-cert1-uuid"
    }
}
```

Return a Certificate in PEM Format

To return a certificate in PEM format, specify one of the following values in the Accept header:

```
application/pkix-cert
application/x-x509-ca-cert
application/x-x509-user-cert
application/x-pem-file
```

Example Request:

```
GET /api/setting/v2/certificates/system/system-cert1-uuid HTTP/1.1
Host: alice.example.com:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/x-pem-file
```

Example Result:

```
HTTP/1.1 200 OK
Date: Thu, 13 May 2021 06:29:33 GMT
Content-Length: 1440
Content-Type: application/x-pem-file; charset=utf-8
X-Zfssa-Setting-Api: 2.0
X-Zfssa-Api-Version: 2.0

-----BEGIN CERTIFICATE-----
MIID+TCCAuGgAwIBAgIIXKTieQAAAAIwDQYJKoZIhvcNAQELBQAwVjEgMB4GA1UE
...
sUSSZgilvMJ4G8jtx6JSbG4DzDkvo8vq7GSika7h+hi5cbDiZdsOL9kDtBIqSAVN
Z1gjaFpzgio6wRvaIA==
-----END CERTIFICATE-----
```

Upload a Key or Certificate

When you receive the signed certificate from the CA, use the following command to upload the certificate. Specify one of the following values in the Content-Type header:

```
application/pkix-cert
application/x-x509-ca-cert
application/x-x509-user-cert
application/x-pem-file
```

Example Request:

```
POST /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/x-pem-file

-----BEGIN CERTIFICATE-----
MIIDdTCCAl2gAwIBAgICH5cwDQYJKoZIhvcNAQELBQAwazELMAkGA1UEBhMCdXMx
...
cgfvd1NUEvSdlb2+cjRBd9uSdtLfv7H5BKTKEdOXikv9+f150MytMEo4ABt0pEyp
/KwtRsdIxmzAjmNqfQPR6eAHVfQ/CGwh6Q==
-----END CERTIFICATE-----
```

Example Result:

```
HTTP/1.1 200 OK
Date: Tue, 11 May 2021 18:04:22 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Location: /api/setting/v2/certificates/system/system-cert3-uuid
Content-Length: 987

{
    "certificate": {
        "uuid": "system-cert3-uuid",
        "type": "cert",
        "data": {
            "subject": [
                {
```

```
                "countryName": "US"
            },
            {
                "stateOrProvinceName": "CA"
            },
            {
                "localityName": "Exampleton"
            },
            {
                "organizationName": "Example Corp, Inc"
            },
            {
                "commonName": "alice.example.com"
            }
        ],
        "issuer": [
            {
                "countryName": "US"
            },
            {
                "stateOrProvinceName": "AK"
            },
            {
                "localityName": "Trustville"
            },
            {
                "organizationName": "Totally Trustworthy Certificates, Inc"
            },
            {
                "commonName": "Most Trusted Certificate"
            }
        ],
        "serialNumber": "64",
        "validity": {
            "notBefore": "20210520T21:08:27",
            "notAfter": "20220520T21:08:27"
        },
        "extensions": {
            "basicConstraints": {
                "value": [
                    {
                        "CA": false
                    }
                ]
            },
            "subjectKeyIdentifier": {
                "value": "subjectKeyIdentifier"
            },
            "authorityKeyIdentifier": {
                "value": [
                    {
                        "keyid": "authorityKeyIdentifier"
                    }
                ]
            },
            "subjectAltName": {
                "value": [
                    {
                        "DNS": "alice.example.com"
                    },
                    {
```

```
                                    "IP": "alice.example.com-ipaddr"
                                }
                            ]
                        }
                    }
                },
                "sha1fingerprint": "sha1fingerprint",
                "href": "/api/setting/v2/certificates/system/system-cert3-uuid"
            }
        }
```

Specify the Services for Which a Certificate Should Be Trusted

You cannot modify any properties of a system certificate or a CSR. Set the properties of a CSR before you post the CSR.

You can set the value of the `services` property of a trusted certificate. The `services` property is the list of services for which the certificate should be trusted.

The following example sets the `services` property of a trusted certificate.

```
PUT /api/setting/v2/certificates/trusted/trusted-cert2-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Content-Type: application/json

{"certificate": { "services": ["ldap"] }}
```

The following example sets multiple services for which the certificate should be trusted.

```
PUT /api/setting/v2/certificates/trusted/trusted-cert2-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Content-Type: application/json

{"certificate": {"services": [ "ldap", "cloud" ] }}
```

Set the System Default Certificate

The following example assigns a default system certificate.

```
PUT /api/setting/v2/certificates/system HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: alice.example.com:215
Accept: application/json
Content-Type: application/json

{ "default": "system-cert1-uuid" }
```

Destroy a Certificate

The `DELETE` command destroys the specified certificate, request, or key. If successful, HTTP status 204 (No Content) is returned.

```
DELETE /api/setting/v2/certificates/system/system-cert2-uuid HTTP/1.1
```

Enable HTTP Strict Transport Security

HTTP Strict Transport Security (HSTS) allows only secure HTTPS connections, and not HTTP connections, for a specified period of time. Before using HSTS, familiarize yourself with HSTS prerequisites, understand browser behavior with HSTS enabled, and install a certificate signed by a certificate authority.

> **Note:**
>
> Failure to keep the certificate valid and appropriate could negate HSTS security advantages or could cause a browser to not connect with Oracle ZFS Storage Appliance.

As shown in the following example, the default maximum length of time that HSTS will remain enabled is 63072000 seconds, or 2 years.

Example Request:

```
GET /api/setting/v2/security HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Date: Fri, 14 May 2021 19:22:27 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Content-Length: 109

{
  "Security settings": {
    "href": "/api/setting/v2/security",
    "hsts_enable": false,
    "hsts_max_age": 63072000
  }
}
```

To enable HSTS for this appliance, set the `hsts_enable` property to `true`.

```
PUT /api/setting/v2/security HTTP/1.1
Host: alice.example.com:215
Content-Type: application/json

{"hsts_enable": true}
```

Create a Server Certificate

The first step to create a server certificate is to create a certificate signing request (CSR). Post the CSR on Oracle ZFS Storage Appliance and send it to your CA. After you receive the signed certificate from the CA, upload that signed certificate as described in Upload a Key or Certificate. The signed certificate replaces the request.

Return a Request Template

The `template` command returns a framework for a CSR, including default values for minimum required properties.

Example Request:

```
GET /api/setting/v2/certificates/system/template HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Date: Thu, 13 May 2021 08:03:03 GMT
Content-Length: 261
Content-Type: application/json; charset=utf-8
X-Zfssa-Setting-Api: 2.0
X-Zfssa-Api-Version: 2.0

{
    "request": {
        "type": "request",
        "data": {
            "subject": [
                {
                    "commonName": "alice.example.com"
                }
            ],
            "extensions": {
                "subjectAltName": {
                    "value": [
                        {
                            "IP": "alice.example.com-ipaddr"
                        },
                        {
                            "DNS": "alice.example.com"
                        }
                    ]
                }
            }
        },
        "href": "/api/setting/v2/certificates/system/template"
    }
}
```

Populate and Upload the Request

If you use this `template` output, only include the `data` element.

For additional properties that you might want to specify in your CSR, list the properties of an existing system certificate, as shown in List Certificates.

When you are satisfied with your CSR, upload the CSR to the host, as shown in the following example. Once you have uploaded the CSR, you can no longer change it.

Example Request:

```
POST /api/setting/v2/certificates/system HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-type: application/json

{
    "data": {
        "subject": [
            {
                "commonName": "alice.example.com"
            },
            {
                "organizationName": "Example Corp, Inc"
            },
            {
                "localityName": "Exampleton"
            },
            {
                "stateOrProvinceName": "CA"
            },
            {
                "countryName": "US"
            }
        ],
        "extensions": {
            "subjectAltName": {
                "value": [
                    {
                        "DNS": "alice.example.com"
                    },
                    {
                        "IP": "alice.example.com-ipaddr"
                    }
                ]
            }
        }
    }
}
```

Example Result:

```
HTTP/1.1 201 Created
Date: Fri, 14 May 2021 01:17:45 GMT
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Location: /api/setting/v2/certificates/system/65119889-98d3-4fc4-bff5-f007a55f6cb3
Content-Length: 379

{
    "request": {
        "uuid": "csr-uuid",
        "type": "request",
        "data": {
            "subject": [
                {
                    "commonName": "alice.example.com"
                },
                {
                    "organizationName": "Example Corp, Inc"
                },
                {
                    "localityName": "Exampleton"
                },
                {
                    "stateOrProvinceName": "CA"
                },
                {
                    "countryName": "US"
                }
            ],
            "extensions": {
                "subjectAltName": {
                    "value": [
                        {
                            "DNS": "alice.example.com"
                        },
                        {
                            "IP": "alice.example.com-ipaddr"
                        }
                    ]
                }
            }
        },
        "href": "/api/setting/v2/certificates/system/csr-uuid"
    }
}
```

Transfer the Request to the CA

The uploaded CSR has a UUID that you can use to display the properties or retrieve the request in PEM format.

To return the CSR in PEM format, specify one of the following values in the Accept header:

```
application/pkcs10
application/x-pem-file
```

Example Request:

```
GET /api/setting/v2/certificates/system/csr-uuid HTTP/1.1
Host: alice.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/x-pem-file
```

Example Result:

```
HTTP/1.1 200 OK
Date: Fri, 14 May 2021 03:47:21 GMT
Content-Type: application/x-pem-file; charset=utf-8
X-Zfssa-Api-Version: 2.0
X-Zfssa-Setting-Api: 2.0
Content-Length: 997

-----BEGIN CERTIFICATE REQUEST-----
MIICpjCCAY4CAQAwJDEiMCAGA1UEAwwZYXJkb2NoLWt6LTIudWsub3JhY2xlLmNv
...
Bc0Q9FVRVv89AkmeAlF7727aIqmgmFcIUIIrEKTG4PSacedaoBsbjpvrizCuMhyo
vgUkOPE/0xLAfw==
-----END CERTIFICATE REQUEST-----
```

Transfer the CSR to your CA in the prescribed way. When you receive the signed certificate from the CA, upload the signed certificate as shown in Upload a Key or Certificate.

4 RESTful API Alert Service

Important Oracle ZFS Storage Appliance events, such as hardware and software faults, trigger alerts. Alerts appear in the logs and can also be configured to perform other alert actions such as send an email or resume a dataset.

The RESTful API alert service enables you to create custom alert actions (responses to event alerts) and custom Analytics statistics threshold alerts.

Alert Service Commands

The following table shows the alert service commands. More than one alert action (response) can be defined for a single alert event. For example, you can send email to two different groups, write a syslog message, and resume a dataset all in response to a single alert. An *actions-###* object is the set of alert events and all responses to the alert. An *action-###* object is one of the responses.

Table 4-1   Alert Service Commands

Request	Append to Path /api/alert/v{1|2}	Description
GET	Use only /api/alert/v{1|2}	List the alert service commands
GET	/actions	List all alert actions objects
POST	/actions	Create a new alert action
GET	/actions/*actions-###*	List the specified alert actions properties
PUT	/actions/*actions-###*	Modify the specified alert actions object
DELETE	/actions/*actions-###*	Destroy the specified actions object
POST	/actions/*actions-###*	Create a new alert actions action
GET	/actions/*actions-###*/*action-###*	List the specified alert actions action properties
PUT	/actions/*actions-###*/*action-###*	Modify the specified alert actions action object
POST	/postalert	Posts a custom alert
DELETE	/actions/*actions-###*/*action-###*	Destroy the specified alert actions action object
GET	/thresholds	List all threshold alert objects
POST	/thresholds	Create a new threshold alert
GET	/thresholds/*threshold-alert-uuid*	List the specified threshold alert properties
PUT	/thresholds/*threshold-alert-uuid*	Modify the specified threshold alert object
DELETE	/thresholds/*threshold-alert-uuid*	Destroy the specified threshold alert object
GET	/events	Listen for new alert events

Alert Actions

An alert action is a response to an event alert. To create an alert action, specify one or more events, and specify one or more actions to take when an alert is sent for that event such as send an email or execute a workflow. More than one alert action can be specified for any particular event alert.

The `category` property specifies the category of event for which the alert action will be performed. Each category includes one or more events. The events within each category are listed when you create or list the alert action. By default, the alert action will be performed for all events in the category. If the alert action should be performed for only a subset of events, change to `false` the values of properties that represent events that should not cause the alert action to be performed.

The following table describes the event categories that you can specify.

Table 4-2 Alert Action Event Categories

Category	Description
`ad`	Active Directory or SMB Kerberos client authentication degraded
`all`	High-level events such as all alerts or defects, service alerts, and hardware faults
`analytics`	High-level events such as datasets auto-suspend warning, memory total exceeded, and usage exceeded. For events for specific Analytics statistics, see Threshold Alerts.
`appliance_software`	Events that prevent software update or that result in kernel panic
`cloud_snapshot`	
`cluster`	Cluster events, including link failures and peer errors
`custom`	An alert action for a custom event, which is specified in a workflow. See Custom Alerts.
`hardware`	Appliance boot and hardware configuration changes
`hardware_faults`	Any hardware fault

`ndmp`
`backup`
`restore`
	NDMP TAR/DUMP backup and restore start and finish events

`network`	Network port, datalink, and IP interface events and failures
`scrk`	Support bundle upload events

`replication`
`replication_source`
`replication_target`
	Send and receive events and failures

`smf`	Software services failure events
`shadow`	Migration errors or migration complete

Table 4-2 (Cont.) Alert Action Event Categories

Category	Description
`thresholds`	Enables you to add an action to an existing threshold event alert as described in Threshold Alerts
`zfs_pool`	Storage pool events, including scrub and hot space activation

The `handler` property specifies the type of action you want to take when the specified event occurs. Most values of `handler` require additional properties to be set, as shown in the following table.

Table 4-3 Alert Action Response Types

Response Type (`handler`)	Handler Properties	Response Type Description
`email`	`address` `subject`	Sends an email with the specified subject to the specified recipients. To send to multiple individual recipients, separate email addresses with a comma and a space on one line. Use the SMTP service to configure how email is sent. See List Services.
`snmp_trap`	None	Sends an SNMP trap that contains alert details. Use the SNMP service to configure an SNMP trap destination. See List Services.
`syslog`	None	Sends a system message that contains alert details to one or more remote systems. Use the `syslog` service to configure syslog destinations. See List Services. For more information about sending syslog messages, see Syslog Configuration in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.
`resume_dataset`	`dataset`	Resumes an Analytics dataset. Resuming and suspending datasets can be useful for diagnosing intermittent performance issues and for other cases when keeping a dataset continuously enabled is not desirable. For more information, see About Analytics Datasets in *Oracle ZFS Storage Appliance Analytics Guide, Release OS8.8.x*.
`suspend_dataset`	`dataset`	Suspends an Analytics dataset.
`resume_worksheet`	`worksheet`	Resumes an Analytics worksheet. Resuming and suspending worksheets can be useful for the same reasons as resuming and suspending datasets. A worksheet might contain numerous datasets. For more information, see Worksheet Graphs and Plots in *Oracle ZFS Storage Appliance Analytics Guide, Release OS8.8.x*.

Table 4-3 (Cont.) Alert Action Response Types

Response Type (`handler`)	Handler Properties	Response Type Description
`suspend_worksheet`	`worksheet`	Suspends an Analytics worksheet.
`execute_workflow`	`workflow`	Executes the specified workflow. For more information about workflows that are eligible to be alert actions, see Using Workflows for Alert Actions in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

List All Alert Actions

When you list all alert actions, only the event category and each event in that category are listed for each alert action. To also show the responses that are defined for an alert action, see List a Single Alert Action.

Example Request to Get Alert Actions:

```
GET /api/alert/v1/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 1395

{
    "actions": [
        {
            "category": "smf",
            "uuid": "actions-uuid1",
            "failed_services": true,
            "degraded_services": true,
            "repaired_services": false,
            "actions": "actions-000",
            "href": "/api/alert/v2/actions/actions-000"
        },
        {
            "category": "analytics",
            "uuid": "actions-uuid2",
            "analytics_datasets_auto-suspend_notify": false,
            "analytics_datasets_auto-suspend_warning": false,
            "analytics_memory_total_exceeded": true,
            "analytics_memory_total_normal": false,
            "analytics_usage_exceeded": true,
            "analytics_usage_normal": false,
            "actions": "actions-001",
            "href": "/api/alert/v2/actions/actions-001"
        }
    ]
}
```

List a Single Alert Action

When you specify a particular alert action to list, the event category, each event in that category, and each response or action are listed for that alert action.

The following alert action has three responses that all will be performed when one of the `true` events occurs.

Example Request:

```
GET /api/alert/v1/actions/actions-000 HTTP/1.1
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 331

{
    "actions": {
        "href": "/api/alert/v2/actions/actions-000",
        "category": "smf",
        "uuid": "actions-uuid1",
        "failed_services": true,
        "degraded_services": true,
        "repaired_services": false,
        "action-000": {
            "handler": "email",
            "address": "admin@example.com",
            "subject": "failed or degraded service",
            "href": "/api/alert/v2/actions/actions-000/action-000"
        },
        "action-001": {
            "handler": "email",
            "address": "it-team@example.com",
            "subject": "failed or degraded service",
            "href": "/api/alert/v2/actions/actions-000/action-001"
        },
        "action-002": {
            "handler": "syslog",
            "href": "/api/alert/v2/actions/actions-000/action-002"
        }
    }
}
```

Create an Alert Action

When you create an alert action, you must specify a value for the `category` property, which is the category of event for which you are defining this custom response. See table "Alert Action Event Categories" in Alert Actions for the list of `category` values.

Example Request:

```
POST /api/alert/v1/actions HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 30
```

```
{"category": "smf"}
```

Example Result:

The result lists all the events in the specified event category. By default, all events in the category will cause the response actions to be performed (they are all set to `true`).

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 118
Location: /api/alert/v1/actions/actions-000

{
    "actions": {
        "href": "/api/alert/v2/actions/actions-000",
        "category": "smf",
        "uuid": "actions-uuid",
        "failed_services": true,
        "degraded_services": true,
        "repaired_services": true
    }
}
```

Modify an Alert Action

If some of the events in the specified event category should not cause the response actions to be performed, set to `false` the properties that represent those events.

In the following example, you might want to define a different response for a repaired service than for a failed or degraded service.

Example Request:

```
PUT /api/alert/v1/actions/actions-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 30

{"repaired_services": false}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 195

{
    "actions": {
        "href": "/api/alert/v2/actions/actions-000",
        "category": "smf",
        "uuid": "actions-uuid",
        "failed_services": true,
        "degraded_services": true,
        "repaired_services": false
    }
}
```

Specify a Response for an Event

By default, event alerts are logged to the alerts log. The preceding examples specified the events for which you want to define a response in addition to posting to the alerts log. To define the response to those events, specify the value of the `handler` property for the particular alert action. See table "Alert Action Response Types" in Alert Actions for the list of `handler` values.

Example Request:

This example creates an alert action of send an email to `admin` for the `actions-000` alert.

```
POST /api/alert/v1/actions/actions-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 68

{"handler": "email", "address": "admin@example.com", "subject": "failed or degraded
service"}
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 177
Location: /api/alert/v1/actions/actions-000/action-000

{
    "action": {
        "href": "/api/alert/v1/actions/actions-002/action-000",
        "handler": "email",
        "address": "admin@example.com",
        "subject": "failed or degraded service"
    }
}
```

To specify additional responses for the same event, issue the `POST` request again for the same alert action and specify a different handler, or specify the same handler and a different argument for the handler. If you specify more than one `handler` property in one request, all `handler` properties except for the last one are ignored.

The following examples are shortened. These requests create `/api/alert/v1/actions/actions-000/action-001` and `/api/alert/v1/actions/actions-000/action-002` as shown in List a Single Alert Action.

```
POST /api/alert/v1/actions/actions-002 HTTP/1.1
...
{"handler": "email", "address": "it-team@example.com", "subject": "failed or degraded
service"}

POST /api/alert/v1/actions/actions-002 HTTP/1.1
...
{"handler": "syslog"}
```

Modify a Response for an Event

To modify a response, specify the HREF for the response that you want to modify.

```
PUT /api/alert/v1/actions/actions-000/action-001 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv
Content-Type: application/json
Content-Length: 28

{"address": "it-group@example.com"}
```

Delete a Response for an Event

To delete a response, specify the HREF for the response that you want to delete.

```
DELETE /api/alert/v1/actions/actions-000/action-000 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-Session: uerqghq84vbdv

HTTP/1.1 204 No Content
```

Delete an Alert Action

To delete an alert action, specify the HREF for the alert action that you want to delete.

Example Request:

```
DELETE /api/alert/v1/actions/actions-003 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-Api: 1.0
```

Custom Alerts

A custom alert is an alert action for a custom event. Specify `custom` for the value of `category`.

The custom event is defined in a workflow. Instead of selecting from predefined events, specify the following properties the describe the event defined in the workflow.

Table 4-4 Custom Alert Properties

Property	Type	Description
`severity`	String	Optional. The severity of the event that precipitated the alert. Valid values are: Minor, Major, or Critical.
`description`	String	Required. A description of the event that precipitated the alert.
`response`	String	Optional. A description of the actions that will be performed by the system to mitigate the effects of this event.

Table 4-4 (Cont.) Custom Alert Properties

Property	Type	Description
`impact`	String	Optional. A description of the effect that this event has on the appliance.
`recommended_action`	String	Optional. A description of the actions that the administrator should take to mitigate the effects of this event.

Configure Authorizations to Create and Post Custom Alerts

To create custom alerts, the user must have the `allow_configure` authorization in the `alert` scope. To post custom alerts, the user must have the `allow_post` authorization in the `alert` scope. See table "Authorizations Required to Use Custom Alerts" in section Creating and Posting Custom Alerts from Within a Workflow in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Create a Custom Alert

Specify `custom` for the value of `category`, and set a value for at least the `description` property from the properties table in Custom Alerts.

Example Request:

```
POST /api/alert/v1/actions
Accept: application/json
Content-Type: application/json

{"category": "custom", "severity": "Minor", "description": "Custom alert description",
"response": "What the system will do", "impact": "What happened to the appliance",
"recommended_action": "What the administrator should do"}
```

Example Result:

Note the `uuid` of the alert in the output. You will need this to post the alert.

```
{
    "actions": {
        "href": "/api/alert/v1/actions/actions-004",
        "category": "custom",
        "severity": "Minor",
        "description": "Custom alert description",
        "response": "What the system will do",
        "impact": "What happened to the appliance",
        "recommended_action": "What the administrator should do",
        "uuid": "custom-alert-uuid"
    }
}
```

Modify this custom alert the same way you modify any other alert, as described in Modify an Alert Action, adding or changing the values of the `severity`, `description`, `response`, `impact`, or `recommended_action` properties.

Specify a response to this custom alert the same way you specify a response to any other alert, as described in Specify a Response for an Event and Modify a Response for an Event.

Delete an alert response or a custom alert in the same way as for any other alert, as described in Delete a Response for an Event and Delete an Alert Action.

Post a Custom Alert

You must provide the UUID of the custom alert to post. The UUID is shown when you create the alert and when you list the alert. See Create a Custom Alert and List a Single Alert Action.

You can only post an alert that has a `category` value of `custom`.

In addition to the UUID of the alert, you must specify any of the properties that are not specified in the alert and are listed in the properties table in Custom Alerts. Properties that are specified in the alert can be given new values when the alert is posted.

Example Request:

```
POST /api/alert/v1/postalert
Accept: application/json
Content-Type: application/json

{"uuid": "custom-alert-uuid"}
```

Example Result:

```
{
    "uuid": "posted-alert-uuid"
}
```

Threshold Alerts

A threshold alert is a custom alert in which a threshold is defined for a particular Analytics statistic, and the alert action is executed when the statistic value is outside that threshold. See also .

The following table describes the properties to set to specify the Analytics statistic, define the threshold for that statistic, and define when alert actions will be executed for that threshold alert.

Table 4-5 Threshold Alert Properties

Property	Type	Description
`statname`	AnalyticsStatistics	Required. The Analytics statistic to monitor.
`limit`	PositiveInteger	Required. The threshold value that triggers the alert. The percent or the number of bytes, operations, accesses, or requests per second.
`type`	ChooseOne	How to compare the threshold value (`limit`) to the current statistic value. • `normal` – The current statistic value exceeds the threshold value. This is the default. • `inverted` – The current statistic value falls below the threshold value.

Table 4-5 (Cont.) Threshold Alert Properties

Property	Type	Description
`minpost`	Duration	Length of time in seconds that the statistic value must remain in the threshold condition before the alert action is executed. Default value is five minutes.
`days`	ChooseOne	Which days to send these alerts: `all` days, `weekdays`, or `weekends`. Default value is `all`.
`window_start` `window_end`	TimeOfDay	The window of time during which to execute this alert action. Specify times from 00:00 through 23:30 UTC on 30-minute increments. To execute this alert action any time the conditions are met, specify `none` as either the start time or the end time. Default values are `none` for `window_start` and `00:00` for `window_end`.
`frequency`	Duration	Length of time in seconds between re-executing the alert action while the statistic value remains in the threshold condition. Default value is five minutes.
`minclear`	Duration	Length of time in seconds that the statistic value must remain outside the threshold condition before a followup alert action is executed. Default value is five minutes.

List Threshold Alerts

The following request lists all of the configured threshold alerts.

Example Request:

```
GET /api/alert/v1/thresholds HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Date: Tue, 15 Oct 2019 10:38:40 GMT
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 689

{
    "thresholds": [
        {
            "uuid": "threshold-uuid1",
            "statname": "cpu.utilization",
            "type": "normal",
            "limit": 80,
            "minpost": 300,
            "days": "weekdays",
            "window_start": "08:00",
            "window_end": "19:30",
            "frequency": 300,
            "minclear": 300,
            "threshold": "threshold-000",
            "href": "/api/alert/v1/thresholds/threshold-uuid1"
        },
        {
            "uuid": "threshold-uuid2",
            "statname": "cap.meta_percentused[pool]",
            "type": "normal",
            "limit": 85,
            "minpost": 300,
            "days": "all",
            "window_start": "none",
            "window_end": "00:00",
            "frequency": 0,
            "minclear": 0,
            "threshold": "threshold-001",
            "href": "/api/alert/v1/thresholds/threshold-uuid2"
        }
    ]
}
```

Use the following request to list all properties for only the specified threshold alert.

```
GET /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
```

Create a Threshold Alert

This example creates a threshold alert for the event that datalink bytes per second exceeds 100000 KB. All other properties have default values.

To create a custom threshold alert, the user must have the `allow_configure` authorization in the `alert` scope.

Example Request:

```
POST /api/alert/v1/thresholds HTTP/1.1
Host: zfs-storage.example.com
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 50

{"statname": "datalink.kilobytes", "limit": 100000}
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 321
Location: /api/alert/v1/thresholds/threshold-uuid

{
    "threshold": {
        "href": "/api/alert/v1/thresholds/threshold-uuid",
        "uuid": "threshold-uuid",
        "statname": "datalink.kilobytes",
        "type": "normal",
        "limit": 100000,
        "minpost": 300,
        "days": "all",
        "window_start": "none",
        "window_end": "00:00",
        "frequency": 300,
        "minclear": 300
    }
}
```

Modify a Threshold Alert

Use this command to modify the properties of the specified threshold alert.

Example Request:

```
PUT /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215

{"days": "weekdays", "minpost": 120}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 326

{
    "threshold": {
        "href": "/api/alert/v1/thresholds/threshold-uuid",
        "uuid": "threshold-uuid",
        "statname": "datalink.kilobytes",
        "type": "normal",
        "limit": 100000,
        "minpost": 120,
        "days": "weekdays",
        "window_start": "none",
        "window_end": "00:00",
        "frequency": 300,
        "minclear": 300
    }
}
```

Delete a Threshold Alert

Delete the specified threshold alert.

Example Request:

```
DELETE /api/alert/v1/thresholds/threshold-uuid HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-Api: 1.0
```

5

Analytics Services

Analytics enables you to graph a variety of statistics in real time and record data for later retrieval. You can perform both long-term monitoring and short-term analysis. Analytics uses DTrace to dynamically create custom statistics that allow different layers of the operating system stack be analyzed in detail.

Analytics Commands

The following Analytics services are available at: `http://`*`hostname`*`/api/analytics/v{1|2}`.

Table 5-1 Analytics Commands

Request	Append to Path /analytics/v{1|2}	Description
GET	Use only /analytics/v{1|2}	List analytics service information
POST	/worksheets	Create a new analytics dataset
GET	/worksheets/*worksheet*	Get the specified analytics dataset properties
GET	/worksheets	List all analytics dataset objects
PUT	/worksheets/*worksheet*	Modify the specified analytics dataset object
DELETE	/worksheets/*worksheet*	Destroy the specified worksheet object
PUT	/worksheets/*worksheet*/suspend	Suspend all worksheet datasets
PUT	/worksheets/*worksheet*/resume	Resume all worksheet datasets
POST	/worksheets/*worksheet*/datasets	Create a new worksheet dataset
GET	/worksheets/*worksheet*/datasets/*dataset*	Get the specified worksheet dataset properties
GET	/worksheets/*worksheet*/datasets	List all worksheet dataset objects
PUT	/worksheets/*worksheet*/datasets/*dataset*	Modify the specified worksheet dataset object
DELETE	/worksheets/*worksheet*/datasets/*dataset*	Destroy the specified dataset object
POST	/datasets	Create a new analytics dataset
GET	/datasets/*dataset*	Get the specified analytics dataset properties
GET	/datasets	List all analytics dataset objects
PUT	/datasets/*dataset*	Modify the specified analytics dataset object
DELETE	/datasets/*dataset*	Destroy the specified dataset object
PUT	/datasets	Suspend or resume all datasets
PUT	/datasets/*dataset*/data	Save this dataset (if unsaved)

Table 5-1 (Cont.) Analytics Commands

Request	Append to Path /analytics/v{1|2}	Description
DELETE	/datasets/*dataset*/data	Remove data at the given [granularity] from this dataset
GET	/settings	List analytics settings
PUT	/settings	Modify analytics settings

Analytics Settings

The properties described in the following table enable you to collect all analytic data, set the number of hours of data to retain, and set a hostname lookup policy.

Property	Description
`retain_second_data`	Retention interval in hours for per-second data
`retain_minute_data`	Retention interval in hours for per-minute data
`retain_hour_data`	Retention interval in hours for per-hour data
`hostname_lookup`	Hostname lookup policy

Get Settings

This command gets the current values of analytics properties.

Example Request:

```
GET /api/analytics/v1/settings HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 131
X-Zfssa-Analytics-Api: 1.0

{
    "settings": {
        "href": "/api/analytics/v1/settings",
        "retain_hour_data": 600,
        "retain_minute_data": 400,
        "retain_second_data": 200,
        "hostname_lookup": true
    }
}
```

Modify Settings

The modify settings command is used to modify analytics settings, such as data retention values and the hostname lookup policy.

Example Request:

```
PUT /api/analytics/v1/settings HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 60

{"retain_hour_data":600, "retain_minute_data":400, "retain_second_data":200,
"hostname_lookup":true}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 101
X-Zfssa-Analytics-Api: 1.0

{
    "settings": {
        "href": "/api/analytics/v1/settings",
        "retain_hour_data": 600,
        "retain_minute_data": 400,
        "retain_second_data": 200,
        "hostname_lookup": true
    }
}
```

Analytics Worksheets

A worksheet is the BUI screen on which statistics are graphed. Multiple statistics can be plotted at the same time, and worksheets can be assigned a title and saved for future viewing. The act of saving a worksheet automatically executes the archive action on all open statistics, meaning that whatever statistics were open continue to be read and archived forever. The worksheet commands can be used to manage the worksheets available from the BUI.

The following table shows properties that are used in analytics worksheets.

Property	Description
`ctime`	Time and date when this worksheet was created
`mtime`	Time and date when this worksheet was last modified
`name`	Name of this worksheet
`owner`	Owner of this worksheet
`uuid`	Universal unique identifier for this worksheet

List Worksheets

Lists all currently configured analytics worksheets.

Example Request:

```
GET /api/analytics/v1/worksheets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-Api: 1.0

{
    "worksheets": [{
        "href": "/api/analytics/v1/worksheets/ab59bcbc...",
        "uuid": "ab59bcbc-080a-cf1a-98c9-9f485bc3a43d"
    }, {
        "href": "/api/analytics/v1/worksheets/bb3ee729...",
        "uuid": "bb3ee729-080a-cf1a-98c9-9f485bc3a43d"
    }]
}
```

Get Analytics Worksheet

Gets a single analytics worksheet.

Example Request:

```
GET /api/analytics/v1/worksheets/ab59bcbc-080a-cf1a-98c9-9f485bc3a43d
    HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-Api: 1.0

{
    "worksheet": {
        "ctime": "Thu Jun 13 2013 02:17:14 GMT+0000 (UTC)",
        "href": "/api/analytics/v1/worksheets
                 /ab59bcbc-080a-cf1a-98c9-9f485bc3a43d",
        "mtime": "Sun Jun 23 2013 16:22:01 GMT+0000 (UTC)",
        "name": "myworksheet",
        "owner": "root",
        "uuid": "ab59bcbc-080a-cf1a-98c9-9f485bc3a43d"
    }
}
```

Create Worksheets

Creates a new analytics worksheet.

Example Request:

```
POST /api/analytics/v1/worksheets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
```

```
Content-Length: 26

{"name": "myworksheet"}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Length: 280
Location: /api/analytics/v1/worksheets/bb3ee729-4480-4609-89b2-fae2dc016bec

{
    "worksheet": {
        "uuid": "bb3ee729-4480-4609-89b2-fae2dc016bec",
        "name": "myworksheet",
        "owner": "root",
        "ctime": "Fri Aug 23 2013 20:35:00 GMT+0000 (UTC)",
        "mtime": "Fri Aug 23 2013 20:35:00 GMT+0000 (UTC)",
        "href": "/api/analytics/v1/worksheets
                 /bb3ee729-4480-4609-89b2-fae2dc016bec"
    }
}
```

Rename Worksheets

Renames a saved worksheet.

Example Request:

```
PUT /api/analytics/v1/worksheets/a442e761-4048-4738-b95f-be0824d7ed09
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26

{"name": "test"}
```

Example Result:

```
HTTP/1.1 202 Accepted
Date: Tue, 20 Dec 2016 00:33:06 GMT
Server: TwistedWeb/192.0.2
Content-Length: 279
X-Zfssa-Analytics-Api: 1.1
X-Zfssa-Api-Version: 1.0
Content-Type: application/json; charset=utf-8

{
    "worksheet": {
        "href": "/api/analytics/v1/worksheets/a442e761-4048-4738-b95f-be0824d7ed09",
        "uuid": "a442e761-4048-4738-b95f-be0824d7ed09",
        "name": "test",
        "owner": "root",
        "ctime": "Wed Dec 14 2016 03:58:28 GMT+0000 (UTC)",
        "mtime": "Tue Dec 20 2016 00:25:57 GMT+0000 (UTC)"
    }
}
```

Destroy Worksheets

Destroys an analytics worksheet. In this example, the worksheet name is used as the worksheet identifier but the uuid identified in the href can also be used. The behavior of this command matches the behavior of the CLI command that destroys worksheets.

Example Request:

```
DELETE /api/analytics/v1/worksheets/name=myworksheet HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Analytics-Api: 1.0
```

List Worksheet Datasets

Lists all datasets in the specified worksheet.

The following table shows properties that are used in dataset configuration.

Property	Description
`name`	Name of the underlying statistic for this dataset
`drilldowns`	Drilldowns currently highlighted, if any
`seconds`	Number of seconds being displayed for this dataset

Example Request:

```
GET /api/analytics/v1/worksheets/name=myworksheet/datasets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Add Worksheet Dataset

Creates a worksheet dataset.

Example Request:

```
POST /api/analytics/v1/worksheets/name=myworksheet/datasets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26

{"name": "nfs4.ops", "seconds": 300}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
X-Zfssa-Analytics-Api: 1.0
Location: /api/analytics/v1/worksheets/name=me/datasets/nfs4.ops
Content-Length: 162
```

```
{
    "dataset": {
        "href": "/api/analytics/v1/worksheets/name=me/datasets/dataset-008",
        "name": "nfs4.ops",
        "width": 0,
        "drilldowns": [],
        "seconds": 300,
        "time": ""
    }
}
```

Modify Worksheet Dataset

Modifies an existing worksheet dataset.

Example Request:

```
PUT /api/analytics/v1/worksheets/name=myworksheet/datasets/dataset-008
    HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 26

{"seconds": 60}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 161
X-Zfssa-Analytics-Api: 1.0

{
    "dataset": {
        "href": "/api/analytics/v1/worksheets/name=me/datasets/dataset-008",
        "name": "nfs4.ops",
        "width": 0,
        "drilldowns": [],
        "seconds": 60,
        "time": ""
    }
}
```

Analytics Datasets

Analytics datasets use the following properties. All properties except for `suspended` are immutable.

Property	Description
`name`	Name of the underlying statistic for this dataset
`grouping`	Group to which this statistic belongs
`explanation`	Explanation of underlying statistic
`incore`	Bytes of dataset data in-core
`size`	Bytes of dataset data on-disk

Property	Description
`suspended`	Boolean indicating whether dataset is currently suspended
`activity`	Pending dataset activity flag

Available datasets:

- ad.avglatency
- ad.avglatency[op]
- ad.avglatency[result]
- ad.binds
- ad.binds[hostname]
- ad.binds[result]
- ad.ops
- ad.ops[op]
- ad.ops[result]
- arc.accesses[hit/miss]
- arc.l2_accesses[hit/miss]
- arc.l2_size
- arc.size
- arc.size[component]
- cpu.utilization
- cpu.utilization[mode]
- dnlc.accesses[hit/miss]
- fc.bytes
- fc.ops
- ftp.kilobytes
- http.reqs
- io.bytes
- io.bytes[op]
- io.disks[utilization=95][disk]
- io.ops
- io.ops[disk]
- io.ops[op]
- iscsi.bytes
- iscsi.ops
- metacap.bytesused
- metacap.percentused

- ndmp.diskkb
- nfs2.ops
- nfs2.ops[op]
- nfs3.ops
- nfs3.ops[op]
- nfs4.ops
- nfs4.ops[op]
- nfs4-1.ops
- nfs4-1.bytes
- nic.kilobytes
- nic.kilobytes[device]
- nic.kilobytes[direction]
- sftp.kilobytes
- smb.ops
- smb.ops[op]

List Datasets

Lists all configured analytic datasets.

Example Request:

```
GET /api/analytics/v1/datasets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-Api: 1.0

{
    "datasets": [{
        "dataset": "dataset-000",
        "href": "/api/analytics/v1/datasets/arc.accesses[hit/miss]",
        "name": "arc.accesses[hit/miss]"
    }, {
        "dataset": "dataset-001",
        "href": "/api/analytics/v1/datasets/arc.l2_accesses[hit/miss]",
        "name": "arc.l2_accesses[hit/miss]",
    }, {
        "dataset": "dataset-002",
        "href": "/api/analytics/v1/datasets/arc.l2_size",
        "name": "arc.l2_size",
    }, {
        "dataset": "dataset-003",
        "href": "/api/analytics/v1/datasets/arc.size",
        "name": "arc.size",
    }, {
        "dataset": "dataset-004",
        "href": "/api/analytics/v1/datasets/arc.size[component]",
        "name": "arc.size[component]",
    }, {
        ...
    }]
}
```

Get Dataset

Gets properties from the specified dataset.

Example Request:

```
GET /api/analytics/v1/datasets/nfs4.ops HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237
X-Zfssa-Analytics-Api: 1.0

{
    "dataset": {
        "activity": "none",
        "dataset": "dataset-030",
        "explanation": "NFSv4 operations per second",
        "grouping": "Protocol",
        "href": "/api/analytics/v1/datasets/nfs4.ops",
        "incore": 296128,
        "name": "nfs4.ops",
        "size": 53211540,
        "suspended": false
    }
}
```

Create Datasets

Creates a new dataset.

Example Request:

```
POST /api/analytics/v1/datasets HTTP/1.1
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 26

{"statistic": "test.sine"}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 200
```

```
Location: /api/analytics/v1/datasets/test.sine

{
    "dataset":{
        "href": "/api/analytics/v1/datasets",
        "name": "test.sine",
        "grouping": "Test",
        "explanation": "sine per second",
        "incore": 34752,
        "size": 31912,
        "suspended": false,
        "activity": "none"
    }
}
```

Modify Dataset

The modify dataset command is used to suspend or resume data collection of a single dataset.

Example Suspend Request:

```
POST /api/analytics/v1/datasets/nfs4.ops

{"suspended":true}
```

Example Resume Request:

```
POST /api/analytics/v1/datasets/nfs4.ops
        {"suspended":false}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 228
X-Zfssa-Analytics-Api: 1.0

{
    "dataset" {
        ...
        "suspended": false
    }
}
```

Destroy Datasets

Destroys a dataset.

Example Request:

```
DELETE /api/analytics/v1/datasets/test.sine HTTP/1.1
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Analytics-Api: 1.0
```

Save Dataset

Saves a dataset.

Example Request:

```
PUT /api/analytics/v1/datasets/nfs4.ops/data
```

Example Result:

```
HTTP/1.1 202 Accepted
```

Prune Dataset Data

The following table shows query parameters that are used in pruning datasets.

Parameter	Description
`granularity`	Prune granularity. The data within a dataset can be pruned at a granularity value of `second`, `minute`, or `hour`.
`endtime`	Prune data collected prior to the given `endtime`. For the format of this time value, see Query Parameter: start.

Example Request:

The following example deletes all per-second, per-minute, and per-hour data in the `nfs4.ops` dataset. Data that was collected daily, weekly, monthly, or annually remains in the dataset.

```
DELETE /api/analytics/v1/datasets/nfs4.ops/data?granularity=hour
```

Example Result:

```
HTTP/1.1 204 No Content
```

Example Request:

The following example deletes all per-second, per-minute, and per-hour data in the `nfs4.ops` dataset that was collected prior to the specified `endtime`.

```
DELETE /api/analytics/v1/datasets/nfs4.ops/data?
granularity=hour&endtime=20130910T00:00:00
```

Get Dataset Data

Returns data from an active analytic dataset. Both per-second and granular data retrieval are supported.

The following table shows time-based query parameters for getting dataset data.

Parameter	Description
`start`	The time to begin collecting sample data or the sample index at which to start collecting data. The `start` time value can be a specific time or can be the keyword `now`. For the format of a specific time value, see Query Parameter: start. The default `start` time is the current time minus the value of `seconds`.
`seconds`	Number of seconds to collect sample data. The default value is 1. The `seconds` parameter is ignored if the `span` and `granularity` parameters are specified.
`span`	Duration of time to collect sample data: `minute`, `hour`, `day`, `week`, `month`, or `year`.
`granularity`	The granularity within a given span from which the average of data points is given: `minute`, `hour`, `day`, `week`, `month`, or `year`.

The start time cannot be in the future. If the number of seconds to collect data goes beyond the current time, the server waits for each sample before returning the data.

To retrieve granular data, use a combination of parameters `span` and `granularity`. When `span` and `granularity` are used, the `seconds` parameter is ignored. If either `span` or `granularity` is entered incorrectly, the request is ignored and the `seconds` parameter is used instead. An incorrect or unsupported request displays the error message "Input span and granularity are not supported."

The `span` and `granularity` parameters can be combined in the following ways:

- If `span` is `minute`, `granularity` can only be `minute`.
- If `span` is `hour`, `granularity` can be `minute` or `hour`.
- If `span` is `day`, `granularity` can be `minute`, `hour`, or `day`.
- If `span` is `week`, `granularity` can be `hour`, `day`, or `week`.
- If `span` is `month`, `granularity` can be `day`, `week`, or `month`.
- If `span` is `year`, `granularity` can be `week`, `month`, or `year`.

The following table shows the dataset data properties that are returned.

Property	Description
`data`	Array of sample data
`sample`	Index number of the sample data
`startTime`	The time that `sample` was returned
`min`	The minimum value per second within the specified granularity
`max`	The maximum value per second within the specified granularity

Example request to collect two seconds of live data:

```
GET /api/analytics/v1/datasets/io.ops[op]/data?start=now&seconds=2 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: text/x-yaml
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: text/x-yaml
X-Zfssa-Analytics-Api: 1.0
Transfer-Encoding: chunked

{
  "data": [
    {
      "sample": 457642682,
      "data": {
        "data": [
          {
            "key": "write",
            "value": 199
          }
        ],
        "value": 199
      },
      "startTime": "20200818T18:43:47",
      "samples": 457642683
    },
    {
      "sample": 457642683,
      "data": {
        "data": [
          {
            "key": "write",
            "value": 289
          }
        ],
        "value": 289
      },
      "startTime": "20200818T18:43:48",
      "samples": 457642684
    }
  ]
}
```

Example request to collect seven days of data within the span of one week:

```
GET /api/analytics/v1/datasets/io.ops[op]/data?
start=20200811T15:00:00&granularity=day&span=week
    HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: text/x-yaml
```

The following request reports that the `io.ops[op]` dataset is "I/O operations per second broken down by type of operation" (read or write):

```
GET /api/analytics/v1/datasets/io.ops[op]
```

Example Result. Five of the seven output samples are omitted for brevity:

```
HTTP/1.1 200 OK
Content-Type: text/x-yaml
X-Zfssa-Analytics-Api: 1.0
Transfer-Encoding: chunked

{
  "data": [
    {
      "sample": 457197423,
      "data": {
        "max": 3156,
        "data": [
          {
            "max": 588,
            "key": "read",
            "value": 6,
            "min": 0
          },
          {
            "max": 3156,
            "key": "write",
            "value": 45,
            "min": 0
          }
        ],
        "value": 52,
        "min": 0
      },
      "startTime": "20200811T15:00:00",
      "samples": 457644011
    },
    {
      "sample": 457283823,
      "data": {
        "max": 3675,
        "data": [
          {
            "max": 588,
            "key": "read",
            "value": 6,
            "min": 0
          },
          {
            "max": 3675,
            "key": "write",
            "value": 45,
            "min": 0
          }
        ],
        "value": 52,
        "min": 0
      },
      "startTime": "20200812T15:00:23",
      "samples": 457644011
    }
  ]
}
```

You can also use a `sample` value as the `start` value. The result of the following request is the data from the specified one-second sample:

```
GET /api/analytics/v1/datasets/io.ops[op]/data?start=457642682 HTTP/1.1
```

6

Hardware Services

This section describes management of the hardware cluster, chassis, and components.

Cluster

The `cluster` commands set up clustering and manage clustered resources.

Table 6-1 Cluster Commands

Request	Append to Path /hardware/v{1|2}	Description
GET	/cluster	List cluster properties and resources
GET	/cluster/resources/*resource-path*	List properties for the specified cluster resource
PUT	/cluster/resources/*resource-path*	Modify the specified cluster resource
PUT	/cluster/failback	Fail back all resources assigned to the cluster peer
PUT	/cluster/takeover	Take over all resources assigned to the cluster peer
PUT	/cluster/unconfig	Unconfigure a clustered appliance to standalone mode
GET	/cluster/links	Show cluster card link status
PUT	/cluster/setup	Perform initial cluster setup

List Cluster Properties

The `cluster` command lists the current cluster configuration state and resource properties.

Example Request:

```
GET /api/hardware/v2/cluster HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 529
X-Zfssa-Api: 1.0

{
  "cluster": {
    "state": "AKCS_OWNER",
    "description": "Active (takeover completed)",
    "peer_asn": "d23331e6-41f4-6a15-ac09-a4353e33b43a",
    "peer_hostname": "peer-1",
    "peer_state": "AKCS_STRIPPED",
    "peer_description": "Ready (waiting for failback)",
    "resources": [
      {
        "owner": "peer-1",
        "type": "private",
        "user_label": "peer-1",
        "details": [
          "ipaddr"
        ],
        "href": "/api/hardware/v2/cluster/resources/net/vnic1"
      },
      {
        "owner": "peer-1",
        "type": "singleton",
        "user_label": "",
        "details": [
          "8.03T"
        ],
        "href": "/api/hardware/v2/cluster/resources/zfs/cas1"
      },
      {
        "owner": "peer-2",
        "type": "singleton",
        "user_label": "",
        "details": [
          "18.7T"
        ],
        "href": "/api/hardware/v2/cluster/resources/zfs/cas2"
      }
    ]
  }
}
```

Use the `href` property from one of the resources listed by the `cluster` command to list the properties for only that specific cluster resource, as shown in the following example:

```
GET /api/hardware/v2/cluster/resources/net/vnic1 HTTP/1.1
```

Modify a Cluster Resource

Use the `PUT` request with the `href` property from one of the resources listed by the `cluster` command to set the properties for that cluster resource.

Cluster Link Status

The `cluster/links` command returns the current link status of the cluster card.

Before performing initial cluster setup, use `cluster/links` to ensure that all links are in the AKCIOS_ACTIVE state. A connection that is not in the AKCIOS_ACTIVE state can mean that the other system is restarting/rebooting, or it can mean that the link is not cabled correctly or the cluster cables are not secure in their connectors. For cluster cabling instructions, see Connecting Cluster Cables in *Oracle ZFS Storage Appliance Cabling Guide, Release OS8.8.x*.

The output of the `cluster/links` command is different for different controller models. For Oracle ZFS Storage ZS9-2 controllers, status is returned for two cluster I/O links. For all other controllers, status is returned for three cluster I/O links. For a description of cluster I/O links, see Cluster Interconnect I/O in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
GET /api/hardware/v2/cluster/links HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result for Oracle ZFS Storage ZS9-2:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 84

{
    "links": {
        "lio_dev/i40e0 = AKCIOS_ACTIVE\n
        lio_dev/i40e1 = AKCIOS_ACTIVE"
    }
}
```

Example Result for Oracle ZFS Storage ZS7-2 or Oracle ZFS Storage ZS5-2 controllers:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 181

{
  "links": "\n\tclustron_ng3:0/clustron_uart:0 = AKCIOS_ACTIVE
            \n\tclustron_ng3:0/clustron_uart:1 = AKCIOS_ACTIVE
            \n\tclustron_ng3:0/dlpi:0 = AKCIOS_ACTIVE\n\n"
}
```

Other controllers show similar `links` output. The only difference is in the portion that precedes `/clustron` or `/dlpi`. For example, `clustron3_ng3:0` in the preceding example is `clustron3:0` for Oracle ZFS Storage ZS5-4 controllers, `clustron2:0` for Oracle ZFS Storage ZS4-4 controllers, and `clustron2_embedded:0` for Oracle ZFS Storage ZS3-2 controllers.

Cluster Management Commands

Cluster management includes `failback`, `takeover`, and `unconfig`. On success, the commands return HTTP status 202 (Accepted). If the cluster is not in the correct state to accept the command, an HTTP status 409 (Conflict) is returned.

Takeover is automatically attempted whenever peer failure is detected. Takeover can also be performed by an administrator.

Failback must be performed by an administrator. The failback operation is asynchronous. When the REST client sends a `failback` command, HTTP status 202 is returned when the request has been successfully received. To monitor failback progress, the client can listen for alerts or poll the cluster state.

For more information about takeover and failback, see Cluster Takeover and Failback in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
PUT /api/hardware/v2/cluster/failback HTTP/1.1
```

Unconfiguring a cluster node configures the node to standalone operation. In general, do not unconfigure a cluster node yourself. Unconfiguring a cluster node is destructive. Unconfiguring involves more than just the `unconfig` command. For more information, see Unconfiguring a Cluster Node in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Cluster Setup

Setup is one step in initial cluster configuration. For more information, see Upgrading a Standalone Appliance to a Clustered Configuration (BUI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

The `cluster/setup` command performs initial cluster configuration for the system. Specify values for the `nodename` and `password` properties. If setup is successful, HTTP status 202 Accepted is returned.

The `cluster/setup` command fails unless both of the following conditions exist:

- All cluster links are in the AKCIOS_ACTIVE state. See Cluster Link Status.
- The peer is powered on but not configured.

> **Note:**
>
> An initial cluster configuration setup can take several minutes to complete.

Example Request:

```
PUT /api/hardware/v2/cluster/setup HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json

{"nodename": "zfs-storage-2", "password": "password"}
```

Chassis

The hardware commands are used to get a list of appliance hardware chassis and components.

Table 6-2 Hardware Commands

Request	Append to Path /hardware/v{1|2}	Description
GET	/chassis	List hardware chassis
GET	/chassis/*chassis*	Get the specified hardware chassis properties

Table 6-2 (Cont.) Hardware Commands

Request	Append to Path /hardware/v{1|2}	Description
PUT	/chassis/*chassis*	Modify the specified hardware chassis properties
GET	/chassis/*chassis*/*fru_type*	List hardware chassis components
GET	/chassis/*chassis*/*fru_type*/*fru*	Get the specified chassis component properties
PUT	/chassis/*chassis*/*fru_type*/*fru*	Modify hardware chassis component properties

List Chassis

The get chassis command does not take any arguments and returns a list of system chassis objects. An HTTP status 200 (OK) is returned for a successful command.

Property	Type	Description
`name`	string	Chassis name
`model`	string	Chassis model number
`manufacturer`	string	Chassis manufacturer
`serial`	string	Chassis serial number
`revision`	string	Chassis revision level
`part`	string	Chassis replacement part number
`type`	string	Chassis storage type
`faulted`	boolean	Fault indicator
`uuid`	string	Chassis UUID identifier

Example Request:

```
GET /api/hardware/v1/chassis HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Length: 788
Content-Type: application/json
X-Zfssa-Appliance-Api: 1.0

{
    "hardware": [{
        "faulted": false,
        "href": "/api/hardware/v1/chassis/chassis-000",
        "manufacturer": "Oracle",
        "model": "Oracle ZFS Storage ZS3-2",
        "name": "hostname",
        "rpm": "--",
        "serial": "1211FM200C",
        "type": "system"
    }, {
        "faulted": false,
        "href": "/api/hardware/v1/chassis/chassis-001",
        "locate": false,
        "manufacturer": "Oracle",
        "model": "Oracle Storage DE2-24C",
        "name": "1235FM4002",
        "part": "7046842",
        "path": 2,
        "revision": "0010",
        "rpm": 7200,
        "serial": "1235FM4002",
        "type": "storage"
    }, {
        "faulted": false,
        "href": "/api/hardware/v1/chassis/chassis-002",
        "locate": false,
        "manufacturer": "Oracle",
        "model": "Oracle Storage DE2-24P",
        "name": "50050cc10c206b96",
        "part": "7046836",
        "path": 2,
        "revision": "0010",
        "rpm": 10000,
        "serial": "50050cc10c206b96",
        "type": "storage"
    }]
}
```

Get Chassis Components

This command returns all the hardware components within the specified chassis. An HTTP status 200 (OK) is returned for a successful command.

Example Request:

```
GET /api/hardware/v1/chassis/chassis-001 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "chassis": {
        "type": "storage"
        "faulted": false,
        "href": "/api/hardware/v1/chassis/chassis-001",
        "locate": false,
        "manufacturer": "Oracle",
        "model": "Oracle Storage DE2-24C",
        "name": "1235FM4002",
        "part": "7046842",
        "path": 2,
        "revision": "0010",
        "rpm": 7200,
        "serial": "1235FM4002",
        "disk": [{
            "device": "c0t5000CCA01A76A2B8d0",
            "faulted": false,
            "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-000",
            "interface": "SAS",
            "label": "HDD 0",
            "locate": false,
            "offline": false,
            "readytoremove": false,
            "manufacturer": "HITACHI",
            "model": "H7230AS60SUN3.0T",
            "pathcount": 2,
            "present": true,
            "revision": "A310",
            "rpm": 7200,
            "serial": "001210R37LVD--------YHJ37LVD",
            "size": 3000592982016,
            "type": "data",
            "use": "peer"
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-001",
            ...
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-002",
            ...
        },  ...  {
            "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-023",
             ...
        }],
    "fan": [
        {
            "href": "/api/hardware/v1/chassis/chassis-001/fan/fan-000",
            ...
        },  ...  {
            "href": "/api/hardware/v1/chassis/chassis-001/fan/fan-007",
        }],
    "psu": [
        {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-000",
            ...
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-001",
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-002",
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/psu/psu-003",
        }],
    "slot": [{
            "href": "/api/hardware/v1/chassis/chassis-001/slot/slot-000",
        }, {
            "href": "/api/hardware/v1/chassis/chassis-001/slot/slot-001",
        }],
    }
}
```

Get Hardware Component

This command returns the properties from a single hardware component. An HTTP status 200 (OK) is returned for a successful command. The response object contains the component properties shown in the following table.

The `offline`, `readytoremove`, and `use` properties apply only to disks in a pool.

Property	Type	Description
`device`	string	The field-replaceable unit (FRU) device ID
`faulted`	boolean	Whether the FRU is faulted
`interface`	string	FRU interface type
`label`	string	FRU location label
`locate`	boolean	Whether the FRU locate LED is on
`manufacturer`	string	FRU manufacturer
`model`	string	FRU model
`offline`	boolean	Whether the disk is offline
`pathcount`	integer	Total number of I/O paths to the disk shelf
`present`	boolean	FRU presence indicator
`readytoremove`	boolean	Whether the disk drive is ready to remove after fault
`revision`	string	Firmware or hardware revision of the FRU
`rpm`	number	Platter RPM (disk only)
`serial`	string	FRU serial number
`size`	number	FRU size (capacity)
`type`	string	Component type
`use`	string	Component usage enumeration

Example Request:

```
GET /api/hardware/v1/chassis/chassis-001/disk/disk-011 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "disk": {
        "device": "c0t5000CCA01A764FB0d0",
        "faulted": false,
        "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-011",
        "interface": "SAS",
        "label": "HDD 11",
        "locate": false,
        "offline": false,
        "readytoremove": false,
        "manufacturer": "HITACHI",
        "model": "H7230AS60SUN3.0T",
        "pathcount": 2,
        "present": true,
        "revision": "A310",
        "rpm": 7200,
        "serial": "001210R322ED--------YHJ322ED",
        "size": 3000592982016,
        "type": "data",
        "use": "peer"
    }
}
```

Modify Component Property

A PUT request can be used to set properties on a selected hardware component. A successful request returns HTTP status 201 (Accepted) as well as the component properties in JSON format.

Example Request:

```
PUT /api/hardware/v1/chassis/chassis-001/disk/disk-011 HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Accept: application/json
Content-Type: application/json
Content-Length: 16

{"locate": true}
```

Example JSON Response:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Length: 403
Content-Type: application/json

{
    "disk": {
        "href": "/api/hardware/v1/chassis/chassis-001/disk/disk-011",
        ...,
        "locate": true
    }
}
```

7
Log Commands

The log commands manage the logs available under the CLI `maintenance logs` menu. For individual service log information, see the service API.

Manage Logs Commands

The following table show how to call the manage logs commands.

Table 7-1 Manage Logs Commands

Request	Append to Path /api/log/v{1|2}	Description
GET	Use only /api/log/v{1|2}	List the log service commands
GET	/logs	List all log types
GET	/logs/?start=*index/time*&limit=*entry limit*	Get log entries for the selected range
GET	/logs/alert	List all alert logs
GET	/logs/alert?start=*index/time*&limit=*entry limit*	Get log entries for the selected range
GET	/collect	Download a collection of all log entries
GET	/collect?start=*index/time*&limit=*entry limit*	Download a collection of log entries from a selected range

List Logs

This command lists all of the logs available on an appliance. Each log returns the number of entries in the log and a time stamp of the last entry.

> **Note:**
>
> The `depth` query parameter and the `match_`*`property-name`*`=`*`value`* query parameter are not supported.

Example Request:

```
GET /api/log/v1/logs HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 532
```

```
X-Zfssa-Api: 1.0

{
    "logs": [
        {
            "href": "/api/log/v1/logs/fault",
            "name": "faults",
            "size": 16,
            "updated": "20130614T22:51:48"
        },
        {
            "href": "/api/log/v1/logs/audit",
            "name": "audits",
            "size": 460149,
            "updated": "20130730T22:10:41"
        },
        {
            "href": "/api/log/v1/logs/alert",
            "name": "alerts",
            "size": 13054,
            "updated": "20130728T00:06:10"
        },
        {
            "href": "/api/log/v1/logs/phone-home",
            "name": "phone-home",
            "size": 249,
            "updated": "20130730T03:22:35"
        },
        {
            "href": "/api/log/v1/logs/system",
            "name": "system",
            "size": 344,
            "updated": "20130724T03:21:55"
        }
    ]
}
```

Get Log Entries

Log entries can be returned from the specified appliance log. Each log entry returns the date/time of the entry along with log specific content properties.

Note:

Depending on the number of logs, older log entries might not be available due to memory constraints. This same limit occurs in the BUI and CLI. To obtain all system logs, use the `collect` function described in Manage Logs Commands.

Parameter	Description
`start=`*`index`*	Start returning logs from the given index/time
`limit=`*`number`*	Limit number of log entries returned

The start index defaults to the value of 0, which returns the first log that was generated. Negative values and values greater than or equal to the log size are not allowed. The start index can also be a time string; for example, 20130724T03:21:55.

> **Note:**
>
> REST only accepts UTC time. Time values that are older than one month from the current time are not accepted. Retrieval of older logs must use an index number for the start value. The limit value limits the number of logs returned for a given request. No more than the given limit value is returned.

Example Request:

```
GET /api/log/v1/logs/audit?limit=4&start=1000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
X-Zfssa-Api: development
Transfer-Encoding: chunked

{
    "logs": [
        {
            "address": "192.0.2.0",
            "annotation": "",
            "summary": "User logged in",
            "timestamp": "20131022T22:54:19",
            "user": "root"
        }, {
            "address": "192.0.2.0",
            "annotation": "",
            "summary": "Destroyed share \"zfs-storage-1:tst.volumes.py.34111.project/
tst.volumes.py.34111.lun.7\"",
            "timestamp": "20131022T22:52:34",
            "user": "root"
        }, {
            "summary": "Joined workgroup \"RESTTESTWG\"",
            "timestamp": "20131022T22:54:23",
            "user": "<system>"
        }, {
            "address": "192.0.2.0",
            "annotation": "",
            "summary": "User logged in",
            "timestamp": "20131022T22:54:19",
            "user": "root"
        }
    ]
}
```

Download Logs

The download logs command returns a gzipped tar file containing all of the system logs. The file disposition name is set to `logs.tar.gz`. Because the data is created and streamed in real time, it is not possible to resume a download.

Download Log

If only one log type is desired to be downloaded, its name can be appended to the `collect` resource as shown in the table. The text of the log is streamed back to the client. If `gzip` compression is requested, the text stream is compressed with `gzip`. Other compression types are not supported and are ignored.

8

Network Commands

The network commands described in this section are used to view network addresses and devices as well as configure network datalinks, interfaces, and routes.

Networking Configuration

The network configuration features let you create a variety of advanced networking setups out of your physical network ports, including link aggregations, virtual NICs (VNICs), virtual LANs (VLANs), and IP network multipathing (IPMP) groups. You can then define any number of IPv4 and IPv6 addresses for these abstractions, for use in connecting to the various data services on the system.

There are four components to a system's network configuration:

- **Devices** – Physical network ports that correspond to your physical network connections or IP on InfiniBand (IPoIB) partitions.
- **Datalinks** – The basic construct for sending and receiving packets. Datalinks may correspond 1:1 with a device (that is, with a physical network port) or IB Partition, or you can define Aggregation, VLAN and VNIC datalinks composed of other devices and datalinks.
- **Interface** – The basic construct for IP configuration and addressing. Each IP interface is associated with a single datalink, or is defined as an IPMP group, which is comprised of other interfaces.
- **Routing** – IP routing configuration, which controls how the system directs IP packets.

In this model, network devices represent the available hardware; they have no configurable settings. Datalinks are a layer 2 entity and must be created to apply settings such as LACP to these network devices. Interfaces are a layer 3 entity containing the IP settings, which they make available via a datalink. This model has separated network interface settings into two parts: datalinks for layer 2 settings and interfaces for layer 3 settings.

Network Datalinks

The network datalinks command provides datalink management on Oracle ZFS Storage Appliance. You can list, modify, create, and delete datalink resources.

Table 8-1 Network Datalink Commands

Request	Append to Path /network/v{1|2}	Description
POST	/datalinks	Create a new network datalink
GET	/datalinks/*datalink*	Get the specified network datalink properties
GET	/datalinks	List all network datalink objects
PUT	/datalinks/*datalink*	Modify the specified network datalink object
DELETE	/datalinks/*datalink*	Destroy the specified datalink object

Table 8-2 Physical Device Datalink Properties

Property	Type	Description
class	String	“device” (“immutable”)
label	NetworkLabel	Label
links	ChooseOne	Links [“igb1”, “igb0”, “ixgbe2”, “ixgbe3”, “igb4”, “igb3”, “ixgbe1”, “igb2”, “igb5”]
jumbo	Boolean	Use Jumbo Frames [“true”, “false”] (“deprecated”)
mtu	PositiveInteger	Max transmission unit (MTU)
lro	ChooseOne	Large receive offload (LRO) ["on", "off"]
speed	ChooseOne	Link Speed [“auto”, “10”, “100”, “1000”, “10000”]
duplex	ChooseOne	Link Duplex [“auto”, “half”, “full”]

Table 8-3 VNIC Device Datalink Properties

Property	Type	Description
class	String	“vnic” (“immutable”)
label	NetworkLabel	Label
links	ChooseOne	Links [“ixgbe0”]
mtu	PositiveInteger	Max transmission unit (MTU)
lro	ChooseOne	Large receive offload (LRO) ["on", "off"]
id	VLAN	VLAN ID

Table 8-4 VLAN Device Datalink Properties

Property	Type	Description
class	String	“vlan” (“immutable”)
label	NetworkLabel	Label
links	ChooseOne	Links [“ixgbe0”]
mtu	PositiveInteger	Max transmission unit (MTU)
lro	ChooseOne	Large receive offload (LRO) ["on", "off"]
id	VLAN	VLAN ID

Table 8-5 Aggregation Based Device Datalink Properties

Property	Type	Description
class	String	“aggregation” (“immutable”)
label	NetworkLabel	Label
links	ChooseN	Links [“igb1”, “igb0”, “ixgbe2”, “ixgbe3”, “igb4”, “igb3”, “ixgbe1”, “igb2”, “igb5”]

Table 8-5 (Cont.) Aggregation Based Device Datalink Properties

Property	Type	Description
`jumbo`	Boolean	Use Jumbo Frames [“true”, “false”] (“deprecated”)
`mtu`	PositiveInteger	Max transmission unit (MTU)
`lro`	ChooseOne	Large receive offload (LRO) ["on", "off"]
`policy`	ChooseOne	Policy [“L2”, “L3”, “L4”, “L2+L3”, “L2+L4”, “L3+L4”]
`mode`	ChooseOne	Mode [“active”, “passive”, “off”]
`timer`	ChooseOne	Timer [“short”, “long”]
`key`	Integer	Aggregation Key (“immutable”)

Table 8-6 IP-Partition-Based Device Datalink Properties

Property	Type	Description
`class`	String	“partition” (“immutable”)
`label`	NetworkLabel	Label
`links`	ChooseOne	Links
`pkey`	Pkey	Partition Key
`linkmode`	ChooseOne	Link Mode [“cm”, “ud”]

List Network Datalinks

Lists all configured datalinks on Oracle ZFS Storage Appliance. Each object in the datalinks list contains an `href` to get the operation on a single datalink resource along with datalink properties.

Example Request:

```
GET /api/network/v1/datalinks HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example JSON Data:

```
{
    "datalinks": [{
        "href": "/api/network/v1/datalinks/ixgbe0",
        ...
    }, {
        "href": "/api/network/v1/datalinks/ixgbe1",
        ...
    }, {
        "href": "/api/network/v1/datalinks/ixgbe2",
        ...
    }, {
        "href": "/api/network/v1/datalinks/ixgbe3",
        ...
    }]
}
```

Get Network Datalink

The `GET` method returns a JSON object that contains a datalink property with a list of datalink objects.

```
GET /api/network/v1/datalinks/ixgbe0 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example JSON Data:

```
{
    "datalink": {
        "class": "device",
        "datalink": "ixgbe0",
        "duplex": "auto",
        "href": "/api/network/v1/datalinks/ixgbe0",
        "jumbo": false,
        "label": "Untitled Datalink",
        "links": [
            "ixgbe0"
        ],
        "mac": "0:21:28:a1:d9:68",
        "mtu": 1500,
        "speed": "auto"
    }
}
```

Create Network Datalink

The `POST` command creates a new datalink. One additional property that is needed when creating a new datalink is the `class` property, which defines the class of datalink to create. The datalinks class is defined during datalink creation and can be one of the following class types:

- `device` – Create a device-based datalink
- `vnic` – Create a VNIC-based datalink
- `vlan` – Create a VLAN-based datalink
- `aggregation` – Create an aggregation-based datalink
- `partition` – Create an IB partition datalink

The properties map to the same CLI properties available in the `configuration net datalinks` menu.

Example Request:

```
POST /api/network/v1/datalinks HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 78

{
    "class": "device",
    "jumbo": true,
    "links": ["ixgbe2"],
    "label": "TestDataLink"
}
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Location: /api/network/v1/datalinks/ixgbe2
```

Modify Network Datalink

The PUT method is used to modify datalink properties. For details on setting up datalinks, see the CLI documentation.

Example Request:

```
PUT /api/network/v1/datalinks/ixgbe2 HTTP/1.1

{"jumbo": true}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 219

{
    "datalink": {
        "href": "/api/network/v1/datalinks/ixgbe2",
        "class": "device",
        "label": "MyDataLink",
        "links": ["ixgbe2"],
        "mac": "0:21:28:a1:d9:6a",
        "mtu": 9000,
        "duplex": "auto",
        "jumbo": true,
        "speed": "auto"
    }
}
```

Delete Network Datalink

This command removes the datalink from the system. Use the href path to delete the specified datalink.

Example Request:

```
DELETE /api/network/v1/datalinks/ixgbe2 HTTP/1.1
```

Example Result:

```
HTTP/1.1 204 No Content
```

Network Devices

These commands list the physical network devices on the system. There are no modifiable properties on physical network devices.

Table 8-7 Network Devices Commands

Request	Append to Path /network/v{1|2}	Description
GET	/devices/*device*	Get the specified network device properties
GET	/devices	List all network device objects

Table 8-8 Network Device Properties

Property	Description
`active`	Boolean flag indicating whether the device is active
`duplex`	Duplex of device
`factory_mac`	Factory MAC address
`media`	Device media
`speed`	Device speed, in megabits/second
`up`	Boolean flag indicating whether the device is operational

List Network Devices

This command lists all network devices.

Example Request:

```
GET /api/network/v1/devices HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 412
X-Zfssa-Gns-Api: 1.0

{
    "devices": [{
        "href": "/api/network/v1/devices/ixgbe0",
        ....
    }, {
        "href": "/api/network/v1/devices/ixgbe1",
        ...
    }, {
        "href": "/api/network/v1/devices/ixgbe2",
        ...
    }, {
        "href": "/api/network/v1/devices/ixgbe3",
        ...
    }]
}
```

Get Network Device

This command gets the properties from a single network device.

Example Request:

```
GET /api/network/v1/devices/ixgbe0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 178
X-Zfssa-Gns-Api: 1.0

{
    "devices": {
        "active": false,
        "device": "ixgbe0",
        "duplex": "full-duplex",
        "factory_mac": "0:21:28:a1:d9:68",
        "href": "/api/network/v1/devices/ixgbe0",
        "media": "Ethernet",
        "speed": "1000 Mbit/s",
        "up": true
    }
}
```

Network Interfaces

Table 8-9 Network Interface Commands

Request	Append to Path /api/network/v{1|2}	Description
POST	/interfaces	Create a new network interface
GET	/interfaces/*interface*	Get the specified network interface properties
GET	/interfaces	List all network interface objects
PUT	/interfaces/*interface*	Modify the specified network interface object
DELETE	/interfaces/*interface*	Destroy the specified interface object

Table 8-10 Network Interface Properties

Property	Description
`admin`	Flag indicating whether administration is allowed on this interface
`class`	Class type ("ip", "ipmp") (immutable after create)
`curaddrs`	Current IP Addresses (immutable)

Table 8-10 (Cont.) Network Interface Properties

Property	Description
`enable`	Flag indicating whether this interface is enabled
`label`	User label for interface
`links`	Chose a network link for this interface
`state`	State of Interface (immutable)
`v4addrs`	IPv4 Addresses
`v6dhcp`	IPv4 DHCP flag
`v6addrs`	IPv6 Addresses
`v6dhcp`	IPv6 DHCP flag

List Network Interfaces

This command lists all of the configured network interfaces.

Example Request:

```
GET /api/network/v1/interfaces HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 369

{
    "interfaces": {[
        "href": "/api/network/v1/interfaces/ixgbe0",
        "v4addrs": ["ipaddr-1"]
        ...
    }, {
        "href": "/api/network/v1/interfaces/ixgbe1",
        "v4addrs": ["ipaddr-2"]
        ...
    }, {
        "href": "/api/network/v1/interfaces/ixgbe2",
        "v4addrs": ["ipaddr-3"]
        ...
    }, {
        "href": "/api/network/v1/interfaces/ixgbe3",
        "v4addrs": ["ipaddr-4"]
        ...
    }]
}
```

Get Network Interface

This command gets the full list of properties for a specified network interface.

Example Request:

```
GET /api/network/v1/interfaces/ixgbe0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 292

{
    "interface": {
        "admin": true,
        "class": "ip",
        "curaddrs": ["ipaddr-1"],
        "enable": true,
        "href": "/api/network/v1/interfaces/ixgbe0",
        "interface": "ixgbe0",
        "label": "Untitled Interface",
        "links": ["ixgbe0"],
        "state": "up",
        "v4addrs": ["ipaddr-1"],
        "v4dhcp": false,
        "v6addrs": [],
        "v6dhcp": false
    }
}
```

Create Network Interface

This command creates a new network interface.

Example Request:

```
POST /api/network/v1/interfaces HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: root
X-Auth-Key: password
Content-Type: application/json
Content-Length: 78

{
    "class": "ip",
    "links": ["ixgbe3"],
    "v4addrs":"192.0.2.0/24"
}
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Location: /api/network/v1/interfaces/ixgbe3
```

Modify Network Interface

This command modifies an existing network interface.

Example Request:

```
PUT /api/network/v1/interfaces/ixgbe3 HTTP/1.1

{
    "v4addrs": ["192.0.2.0/24"],
    "interface": "Demo Rest"
}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 219

{
    "admin": true,
    "class": "ip",
    "curaddrs": ["192.0.2.0/24"],
    "enable": true,
    "href": "/api/network/v1/interfaces/ixgbe3",
    "interface": "ixgbe3",
    "label": "Demo Rest",
    "links": ["ixgbe3"],
    "state": "failed",
    "v4addrs": ["192.0.2.0/24"]
    "v4dhcp": false,
    "v6addrs": [],
    "v6dhcp": false
}
```

Delete Network Interface

This command deletes an existing network interface.

> **Note:**
>
> When an interface is deleted, all routes associated with the interface are also removed.

Example Request:

```
DELETE /api/network/v1/interfaces/ixgbe3 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Example Result:

```
HTTP/1.1 204 No Content
```

Network Routes

These commands manage network routes.

Table 8-11 Manage Network Routes

Request	Append to Path /api/network/v{1|2}	Description
POST	/routes	Create a new network route
GET	/routes/*route*	Get the specified network route properties
GET	/routes	List all network route objects
DELETE	/routes/*route*	Destroy the specified route object
GET	/routing	Get net routing properties
PUT	/routing	Modify net routing properties

Table 8-12 Manage Network Route Properties

Property	Description
`type`	Type of route such as “system” or “static” (immutable)
`family`	Address family (either IPv4 or IPv6)
`destination`	Route destination address
`gateway`	Gateway address
`interface`	Network datalink interface

The `href` path to each route uses the route IDs set in the CLI, but these values can change as routes are modified. The API supports selecting single routes using unique properties within the route. The syntax is `routes/name=`*`value`* compared to `routes/route-###`.

List Routes

Lists all of the network routes created on Oracle ZFS Storage Appliance.

Example Request:

```
GET /api/network/v1/routes HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 192

{
    "routes": [{
        "destination": "ipaddr-0",
        "family": "IPv4",
        "gateway": "ipaddr-1",
        "href": "/api/network/v1/routing/route-000",
        "interface": "ixgbe0",
        "mask": 0,
        "route": "route-000",
        "type": "static"
    }, {
        "destination": "ipaddr-2",
        "family": "IPv4",
        "gateway": "ipaddr-3",
        "href": "/api/network/v1/routes/route-001",
        "interface": "ixgbe0",
        "mask": 24,
        "route": "route-001",
        "type": "system"
    }]
}
```

Get Route

Gets the properties for a single route.

Example Request:

```
GET /api/network/v1/routes/destination=ipaddr-1 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 192

{
    "route": {
        "destination": "ipaddr-1",
        "family": "IPv4",
        "gateway": "ipaddr-2",
        "href": "/api/network/v1/routes/route-001",
        "interface": "ixgbe0",
        "mask": 24,
        "route": "route-001",
        "type": "system"
    }
}
```

Add Route

Creates a new network route. The route href values can change if other routes are added to the system. No route information is returned on a create since the returned properties would be identical to the input properties. A successful create returns HTTP status 204 (Created).

Example Request to Create a Static Route:

```
POST /api/network/v1/routes HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 164

{
    "family": "IPv4",
    "destination": "ipaddr-0",
    "mask": "0",
    "gateway": "ipaddr-1",
    "interface": "ixgbe0"
}
```

Example Result:

```
HTTP/1.1 201 Created
```

Delete Route

Deletes an existing network route.

Example Request:

```
DELETE /api/network/v1/routes/route-001 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Example Result:

```
HTTP/1.1 204 No Content
```

9 RESTful API Cloud Service

The RESTful API Cloud service enables an Oracle ZFS Storage Appliance administrator to do the following tasks:

- Back up ZFS snapshot data from Oracle ZFS Storage Appliance to Oracle Cloud Infrastructure Object Storage
- List the snapshot backups that are available in the cloud
- Restore snapshot backups to the appliance
- Delete snapshot backup data when that data is no longer needed

More than one appliance can back up to the same cloud target. A cloud backup can be restored to any appliance that has access to the cloud target.

For detailed information about cloud backup, including the following topics, see Configuring Cloud Backup in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*:

- Configure your Oracle Cloud Infrastructure account, including “standard” or “archive” tier
- Manage data buckets, metadata buckets, and archive buckets
- Clustered configuration effects on cloud backup operations

For information about the Oracle Cloud Infrastructure Object Storage service, including the following operations, see Oracle ZFS Storage Appliance RESTful API Support for the Oracle Cloud Infrastructure Object Storage Service in *Oracle ZFS Storage Appliance Object API Guide for Oracle Cloud Infrastructure Object Storage Service Support, Release OS8.8.x*:

- Enable the Oracle Cloud Infrastructure service
- Create or delete a user key; change operation permissions for a user key

Cloud Service Operations

Use cloud service operations to manage backups of Oracle ZFS Storage Appliance snapshots in Oracle Cloud Infrastructure Object Storage. You can list targets and backups in Oracle Cloud Infrastructure Object Storage, remove a target, delete a backup, restore a backup to an appliance, and cancel or restart a cloud service job.

To create a backup, see Create a Snapshot Backup in Snapshot Backup Operations. Backups are full or incremental backups of share snapshots. To create a snapshot, see Snapshot and Clone Operations.

The following parameters are used in the table of cloud service operations:

format
The format in which the backup was saved in Oracle Cloud Infrastructure Object Storage.

The value of *format* is either `zfs` or `tar`. If *format* is not specified for snapshot backup creation, `zfs` is the default. The `zfs` format supports both filesystem and LUN snapshots; the `tar` format supports only filesystem snapshots. For more information about `zfs` and `tar` formats, see Creating a Cloud Backup (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

backup-id
The identifier for a copy of an appliance snapshot that is saved in Oracle Cloud Infrastructure Object Storage. The value of *backup-id* is the value of the `id` property when you list backups as shown in List Cloud Backups.

For a full snapshot backup, the value of *backup-id* is *pool_id*/*snapshot_id*, as shown in the following example:

```
3e035b7e546e0d02/1cbfdb5ff2259b76
```

For an incremental snapshot backup, the value of *backup-id* is *pool_id*/*child_snapshot_id-parent_snapshot_id*, as shown in the following example:

```
6913a5703bee98dc/46be95ced54e99d9-667f3eb88fd209e1
```

target-id
The Oracle Cloud Infrastructure Object Storage location where a backup is saved. A given backup can be saved to multiple targets. That is, the same *backup-id* can appear in different *target-id* locations.

The value of *target-id* is the value of the `target` property when you list backups.

job-id
The identifier for a running job. The value of *job-id* is the value of the `id` property when you list jobs as shown in List Jobs.

Table 9-1 Cloud Service Commands

Request	Append to Path /api/service/v2/services	Description
GET	/cloud	List the properties and summary data for targets, backups, and jobs
PUT	/cloud	Modify properties
POST	/cloud/targets	Create a new target
GET	/cloud/targets	List targets
GET	/cloud/targets/*target-id*	List properties of the specified target
PUT	/cloud/targets/*target-id*	Modify properties of the specified target
DELETE	/cloud/targets/*target-id*	Remove the specified target from service
GET	/cloud/backups	List all completed backups of any format
GET	/cloud/backups/*format*/*backup-id*/*target-id*	List the specified backup
DELETE	/cloud/backups/*format*/*backup-id*/*target-id*	Delete the specified backup (submit a job request)
POST	/cloud/backups/*format*/*backup-id*/*target-id*/restore	Restore the specified backup (submit a job request)

Table 9-1 (Cont.) Cloud Service Commands

Request	Append to Path /api/service/v2/services	Description
GET	/cloud/jobs	List running jobs and recently completed jobs
GET	/cloud/jobs/*job-id*	List properties of the specified job
PUT	/cloud/jobs/*job-id*/cancel	Cancel the specified running job
PUT	/cloud/jobs/*job-id*/restart	Restart the specified aborted job

Enable the Cloud Service

To enable the cloud service, set the `status` to `enable` as shown in the following example.

Example Request:

```
PUT /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 22

{ "<status>": "enable" }
```

View the Cloud Service Log File

Use the following request to view the log file for the cloud service:

```
GET /api/log/v1/logs/appliance-kit-cloud:default HTTP/1.1
```

List Cloud Service Properties

Example Request:

```
GET /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 20:30:59 GMT
Content-Length: 843
X-Zfssa-Service-Api: 2.0
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8

{
  "service": {
    "href": "/api/service/v2/services/cloud",
    "<status>": "online",
    "tls_version": [
      "TLSv1.1",
      "TLSv1.2"
    ],
    "ciphers": [
      "ECDHE-RSA-AES128-GCM-SHA256",
      "ECDHE-RSA-AES256-GCM-SHA384",
      "DHE-RSA-AES128-GCM-SHA256",
      "DHE-RSA-AES256-GCM-SHA384",
      "AES128-SHA",
      "AES256-SHA",
      "DES-CBC3-SHA"
    ],
    "targets": {
      "href": "/api/service/v2/services/cloud/targets",
      "entries": 2
    },
    "backups": {
      "href": "/api/service/v2/services/cloud/backups",
      "entries": 2548
    },
    "jobs": {
      "href": "/api/service/v2/services/cloud/jobs",
      "entries": 0
    }
  }
}
```

Modify Cloud Service Properties

Example Request:

```
PUT /api/service/v2/services/cloud HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type: application/json
Content-Length: 48

{
    "tls_version": [
        "TLSv1.0","TLSv1.1","TLSv1.2"
    ]
}
```

List Targets

The following example lists all targets.

Example Request:

```
GET /api/service/v2/services/cloud/targets HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

```
HTTP/1.1 200 OK
Date: Wed, 24 Jul 2019 21:06:18 GMT
```

```
Content-Length: 1086
X-Zfssa-Service-Api: 2.0
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8

{
    "targets": [{
        "bucket": "pl-test",
        "href": "/api/service/v2/services/cloud/targets/target-id1",
        "id": "target-id1",
        "key": true,
        "location": "https://objectstorage.us-ashburn-1.oraclecloud.com",
        "name": "oci-ashburn",
        "online": true,
        "proxy_host": "",
        "proxy_on": false,
        "proxy_password": false,
        "proxy_user": "",
        "tenancy": "ocid1.tenancy.oc1..tenancy-id",
        "user": "ocid1.user.oc1..user-id"
    },{
        "bucket": "pl-test2",
        "href": "/api/service/v2/services/cloud/targets/target-id2",
        "id": "target-id2",
        "key": true,
        "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
        "name": "oci-phoenix",
        "online": true,
        "proxy_host": "www-proxy.example.com:80",
        "proxy_on": true,
        "proxy_password": false,
        "proxy_user": "",
        "tenancy": "ocid1.tenancy.oc1..tenancy-id",
        "user": "ocid1.user.oc1..user-id"
    }]
}
```

The following example lists the specified target:

```
GET /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Create a Target

The following parameters are required to create a cloud target.

Parameter	Description
key	An Oracle Cloud Infrastructure account user key for *user-id*
tenancy-id	An Oracle Cloud Infrastructure account tenancy name OCID
user-id	Your Oracle Cloud Infrastructure account user name OCID

The following example creates a target.

Example Request:

```
POST /api/service/v2/services/cloud/targets HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=

{
    "name": "oci4",
    "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
    "user": "ocid1.user.oc1..user-id",
    "bucket": "test-bucket3",
    "tenancy": "ocid1.tenancy.oc1..tenancy-id",
    "key": "key",
    "proxy_on": false,
    "readlimit": -1,
    "writelimit": -1
}
```

Example Result:

```
HTTP/1.1 201 Created
Date: Wed, 24 Jul 2019 21:14:39 GMT
Content-Length: 568
X-Zfssa-Service-Api: 2.0
Location: /api/service/v2/services/cloud/targets/target-id
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8

{
    "target": {
        "bucket": "test-bucket3",
        "href": "/api/service/v2/services/cloud/targets/target-id",
        "id": "target-id",
        "key": true,
        "location": "https://objectstorage.us-phoenix-1.oraclecloud.com",
        "name": "oci4",
        "proxy_on": false,
        "readlimit": -1,
        "state": "offline",
        "tenancy": "ocid1.tenancy.oc1..tenancy-id",
        "user": "ocid1.user.oc1..user-id",
        "writelimit": -1
    }
}
```

Modify a Target

The following table shows the properties that can be modified for a cloud target.

Property	Description
`name`	The name for this cloud target, which must be unique on each Oracle ZFS Storage Appliance
`proxy_on`	If `true`, use a proxy for system communications with the internet. If the value of `proxy_on` is `true`, then you must provide a value for `proxy_host`.
`proxy_host`	The proxy hostname and port number
`proxy_password`	Optional. The proxy password
`proxy_user`	Optional. The proxy username

Property	Description
readlimit	Optional. The maximum rate in bytes per second that data will be read from the cloud target. This value is used when a cloud backup is restored from the cloud. For example, a value of 4194304 limits the read rate from the cloud target to 4 megabytes per second. The value -1 means I/O is not limited.
writelimit	Optional. The maximum rate in bytes per second that data will be written to the cloud target. This value is used when a cloud snapshot is uploaded to a cloud target. For example, a value of 5242880 limits the write rate to the cloud target to 5 megabytes per second. The value -1 means I/O is not limited.

Example Request:

```
PUT /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
Content-Type: application/json
Content-Length: 19
{
    "proxy_on": true,
    "proxy_host": "www-proxy.example.com:80",
    "readlimit": 4194304,
    "writelimit": 5242880
}
```

Delete a Target

Before you delete a target, perform the following checks:

- Check whether a backup to this target is in progress. See List Jobs.
- Determine whether this target has backups. Use the `target` filter as shown in List Cloud Backups to list backups that are stored on this target.

The following example removes the specified cloud target from service.

Example Request:

```
DELETE /api/service/v2/services/cloud/targets/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

```
HTTP/1.1 204 No Content
Date: Wed, 24 Jul 2019 21:20:27 GMT
X-Content-Type-Options: nosniff
X-Zfssa-Service-Api: 2.0
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8
```

List Cloud Backups

Use the following query to list all completed backups of any format, with newer backups listed first. To get information about backups in progress, see List Jobs.

Example Request:

```
GET /api/service/v2/services/cloud/backups HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

In this example, the first backup listed is the child of the second backup listed. The value of `parent` of the first backup is the same as the value of `dataset` of the second backup (`app-data-fullsnap`), and the value of `parent` of the second backup is null. A child backup will be the same `format` as the parent. In this example, both child and parent backups are in `tar` format.

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 21:22:40 GMT
X-Zfssa-Service-Api: 2.0
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked

{
    "backups": [{
        "target": "target-id2",
        "parent": "app-data-fullsnap",
        "started": "2020-01-06T20:03:32Z",
        "completed": "2020-01-06T20:04:36Z",
        "id": "backup-id2",
        "source": "cloudsnap0",
        "href": "/api/service/v2/services/cloud/backups/tar/backup-id2/target-
id2",
        "tier": "standard",
        "dataset": "p1/local/default/app-data@app-data-incsnap",
        "format": "tar",
        "size": 3224982536.0
    },{
        "target": "target-id1",
        "parent": "",
        "started": "2020-01-06T20:01:16Z",
        "completed": "2020-01-06T20:01:17Z",
        "id": "backup-id1",
        "source": "cloudsnap0",
        "href": "/api/service/v2/services/cloud/backups/tar/backup-id1/target-
id1",
        "tier": "standard",
        "dataset": "p1/local/default/app-data@app-data-fullsnap",
        "format": "tar",
        "size": 2149988056.0
    }]
}
```

Use the following request to list the specified backup:

```
GET /api/service/v2/services/cloud/backups/format/backup-id/target-id HTTP/1.1
```

List Cloud Backups Query Parameters

The following query parameters are supported for filtering the list of cloud backups. All parameters can be used in the same query. Use an AND operation (&) between the query parameters as shown in the examples that follow the table.

Table 9-2 Query Parameters for Listing Cloud Backups

Property	Description
`start`	The value of `start` is one of the following: • The backup index. Values for this index are from 0 to one less than the value of the `entries` property in the `backups` section of the request shown in List Cloud Service Properties. Lists the backup selected by the specified index and the oldest backups that were created after the specified backup was created. • A time in format `%Y-%m-%dT%H:%M:%SZ`. This is the same format that is used for the values of the `started` and `completed` properties. This time can be any real time up to the current time. For example, `2019-09-00T00:00:00Z` is not a real time. Lists the oldest backups that were created on or after the specified time.
`end`	The value of `end` is a time in format `%Y-%m-%dT%H:%M:%SZ`. This is the same format that is used for the values of the `started` and `completed` properties. This time can be any real time up to the current time. For example, `2019-09-00T00:00:00Z` is not a real time. Lists the newest backups that were created on or before the specified time.
`limit`	Lists no more than the specified number of backups. The `limit` parameter has no default value.
`target`	The value of the `target` property. Lists the newest backups on that target.
`source`	The value of the `source` property. Lists the newest backups on that source.
`dataset`	The value of the `dataset` property before the `@`. For example, for dataset `p1/local/default/app-data@app-data-incsnap`, specify `p1/local/default/app-data` or `app-data`. Lists the newest backups in the specified dataset. **Note -** A request for a dataset backup lists all backups of all datasets that have names that contain the requested dataset name. For example, a request for backups of `p1/local/default/app-data@app-data-incsnap` will also return backups of datasets named `app-data-incsnap`, `app-data-incsnap-1`, and `myproj-app-data-incsnap`.
`format`	The value of the `format` property, either `zfs` or `tar`. Lists the newest backups in the specified backup format.

The following example lists the oldest backups starting from the backup with index number 2000. If the value of the `entries` property in the `backups` section of the cloud service properties list is 2865, then the following example shows 865 backups: backup 2000 through backup 2864.

```
GET /api/service/v2/services/cloud/backups?start=2000
```

The following example lists only the backup with index number 2000:

```
GET /api/service/v2/services/cloud/backups?start=2000&limit=1
```

The following example lists the 500 oldest backups that were created on or after the specified time:

```
GET /api/service/v2/services/cloud/backups?start=2019-07-12T00:00:00Z&limit=500
```

The following example lists the 500 newest backups that were created on or before the specified time:

```
GET /api/service/v2/services/cloud/backups?end=2019-07-12T00:00:00Z&limit=500
```

The following example lists all backups that were created on or after the specified `start` time but no later than the specified `end` time:

```
GET /api/service/v2/services/cloud/backups?
start=2019-07-11T00:00:00Z&end=2019-07-12T00:00:00Z
```

The following example lists the newest backups of any dataset with `app-data` in the name in the `tar` backup format:

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&format=tar
```

The following example lists the newest backups of any dataset with `app-data` in the name on the *target-id* target:

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&target=target-id
```

The following example lists the newest backups of any dataset with `app-data` in the name on the *target-id* target that have the `cloudsnap0` source:

```
GET /api/service/v2/services/cloud/backups?dataset=app-data&target=target-
id&source=cloudsnap0
```

Delete a Cloud Backup

Before you delete a cloud backup, perform the following checks:

- Check whether a restore of this backup is in progress. See List Jobs.
- Determine whether this backup has children. Children of this backup have a `parent` value that is the same as the `dataset` value of the backup that you want to delete. See List Cloud Backups. To delete a backup that has children backups, specify the `force` option.

The following example submits a job request to delete the specified backup on the specified target from Oracle Cloud Infrastructure Object Storage. An appliance can operate on any backup on any target that the appliance can access, even if that backup was created on a different appliance.

Compare this operation with Delete a Snapshot Backup, which shows how to delete a share snapshot backup on the appliance.

Example Request:

In the following request, backup *backup-id* has no children:

```
DELETE /api/service/v2/services/cloud/backups/format/backup-id/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

In the following request, backup *backup-id* has children backups and the `force` option is applied:

```
DELETE /api/service/v2/services/cloud/backups/format/backup-id/target-id?force=true
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

```
{
    "action": "job-id"
}
```

To view the progress of the cloud backup deletion, use List Jobs to view the job with the above *job-id*.

Restore a Cloud Backup

A cloud backup can be restored on any appliance that has access to that cloud target. The backup can be restored as a newly cloned share on Oracle ZFS Storage Appliance, or the backup can be restored into an existing share.

When a cloud backup with the zfs format contains a snapshot with a retention hold, that hold is preserved when the cloud backup is restored.

The following example submits a job request to restore the specified backup. Specify the pool and project where you want the backup restored. If you want to restore to a new share, specify a name for the new share. If you want to restore to an existing share, specify the name of the share, and specify `useshare` as `true`. By default, the value of `useshare` is `false`.

To view the progress of the restoration, use List Jobs to view the job with the returned *job-id*.

If the `readlimit` property is set for the target, no more than `readlimit` bytes per second are read from the target. See Create a Target.

Example Request:

In the following request, backup `restore6` is a new share:

```
POST /api/service/v2/services/cloud/backups/format/backup-id/target-id/restore HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Length: 55

{
    "pool":"p1",
    "project":"default",
    "share": "restore6"
}
```

In the following request, backup `restore6` is an existing share on Oracle ZFS Storage Appliance, and the `useshare` flag is set to `true`:

```
POST /api/service/v2/services/cloud/backups/format/backup-id/target-id/restore HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

```
Content-Length: 55

{
    "pool":"p1",
    "project":"default",
    "share": "restore6",
    "useshare": true
}
```

Example Result:

```
{
    "action": "job-id"
}
```

List Jobs

The following example lists all running jobs and recently completed jobs, with the newest job listed first.

Example Request:

```
GET /api/service/v2/services/cloud/jobs HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

In this example, two backup jobs are saving two different backups to the same target.

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 21:37:52 GMT
Content-Length: 983
X-Zfssa-Service-Api: 2.0
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8

{
    "jobs": [{
        "href": "/api/service/v2/services/cloud/jobs/job-id2",
        "op": "backup",
        "target": "target-id",
        "targetName": "oci-ashburn",
        "created": "2020-02-06T16:52:42Z",
        "updated": "2020-02-06T16:52:48Z",
        "id": "job-id2",
        "status": "in-progress",
        "rate": 10002432,
        "transferred": 80019456,
        "estimated_size": 43088792088,
        "dataset": "p1/local/default/f-1",
        "backup": "backup-id2",
        "snapshot": "snap3",
        "format": "tar",
        "details": "uploading backup to zfs/backups/tar/backup-id2/000000001"
    },{
        "href": "/api/service/v2/services/cloud/jobs/job-id1",
        "op": "backup",
        "target": "target-id",
        "targetName": "oci-ashburn",
        "created": "2020-02-06T16:52:28Z",
        "updated": "2020-02-06T16:52:48Z",
        "id": "job-id1",
        "status": "in-progress",
        "rate": 1942,
        "transferred": 3884,
        "estimated_size": 0,
        "dataset": "p1/local/default/f-1",
        "backup": "backup-id1",
        "snapshot": "snap2",
        "format": "zfs",
        "details": "uploading backup to zfs/backups/zfs/backup-id1/000000001"
    }]
}
```

The following example lists the specified job:

```
GET /api/service/v2/services/cloud/jobs/job-id2 HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

List Cloud Backup Jobs Query Parameters

The `start` and `limit` query parameters are supported for filtering the list of cloud backup jobs.

- The value of the `start` parameter is the job index. Values for this index are from 0 to one less than the value of the `entries` property in the `jobs` section of the request shown in List Cloud Service Properties.
- The value of the `limit` parameter is the maximum number of jobs to list.

Both parameters can be used in the same query. Use an AND operation (&) between the query parameters as shown in the following examples.

The following example lists the 100 newest running jobs and recently completed jobs, with the newest job listed first.

```
GET /api/service/v2/services/cloud/jobs?limit=100
```

The following example lists only the oldest running or recently completed job.

```
GET /api/service/v2/services/cloud/jobs?start=0&limit=1
```

The following example lists all running jobs and recently completed jobs with a job index number equal to or greater than 4, with the newest job listed first.

```
GET /api/service/v2/services/cloud/jobs?start=4
```

Cancel or Restart a Job

The following example cancels the specified cloud service job.

Example Request:

```
PUT /api/service/v2/services/cloud/jobs/job-id/cancel HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

```
HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2019 21:50:29 GMT
Content-Length: 0
X-Zfssa-Service-Api: 2.0
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8
```

The following example restarts the specified cloud service job.

Example Request:

```
PUT /api/service/v2/services/cloud/jobs/job-id/restart HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
```

Example Result:

```
HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2019 21:51:08 GMT
Content-Length: 0
X-Zfssa-Service-Api: 2.0
X-Zfssa-Api-Version: 2.0
Content-Type: application/json; charset=utf-8
```

Snapshot Backup Operations

Use snapshot backup operations to manage share snapshot backups on Oracle Cloud Infrastructure Object Storage. To create a snapshot, see Snapshot and Clone Operations.

Snapshots that have the same name on different local systems can be backed up to the same cloud target because each snapshot backup is assigned a unique identifier. The same filesystem snapshot can be used for two cloud backups in two different formats.

You can delete a local snapshot after you have backed up that snapshot to the cloud. However, retain local snapshots that could be parent snapshots for future incremental snapshots. If the local snapshot has a retention hold, the snapshot cannot be deleted until the retention hold is released.

Table 9-3 Snapshot Backup Commands

Request	Append to Path: /api/storage/v2/pools/*pool*/projects/*project* Plus one of the following: /filesystems/*fs* or /luns/*lun*	Description
GET	/snapshots/*snapshot*/backups	List all snapshot backups of any format

Table 9-3 (Cont.) Snapshot Backup Commands

Request	Append to Path: /api/storage/v2/pools/*pool*/projects/*project* Plus one of the following: /filesystems/*fs* or /luns/*lun*	Description
GET	/snapshots/*snapshot*/backups/*format*/*backup-id*/*target-id*	List the specified snapshot backup
DELETE	/snapshots/*snapshot*/backups/*format*/*backup-id*/*target-id*	Delete the specified snapshot backup
POST	/snapshots/*snapshot*/backups	Create a new snapshot backup

List Snapshot Backups

The following example lists all cloud backups of any format of snapshot `snap0` on filesystem `f-1`.

Example Request:

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/backups
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.54.0
Accept: */*
```

Example Result:

```
HTTP/1.1 200 OK
Date: Wed, 07 Jan 2020 20:54:47 GMT
Content-Length: 708
X-Zfssa-Storage-Api: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0

{
    "backups": [{
        "finished": "2020-01-07T21:02:14Z",
        "format": "tar",
        "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/
snap0/backups/tar/backup-id/target-id",
        "id": "backup-id",
        "status": "completed",
        "target": "target-id",
        "targetName": "oci-ashburn"
    }]
}
```

The following request lists the specified snapshot backup.

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/
backups/format/backup-id/target-id HTTP/1.1
```

Create a Snapshot Backup

The following example creates a backup of snapshot `snap0` in `tar` format and stores the backup on target `oci-phoenix`. The backup is created in `zfs` format if the format is not specified. To view the progress of the snapshot backup, use List Jobs to view the job with the returned *job-id*.

If `snap0` has a parent backup, and that parent backup does not exist in the cloud, then specify `require_parent_exists` as `false`. By default, `require_parent_exists` is `true`, and the parent backup must exist in the cloud.

If the `writelimit` property is set for the target, no more than `writelimit` bytes per second are written to the target. See Create a Target.

When a snapshot backup with the `zfs` format has a retention hold, that hold is preserved when the cloud backup is later restored.

Example Request:

In the following request, snapshot `snap0` either has no parent, or its parent already exists on `oci-phoenix` in `tar` format:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/
backups HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
{
    "target": "oci-phoenix",
    "format": "tar"
}
```

In the following request, you want to create the backup even though the parent of snapshot `snap0` does not exist on the target. The `require_parent_exists` flag is set to `false`:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/
backups HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
{
    "target": "oci-phoenix",
    "format": "tar",
    "require_parent_exists": false
}
```

Example Result:

```
{
    "action": "job-id"
}
```

Create an Incremental Snapshot Backup

To create an incremental snapshot backup, specify `true` as the value for the `incremental` property and specify the parent snapshot to use in the comparison. In the following example, `snap0` is the parent of `snap1`.

- If `snap0` does not exist in the cloud, then specify `require_parent_exists` as `false`. By default, `require_parent_exists` is `true`, and the parent snapshot must exist in the cloud.
- If `snap0` does exist in the cloud, then `snap0` must exist both on the same local system and on the same cloud target as `snap1`. In the following example, `snap0` must exist in filesystem `f-1` on both the local system and the cloud target.
- The parent and incremental filesystem snapshots must be the same format: `zfs` or `tar`.
- When an incremental snapshot backup with the `zfs` format has a retention hold, that hold is preserved when the cloud backup is later restored.

The following request creates a backup of filesystem `f-1` that is the difference between `snap0` and the current state of filesystem `f-1`. The incremental snapshot backup, `snap1`, is stored on target `oci-ashburn`. The parent snapshot backup, `snap0`, already existed on target `oci-ashburn`. The incremental snapshot backup is in the same format as the parent snapshot backup.

Example Request:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap1/backups
HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*

{
    "target":  "oci-ashburn",
    "incremental": true,
    "parent": "snap0"
}
```

Example Result:

```
{
    "action": "job-id"
}
```

Find the Parents of an Incremental Snapshot Backup

The following example identifies the parents of the specified incremental snapshot backup on the specified target. In this example, the parents of the backup of snapshot `snap2` on target `oci-ashburn` are snapshots `snap1` and `snap0`. The result shows that this appliance has access to both `oci-ashburn` and `oci-phoenix` targets. No results are shown for `oci-phoenix` because results are requested for `oci-ashburn`.

Example Request:

```
POST /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/backups?
props=true HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*

{
    "target": "oci-ashburn"
}
```

Example Result:

```
HTTP/1.1 200 OK
Date: Wed, 22 Jan 2020 22:02:17 GMT
Content-Length: 316
X-Zfssa-Storage-Api: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0

{
    "props": [{
        "choices": [
            "oci-ashburn",
            "oci-phoenix"
        ],
        "data_type": "string",
        "label": "Backup target",
        "name": "target"
    },{
        "choices": [
            "zfs",
            "tar"
        ],
        "data_type": "string",
        "label": "format",
        "name": "format"
    },{
        "choices": [
            true,
            false
        ],
        "data_type": "boolean",
        "label": "Incremental",
        "name": "incremental"
    },{
        "choices": [
            "snap1",
            "snap0"
        ],
        "data_type": "string",
        "label": "Parent",
        "name": "parent"
    }]
}
```

The following example uses GET instead of POST as an alternative way to identify the parents of the specified incremental snapshot backup. In this form, you do not need to specify the target. The results show no parents of `snap2` on the `oci-phoenix` target, which means you cannot create an incremental backup of `snap2` on `oci-phoenix`. You could create a full backup of `snap2` on `oci-phoenix`.

Example Request:

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/
targets HTTP/1.1
```

```
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

```
HTTP/1.1 200 OK
Date: Wed, 07 Jan 2020 22:04:08 GMT
Content-Length: 329
X-Zfssa-Storage-Api: 2.0
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 2.0

{
    "targets": [{
        "format": "zfs",
        "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/
snap2/targets/zfs/target-id1",
        "id": "target-id1",
        "name": "oci-ashburn",
        "parents": [
            "snap0",
            "snap1"
        ]
    },{
        "format": "tar",
        "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/
snap2/targets/tar/target-id1",
        "id": "target-id1",
        "name": "oci-ashburn",
        "parents": [
            "snap0",
            "snap1"
        ]
    },{
        "id": "target-id2",
        "name": "oci-phoenix",
        "parents": [],
        "href": "/api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/
snap2/targets/target-id2"
    }]
}
```

Use the following request to show the parents of a specified snapshot backup:

```
GET /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap2/targets/
format/target-id1 HTTP/1.1
```

Delete a Snapshot Backup

The following example removes the specified snapshot backup. To view the progress of the backup removal, use List Jobs to view the job with the returned *job-id*.

Example Request:

```
DELETE /api/storage/v2/pools/p1/projects/default/filesystems/f-1/snapshots/snap0/
backups/format/backup-id/target-id HTTP/1.1
Host: hostname:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
```

Example Result:

```
{
    "action": "job-id"
}
```

10
RESTful API Problem Service

The RESTful API Problem service is used to view and manage problems discovered by the Oracle ZFS Storage Appliance fault manager.

Repair Problem

The repair problem command marks a problem as repaired. The repair problem command uses the `uuid` input parameter, which is the UUID of the problem to be marked repaired.

Example Request:

```
PUT /api/problem/v1/problems/0d30be41-b50d-4d03-ddb4-edb69ee080f8/repaired
    HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful Response Returns HTTP status 202 (Accepted):

```
HTTP/1.1 202 Accepted
```

Problem Service Commands

Table 10-1 Problem Service Commands

Request	Append to Path /problem/v{1|2}	Description
GET	Use only /problem/v{1|2}	List the problem service commands
GET	/problems	List all current problems
GET	/problems/*problem*	Get detail properties for a problem with the specified uuid
PUT	/problems/*problem*/markrepaired	Mark the specified problem uuid as repaired
GET	/suspend_notification	Show whether notifications are suspended
PUT	/suspend_notification/enable	Suspend notifications
PUT	/suspend_notification/disable	Resume notifications

List Problems

This command lists all problems that are currently active on an Oracle ZFS Storage Appliance system. HTTP status of 200 (OK) is returned for a successful command.

Example Request:

```
GET /api/problem/v1/problems HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "problems": [{
        "code": "AK-8003-Y6",
            "description": "The device configuration for JBOD
                       '1204FMD063' is invalid.",
            "impact": "The disks contained within the enclosure
                   cannot be used as part of a storage pool.",
        "uuid": "0d30be41-b50d-4d03-ddb4-edb69ee080f8",
        "repairable": false,
        "type": "Defect",
            "timestamp": "2013-2-21 17:37:12",
            "severity": "Major",
        "components": [{
                "certainty": 100,
                    "status": "degraded",
                    "uuid": "b4fd328f-92d6-4f0e-fb86-e3967a5473e7",
                "chassis": "1204FMD063",
                "label": "hc://:chassis-mfg=SUN
                    :chassis-name=SUN-Storage-J4410
                    :chassis-part=unknown
                    :chassis-serial=1204FMD063
                                :fru-serial=1204FMD063
                    :fru-part=7041262
                                :fru-revision=3529/ses-enclosure=0",
                "revision": "3529",
                    "part": "7041262",
                    "model": "Sun Disk Shelf (SAS-2)",
                    "serial": "1204FMD063",
                    "manufacturer": "Sun Microsystems, Inc."
        }]
    }]
}
```

List Problem

The list problem command lists a single problem. HTTP status of 200 (OK) is returned for a successful command.

The list problem command uses the `uuid` input parameter, which is the UUID of a single problem.

Example Request:

```
GET /api/problem/v1.0/problems/0d30be41-b50d-4d03-ddb4-edb69ee080f8
    HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "problem": {
        "uuid": "0d30be41-b50d-4d03-ddb4-edb69ee080f8",
        ...
    }
}
```

Suspend Problem Notification

Servicing the appliance can generate false failures. To avoid sending unwanted Service Requests (SRs), you can suspend all notifications during the period when you are performing the service.

For a description of what happens when problem notification is suspended, see Suspending Problem Notification in *Oracle ZFS Storage Appliance Customer Service Manual, Release OS8.8.x*.

Show Status of Notification Suspension

Use the following command to check whether problem notifications are suspended.

Example Request:

```
GET /api/problem/v2/suspend_notification HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

The `period` property shows the number of minutes remaining in the notification suspension.

```
{
    "suspend_notification": {
        "href": "/api/problem/v2/suspend_notification",
        "suspend_notification": "enabled",
        "period": 472
    }
}
```

Suspend Notifications

Use the following command to suspend problem notifications. A successful operation returns HTTP Code 202 (Accepted).

Example Request:

```
PUT /api/problem/v2/suspend_notification/enable HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Resume Notifications

Use the following command to resume problem notifications. A successful operation returns HTTP Code 202 (Accepted).

Example Request:

```
PUT /api/problem/v2/suspend_notification/disable HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

11

RESTful API SAN Service

The RESTful API SAN service lets you connect Oracle ZFS Storage Appliance to your Storage Area Network (SAN).

SAN Overview

A SAN has the following basic components:

- A client that accesses network storage
- An Oracle ZFS Storage Appliance system that provides network storage
- A network that links the client to the storage

These three components remain the same regardless of which protocol is used on the network. In some cases, the network might even be a cable between the initiator and the target, but in most cases some type of switching is involved. The RESTful API SAN service manages four types of SAN resources for each supported protocol:

- **Initiators** – An application or production system end-point that is capable of initiating a SCSI session and sending SCSI commands and I/O requests. Initiators are also identified by unique addressing methods.
- **Initiator groups** – A set of initiators. When an initiator group is associated with a Logical Unit Numbers (LUNs), only initiators from that group can access the LUN.
- **Targets** – A storage system end-point that provides a service of processing SCSI commands and I/O requests from an initiator. A target is created by the storage system administrator, and is identified by unique addressing methods. A target, once configured, consists of zero or more logical units.
- **Target groups** – A set of targets. LUNs are exported over all the targets in one specific target group.

SAN Initiators

The following commands are used to manage SAN initiators.

These commands use the following URI parameters:

protocol
NAS protocol for the initiator: `fc`, `iscsi`, or `srp`

initiator
IQN, WWN, or EUI of the initiator

Table 11-1 Initiator Commands

Request	Append to Path /san/v{1|2}	Description
GET	/*protocol*/initiators	List all SAN initiators for the given protocol: `fc`, `iscsi`, `srp` objects
GET	/*protocol*/initiators/*initiator*	Get the specified SAN initiator for the given protocol: `fc`, `iscsi`, `srp` properties
POST	/*protocol*/initiators	Create a new SAN initiator for the given protocol: `fc`, `iscsi`, `srp`
PUT	/*protocol*/initiators/*initiator*	Modify the specified SAN initiator for the given protocol: `fc`, `iscsi`, `srp` object
DELETE	/*protocol*/initiators/*initiator*	Destroy the specified initiator object

Many of the initiator commands use the properties listed in the following table as return values. The create and modify commands also use the properties as input values.

Table 11-2 Initiator Properties

Property	Protocol	Description
`alias`	all	Alias for this initiator
`initiator`	`fc`	Port world wide name for this initiator (WWN)
`iqn`	`iscsi`	iSCSI qualified name for this initiator
`chapuser`	`iscsi`	Challenge handshake `auth` protocol (CHAP) user name
`chapsecret`	`iscsi`	Challenge handshake `auth` protocol (CHAP) secret
`initiator`	`srp`	Extended Unique Identifier (EUI)

List Initiators

Lists all of the initiators configured on the appliance of a specified protocol type. The response body contains an array of initiator properties named "initiators" in JSON format.

Example Request to List iSCSI Initiators:

```
GET /api/san/v1/iscsi/initiators HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "initiators": [{
        "alias": "init-02",
        "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02",
        "initiator": "iqn.zfs-storage.example.com.sun:02:02",
        "chapsecret": "",
        "chapuser": ""
    },{
        "alias": "init-01",
        "initiator": "iqn.zfs-storage.example.com.sun:02:01",
        "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01",
        "chapsecret": "",
        "chapuser": ""
    }]
}
```

Get Initiator Details

Lists the details of a single iSCSI initiator. The response body contains iSCSI initiator properties as an object named "initiator" in JSON format.

Example Request:

```
GET /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "initiator": {
        "alias": "init-01",
        "href": "/api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:01"
        "initiator": "iqn.zfs-storage.example.com.sun:02:01",
        "chapsecret": "",
        "chapuser": ""
    }
}
```

Create an Initiator

Creates a new iSCSI initiator. You must supply the iSCSI Qualified Name (IQN). The request body contains the iSCSI initiator properties in JSON format. The response includes the location URI of the new iSCSI initiator in the HTTP header and status Code 201 (Created) on success. The response body contains iSCSI initiator properties as an object named "initiator" in JSON format.

Example Request:

```
POST /api/san/v1.0/iscsi/initiators HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "initiator": "iqn.zfs-storage.example.com.sun:02:02",
    "alias":"init-02"
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 181
X-Zfssa-San-Api: 1.0
Location: /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:02:02

{
    "initiator": {
        "alias": "init-02",
        "href": "/api/san/v1/iscsi/initiators/iqn.zfs-
storage.example.com.sun:02:02",
        "initiator": "iqn.zfs-storage.example.com.sun:02:02",
        "chapsecret": "",
        "chapuser": ""
    }
}
```

Modify an Initiator

This command modifies an existing initiator. The request body contains the initiator properties that should be modified in JSON format. The IQN for the initiator is supplied in the URI. HTTP status 202 (Accepted) is returned on success. The response body contains new iSCSI initiator properties as an object named `initiator` in JSON format.

Example Request:

```
PUT /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "alias":"init-01-secure",
    "chapuser":"admin4",
    "chapsecret":"secret"
}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Length: 167
Content-Type: application/json
X-Zfs-Sa-Nas-Api: 1.0

{
    "initiator": {
        "alias": "init-01-secure",
        "href": "/api/san/v1/iscsi/initiators/iqn.zfs-
storage.example.com.sun:01",
        "iqn": "iqn.zfs-storage.example.com.sun:1",
        "chapsecret": "secret",
        "chapuser": "admin4"
    }
}
```

Delete an Initiator

Removes an initiator from Oracle ZFS Storage Appliance.

Example Request:

```
DELETE /api/san/v1/iscsi/initiators/iqn.zfs-storage.example.com.sun:01 HTTP/1.1
Host: zfs-storage.example.com:215
```

Successful Delete returns HTTP Code 204 (No Content):

```
HTTP/1.1 204 No-Content
```

Initiator Groups

The iSCSI initiator commands are used to manage iSCSI initiators and iSCSI initiator groups on an Oracle ZFS Storage Appliance system. The available commands are listed in the table below.

These commands use the following URI parameters:

protocol
NAS protocol for the initiator: `fc`, `iscsi`, or `srp`

name
Name of the initiator group

Each initiator group has a `name` property and an `initiators` property that contains a list of initiators in the initiator group.

Table 11-3 Initiator Group Commands

Request	Append to Path /san/v{1|2}	Description
GET	/*protocol*/initiator-groups	List all SAN initiator groups for the given protocol: `fc`, `iscsi`, `srp` objects
GET	/*protocol*/initiator-groups/*name*	Get the specified SAN initiator group for the given protocol: `fc`, `iscsi`, `srp` properties
POST	/*protocol*/initiator-groups	Create a new SAN initiator group for the given protocol: `fc`, `iscsi`, `srp`
PUT	/*protocol*/initiator-groups/*name*	Modify the specified SAN initiator group for the given protocol: `fc`, `iscsi`, `srp` object
DELETE	/*protocol*/initiator-groups/*name*	Destroy the specified name object

List Initiator Groups

Lists all available iSCSI initiator groups. On success HTTP status 200 (OK) is returned and the body contains a JSON object with a property named "groups" that contains an array of initiator group objects.

Example Request:

```
GET /api/san/v1/iscsi/initiator-groups HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "groups": [{
        "href": "/san/v1/iscsi/initiator-groups/p1-initiators-0",
        "initiators": ["iqn.zfs-storage.example.com.sun:0"],
        "name": "p1-initiators-0"
    }, {
        "href": "/san/v1/iscsi/initiator-groups/p1-initiators-1",
        "initiators": ["iqn.zfs-storage.example.com.sun:1"],
        "name": "p1-initiators-1"
    }]
}
```

Get Initiator Group Details

Gets detailed information from a single iSCSI initiator group. The group can be accessed by following the `href` property returned in the list initiator group command.

Example Request:

```
GET /api/san/v1/iscsi/initiator-groups/test-group HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "group": {
        "href": "/api/san/v1/iscsi/initiator-groups/test-group"
        "initiators": ["iqn.zfs-storage.example.com.sun:02:01"],
        "name": "test-group"
    }
}
```

Create an Initiator Group

Creates an iSCSI initiator group with no members. The request body contains a JSON object with a single `name` parameter containing the group name.

Table 11-4 Initiator Group Create Properties

Property	Type	Description
`name`	string	The name of the initiator group
`initiators`	array	An array of existing initiator IQN properties

Example Request:

```
POST /api/san/v1/iscsi/initiator-groups HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 64
Accept: application/json
```

```
{
   "name":"group-01",
   "initiators": ["iqn.zfs-storage.example.com.sun:02"]
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/san/v1/iscsi/initiator-groups/test-group

{
    "group": {
        "href": "/api/san/v1/iscsi/initiator-groups/test-group",
        "initiators": ["iqn.zfs-storage.example.com.sun:02"],
        "name": "group-01"
    }
}
```

Delete an Initiator Group

Removes an initiator group from Oracle ZFS Storage Appliance.

Example Request:

```
DELETE /api/san/v1.0/iscsi/initiator-groups/group-01 HTTP/1.1
Host: zfs-storage.example.com:215
```

Successful delete returns HTTP status 204 (No Content):

```
HTTP/1.1 204 No-Content
```

Targets

The iSCSI target commands are used to manage iSCSI targets and iSCSI target groups. The available commands are listed in the table below.

The targets commands take the following URI parameters:

protocol
SAN protocol: `fc`, `iscsi`, or `srp`

target
Target ID: IQN, WWN, or EUI

Table 11-5 Target Commands

Request	Append to Path /san/v{1|2}	Description
GET	/*protocol*/targets	List all SAN target for the given protocol: `fc`, `iscsi`, `srp` objects
GET	/*protocol*/targets/*target*	Get the specified SAN target for the given protocol: `fc`, `iscsi`, `srp` properties
POST	/*protocol*/targets	Create a new SAN target for the given protocol: `fc`, `iscsi`, `srp`

Table 11-5 (Cont.) Target Commands

Request	Append to Path /san/v{1|2}	Description
PUT	*/protocol*/targets/*target*	Modify the specified SAN target for the given protocol: `fc`, `iscsi`, `srp` object
DELETE	*/protocol*/targets/*target*	Destroy the specified target object

The get target commands return target properties. The create and modify target commands use the properties listed in the following table as input.

Table 11-6 Target Input Properties

Property	Protocol	Description
`alias`	iscsi	Simple human readable name
`iqn`	iscsi	The iSCSI qualified name
`state`	iscsi	State of the iSCSI target ("online", "offline")
`auth`	iscsi	Optional authentication type ("none", "chap")
`targetchapuser`	iscsi	Optional CHAP user authentication
`targetchapsecret`	iscsi	Optional CHAP secret authentication
`interfaces`	iscsi	List of network interfaces that target is available
`wwn`	fc	Worldwide name for this target
`port`	fc	Physical location of the port
`mode`	fc	Mode of this port (initiator or target)
`speed`	fc	Negotiated speed of this port
`discovered_ports`	fc	Number of discovered remote initiator ports
`alias`	srp	Alias for the SRP target
`eui`	srp	Extended unique identifier for this target

The following properties are used for getting iSCSI target group information.

Table 11-7 Target Group Properties

Property	Type	Description
`protocol`	string	The target group protocol: FC, iSCSI, or SRP
`name`	string	The iSCSI target group name
`targets`	array	A list of iSCSI target IQN group members

List Targets

Lists all of the SAN targets of the specified protocol available on Oracle ZFS Storage Appliance.

Example Request:

```
GET /api/san/v1/iscsi/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1337

{
    "size": 7,
     "targets": [{
        "alias": "tst.volumes.py.12866.target",
        "href": "/api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:
            72b6fa9a-96c4-e511-db19-aadb9bac2052",
        "iqn": "iqn.zfs-storage.example.com.sun:02:72b6fa9a-96c4-
            e511-db19-aadb9bac2052",
        ...
    }, {
        "alias": "tst.volumes.py.96238.target",
         "href": "/api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:
            31d26d2e-6aa0-6054-fe58-8b1fb508b008",
        "iqn": "iqn.zfs-storage.example.com.sun:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
        ...
    }
    ...]
}
```

Get Target Details

Gets properties from a single target. The target can be selected by using the "iqn" property or by using "alias=*alias*".

Example Request:

```
GET /api/san/v1/iscsi/targets/alias=test-target HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 251

{
    "target": {
        "alias": "test-target",
        "auth": "none",
        "href": "/api/san/v1/iscsi/targets/alias=test-target",
        "interfaces": ["ixgbe0"],
        "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-
fe58-8b1fb508b008",
        "targetchapsecret": "",
        "targetchapuser": ""
    }
}
```

Create a Target

Creates a new target. The request body has a JSON object with a single `name` property that is the name of the new iSCSI target group.

Example Request:

```
POST /api/san/v1/iscsi/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 23
Accept: application/json

{"alias": "test-target"}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 233
X-Zfssa-San-Api: 1.0
Location: /api/san/v1/iscsi/targets/iqn.zfs-
storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008

{
    "target": {
        "href": "/api/san/v1/iscsi/targets/iqn.zfs-
storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008",
        "alias": "test-target",
        "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-
fe58-8b1fb508b008",
        "auth": "none",
        "targetchapuser": "",
        "targetchapsecret": "",
        "interfaces": ["ixgbe0"]
    }
}
```

Modify a Target

Modifies an existing iSCSI target. The request body contains a JSON object that contains the iSCSI target properties that are modified. HTTP status 202 (Accepted) is returned on success. The response body contains the resulting iSCSI target properties for the target encoded in a JSON object.

Example Request:

```
PUT /api/san/v1/iscsi/targets/alias=test-target HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 54
Accept: application/json

{"targetchapsecret":"secret", "auth":"chap",
 "targetchapuser":"admin5"}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 189
X-Zfssa-San-Api: 1.0

{
    "target": {
        "href": "/api/san/v1/iscsi/targets/alias=test-target",
        "auth": "chap",
        "targetchapsecret": "secret",
        "alias": "test-target",
        "iqn": "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-
fe58-8b1fb508b008",
        "targetchapuser": "admin5",
        "interfaces": ["ixgbe0"]
    }
}
```

Delete a Target

Removes a SAN target from the Oracle ZFS Storage Appliance system.

Example Request:

```
DELETE /api/san/v1/iscsi/targets/iqn.zfs-storage.example.com.sun:02:e7e688b1 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Successful Delete returns HTTP code 204 (No Content):

```
HTTP/1.1 204 No-Content
```

Target Groups

Target groups are collections of targets. The target groups commands are listed in the following table.

The target groups commands take the following URI parameters:

protocol
NAS protocol for the initiator: `fc`, `iscsi`, or `srp`

target-group
Name of the target group

Table 11-8 Target Groups Commands

Request	Append to Path /san/v{1|2}	Description
GET	/*protocol*/target-groups	List all SAN target group for the given protocol: `fc`, `iscsi`, or `srp` objects
GET	/*protocol*/target-groups/*target-group*	Get the specified SAN target group for the given protocol: `fc`, `iscsi`, or `srp` properties
POST	/*protocol*/target-groups	Create a new SAN target group for the given protocol: `fc`, `iscsi`, or `srp`
PUT	/*protocol*/target-groups/*target-group*	Modify the specified SAN target group for the given protocol: `fc`, `iscsi`, or `srp` object
DELETE	/*protocol*/target-groups/*target-group*	Destroy the specified target-group object

List Target Groups

Lists all of the target groups available for an Oracle ZFS Storage Appliance system. On success, HTTP status 200 (OK) is returned and the body contains a JSON object with a property named `groups` that contains an array of target group objects.

Example Request:

```
GET /api/san/v1/iscsi/target-groups
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 237

{
    "groups": [{
        "href": "/api/san/v1/iscsi/target-groups/test-group",
        "name": "test-group",
        "targets": [
            "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-
fe58-8b1fb508b008"
        ]
    }, {
        "href": "/api/san/v1/iscsi/target-groups/alt-group",
        ...
    }]
}
```

Get Target Group

Gets a single target group. The request takes a single URI parameter, which is the target group name. The response body contains a JSON object property named `group` that contains the target group properties.

Example Request:

```
GET /api/san/v1/iscsi/target-groups/test-group
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "group": {
        "href": "/api/san/v1/iscsi/target-groups/test-group",
        "name": "test-group",
        "targets": [
            "iqn.zfs-storage.example.com.sun:02:0d5a0ed8-44b6-49f8-a594-872bf787ca5a"]
    }
}
```

Create a Target Group

Creates a new iSCSI target group. The request body is a JSON object with a single `name` property that is the name of the new group.

Example Request:

```
POST /api/san/v1/iscsi/target-groups HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
Content-Type: application/json
Content-Length: 97

{"name":"test-group",
 "targets": ["iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-
fe58-8b1fb508b008"]}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 154
X-Zfssa-San-Api: 1.0
Location: /api/san/v1/iscsi/target-groups/test-group

{
    "group": {
        "href": "/api/san/v1/iscsi/target-groups/test-group",
        "name": "test-group",
        "targets": [
            "iqn.zfs-storage.example.com.sun:02:31d26d2e-6aa0-6054-fe58-8b1fb508b008"]
    }
}
```

Delete a Target Group

Deletes an existing target group.

Example Request:

```
DELETE /api/san/v1.0/iscsi/target-groups/test-group
```

Successful delete returns HTTP status 204 (No Content):

```
HTTP/1.1 204 No-Content
```

12 Service Commands

The Service RESTful API is used to list and manage software services running on Oracle ZFS Storage Appliance.

Service Commands

The following service commands are available.

Table 12-1 Service Commands

Request	Append to Path /service/v{1|2}	Description
GET	Use only /service/v{1|2}	List service commands
GET	/services	List all services
GET	/services/*service*	Get configuration and status for the specified service
PUT	/services/*service*	Modify the configuration of the specified service
PUT	/services/*service*/enable	Enable the specified service
PUT	/services/*service*/disable	Disable the specified service

List Services

This command returns the list of configurable services available on Oracle ZFS Storage Appliance along with their enabled status. HTTP status 200 (OK) is returned for a successful command.

Example Request:

```
GET /api/service/v1/services HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result. For brevity, most entries are omitted:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
X-Zfssa-Service-Api: 1.0

{
  "services": [
    {
      "<status>": "disabled",
      "href": "/api/service/v1/services/ad",
      "name": "ad",
      "log": {
        "href": "/api/log/v1/logs/appliance-kit-adstat:default",
        "size": 2
      }
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/nfs",
      "name": "nfs",
      "log": {
        "href": "/api/log/v1/logs/appliance-kit-nfsconf:default",
        "size": 8
      }
    },
    {
      "<status>": "online",
      "href": "/api/service/v1/services/ssh",
      "name": "ssh",
      "log": {
        "href": "/api/log/v1/logs/network-ssh:default",
        "size": 134
      }
    }
  ]
}
```

Get Service

This command gets the details from a single service, including its state and its configuration.

Example Request:

```
GET /api/service/v1/services/ndmp HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "service": {
        "cram_md5_password": "",
        "cram_md5_username": "",
        "dar_support": true,
        "default_pools": [],
        "drive_type": "sysv",
        "href": "/api/service/v1/services/ndmp",
        "ignore_ctime": false,
        "name": "ndmp",
        "restore_fullpath": false,
        "status": "online",
        "tcp_port": 10000,
        "version": 4,
        "zfs_force_override": "off",
        "zfs_token_support": false
    }
}
```

Change Service State

This command changes the state of a given service. The following URI parameters are used:

- *service* - Name of the service
- *state* - New service state: `enable` or `disable`

Example Request:

```
PUT /api/service/v1/services/replication/enable HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful response returns HTTP status 202 (Accepted). The service can also be enabled or disabled by sending a JSON request to the service.

Example Request Using JSON:

```
PUT /api/service/v1/services/replication HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 22

{"<status>": "enable"}
```

To disable the service send the following JSON:

```
{"<status>": "disable"}
```

Modify Service Configuration

Configuration properties on a specified service can be modified by sending a `PUT` request with the new property values defined in the header. Some services may have sub-resources, and they can also be modified by following the href defined in the sub-resource. Successful response returns HTTP status of 202 (Accepted).

> **Note:**
>
> To automatically configure the LDAP service for the currently joined AD domain, append `/ldap?confirm=true` to path `services/ad`. To view the settings after they are configured, use the `GET` request and specify the LDAP service. See Get Service. To customize the settings, use the `PUT` request and specify the LDAP service. See the following example.

The following example re-orders the list of servers for the LDAP service and specifies that the list of servers is in order of preference. For more information about LDAP configuration, see LDAP Configuration in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Current State:

```
GET /api/service/v1/services/ldap HTTP/1.1
Host: zfs-storage.example.com:215
```

```
Accept: application/json

HTTP/1.1 200 OK
Content-Type: application/json

{
   "service":{
      "href":"/api/service/v1/services/ldap",
      "<status>":"online",
      "servers":[
         "ldap-server2.us.example.com:484",
         "ldap-server1.us.example.com:636"
      ],
      "use_server_order":false,
      "proxy_dn":"",
      "proxy_password":false,
      "base_dn":"dc=us,dc=oracle,dc=com",
      "search_scope":"one",
      "cred_level":"proxy",
      "auth_method":"simple",
      "use_tls":false,
      "user_search":[
      ],
      "user_mapattr":[
      ],
      "user_mapobjclass":[
      ],
      "group_search":[
      ],
      "group_mapattr":[
      ],
      "group_mapobjclass":[
      ],
      "netgroup_search":[
      ],
      "netgroup_mapattr":[
      ],
      "netgroup_mapobjclass":[
      ],
      "server-000":{
         "host":"ldap-server2.us.example.com",
         "port":484,
         "status": "online",
         "last_seen": "142s",
         "rtt": "70.285ms",
         "err_msg": "",
         "href":"/api/service/v1/services/ldap/server-000"
      },
      "server-001":{
         "host":"ldap-server1.us.example.com",
         "port":636,
         "status": "online",
         "last_seen": "142s",
         "rtt": "126.013ms",
         "err_msg": "",
         "href":"/api/service/v1/services/ldap/server-001"
      }
   }
}
```

Example Request:

```
PUT api/service/v1/services/ldap HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json

{
   "servers": ["ldap-server1.us.example.com:636,ldap-server2.us.example.com:484"],
   "use_server_order":true
}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Length: 1295
Content-Type: application/json; charset=utf-8
X-Zfssa-Service-Api: 1.0

{
   "service":{
      "href":"/api/service/v1/services/ldap",
      "<status>":"online",
      "servers":[
         "ldap-server1.us.example.com:636",
         "ldap-server2.us.example.com:484"
      ],
      "use_server_order":true,
      "proxy_dn":"",
      "proxy_password":false,
      "base_dn":"dc=us,dc=oracle,dc=com",
      "search_scope":"one",
      "cred_level":"proxy",
      "auth_method":"simple",
      "use_tls":false,
      "user_search":[
      ],
      "user_mapattr":[
      ],
      "user_mapobjclass":[
      ],
      "group_search":[
      ],
      "group_mapattr":[
      ],
      "group_mapobjclass":[
      ],
      "netgroup_search":[
      ],
      "netgroup_mapattr":[
      ],
      "netgroup_mapobjclass":[
      ],
      "server-000":{
         "host":"ldap-server1.us.example.com",
         "port":636,
         "status": "online",
         "last_seen": "142s",
         "rtt": "126.013ms",
         "err_msg": "",
         "href":"/api/service/v1/services/ldap/server-000"
      },
      "server-001":{
         "host":"ldap-server2.us.example.com",
         "port":484,
         "status": "online",
         "last_seen": "142s",
         "rtt": "70.285ms",
         "err_msg": "",
         "href":"/api/service/v1/services/ldap/server-001"
      }
   }
}
```

Service Resources

Some services have sub-resources. See the data returned for each service or the list of service commands to see what sub-resources are available.

Table 12-2 Service Sub Resource Commands

Request	Path	Description
GET	/services/*service/resource*	List service sub-resource
PUT	/services/*service/resource/href*	Modify sub-resource
POST	/services/*service/resource*	Create a new sub-resource
DELETE	/services/*service/resource/href*	Destroy an sub-resource

Each of these commands follow the same pattern as other RESTful API commands where `GET` is used to list or get a specified sub-resource type, `POST` is used to create a new sub-resource type, `PUT` is used to modify the sub-resource and `DELETE` is used to destroy the specified sub-resource.

For a list of sub-resources and properties and commands available for each sub-resource, see Configuring Services in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

13 RESTful API Storage Service

The RESTful API Storage service is used to view configuration and manage aspects of storage pools, projects, filesystems and LUNs. It also manages snapshots and replication.

Storage Pool Operations

For Oracle ZFS Storage Appliance, NAS is configured in pools that characterize the same data redundancy characteristics across all LUNs and filesystems, and pool operations are used to obtain the appliance storage configuration.

Table 13-1   Storage Pool Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/pools	List all storage pools
GET	/pools/*pool*	Get storage pool details
POST	/pools	Configure a new storage pool
PUT	/pools/*pool*	Add or remove storage from a pool
PUT	/pools/*pool*/edit	Set pool properties
GET	/import	List properties for all storage pools available for importing
POST	/import/*pool*?props=true	List properties for specified storage pool for importing
POST	/import/*pool*	Import a storage pool with a unique name
POST	/import/*guid*	Import a storage pool without a unique name
POST	/import/*new-pool-name guid*	Import a storage pool and rename it
PUT	/pools/*pool*/scrub	Start a data scrub on the specified pool
DELETE	/pools/*pool*/scrub	Stop any data scrub job on the specified pool
GET	/pools/*pool*/LBPthreshold	List the logical block provisioning (LBP) threshold limit for the specified pool
PUT	/pools/*pool*/LBPthreshold	Set the LBP threshold limit for the specified pool
DELETE	/pools/*pool*	Unconfigure the specified storage pool

List Pools

This command lists the properties of all storage pools on the Oracle ZFS Storage Appliance system. HTTP status 200 (OK) is returned for a successful command. The HTTP body contains a list of JSON objects describing each pool. The names of the properties are shown in the following table.

> **Note:**
>
> The `depth` query parameter and the `match_`*`property-name`*`=`*`value`* query parameter are not supported.

Table 13-2 Storage Pool Properties

Property	Type	Description
`asn`	string	Serial number of the Oracle ZFS Storage Appliance system that owns the pool
`name`	string	The target pool name
`owner`	string	Hostname of the system that owns the pool
`peer`	string	In a clustered system, the partner head of the appliance cluster
`profile`	string	Data device profile
`scrub_schedule`	string	Number of days between scheduled pool scrubbing operations, or disable scheduled pool scrubbing. See Pool Scrub for allowed values and more pool scrubbing properties.
`state`	string	Pool state: `online`, `offline`, `exported`

Example Request:

```
GET /api/storage/v1/pools HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "pools": [{
        "profile": "mirror3",
        "name": "m1",
        "peer": "peer-hostname",
        "state": "online",
        "owner": "system-hostname",
        "asn": "appliance-serial-number",
        "scrub_schedule": "30 days"
    }, {
        "profile": "raidz1",
        "name": "r1",
        "peer": "peer-hostname",
        "state": "online",
        "owner": "system-hostname",
        "asn": "appliance-serial-number",
        "scrub_schedule": "30 days"
    }]
}
```

Get Pool

This command returns the properties from a single storage pool, along with storage usage information for the pool. HTTP status 200 (OK) is returned for a successful command.

Example Request:

```
GET /api/storage/v1/pools/p1 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "pool": {
        "profile": "raidz1",
        "name": "p1",
        "usage": {
            "available": 57454799311352.0,
            "compression": 1.0,
            "dedupratio": 672791,
            "free": 57454799311352.0,
            "total": 74732430950400.0,
            "usage_child_reservation": 0.0,
            "usage_data": 16011663438848.0,
            "usage_metasize": 0.0,
            "usage_metaused": 0.0,
            "usage_replication": 1693675705344.0,
            "usage_reservation": 0.0,
            "usage_snapshots": 123913627136.0,
            "usage_total": 17829252771328.0,
            "used": 17829252771328.0
        },
        "peer": "00000000-0000-0000-0000-000000000000",
        "state": "online",
        "owner": "admin1",
        "guid": "e32b4cf1a6910baa",
        "asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749"
    }
}
```

Configure Pool

Configures a pool. For the parameters needed to create a pool, see the CLI configuration storage command. A dry run request to create a pool can be done that returns the available property names and values. This is done by setting the `props` query parameter properties to `true`.

Example Request:

```
POST /api/storage/v1/pools?props=true HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json

{
    "name": "p1"
}
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

"props": [{
        "choices": ["custom" ],
        "label": "Chassis 0",
        "name": "0",
        "type": "ChooseOne"
    }, {
        "choices": ["custom"],
        "label": "Chassis 1",
        "name": "1",
        "type": "ChooseOne"
    }, {
        "choices": [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
        "label": "Chassis 1 data",
        "name": "1-data",
        "type": "ChooseOne"
    }, {
        "choices": ["mirror", "mirror3", "raidz1",
            "raidz2", "raidz3_max", "stripe"],
        "label": "Data Profile",
        "name": "profile",
        "type": "ChooseOne"
    }]
}
```

Example Request (to create a pool that uses 8 disks from chassis [1]):

```
POST /api/storage/v1/pools HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json

{
    "name": "p1",
    "profile": "stripe",
    "1-data": 8
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json

{
    "pool": {
        "asn": "314d252e-c42b-e844-dab1-a3bca680b563",
        "errors": [],
        "name": "p1",
        "owner": "zfs-storage",
        "peer": "00000000-0000-0000-0000-000000000000",
        "profile": "stripe",
        "status": "online",
        "usage": {
            "available": 1194000466944.0,
            "dedupratio": 100,
            "total": 1194000908288.0,
            "used": 441344.0
        }
    }
}
```

Add Storage to a Pool

This command is similar to create or configure a pool. Add storage adds additional storage devices to an existing pool. Send href *pool*/`add` with the body containing the desired number of storage devices to add to the pool.

Example Request:

```
PUT /api/storage/v1/pools/p1/add HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json

{
    "2-data": 8
}
```

Example Result:

```
HTTP/1.1 202 Accepted
```

Remove Storage from a Pool

This command is similar to add storage to a pool. Remove storage removes cache and log storage devices from an existing pool. Send href *pool*/`remove` with the body containing the desired type, chassis number, and number of storage devices to remove from the pool.

Example Request:

```
PUT /api/storage/v1/pools/p1/remove HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json

{
    "0-cache" : 2
}
```

Example Result:

```
HTTP/1.1 202 Accepted
```

To display the number of devices that can be removed, set the `props` query parameter to `true`.

Example Request:

```
PUT /api/storage/v1/pools/p1/remove?props=true HTTP/1.1
Host: zfs-storage.example.com
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
      "props": [
          {
              "choices": [
                  "0",
                  "1",
                  "2"
              ],
              "type": "ChooseOne",
              "name": "0-cache",
              "label": "Chassis 0 cache"
          },
          {
              "choices": [
                  "0",
                  "1",
                  "2"
              ],
              "type": "ChooseOne",
              "name": "1-log",
              "label": "Chassis 1 log"
          }
      ]
}
```

Set Pool Properties

This command accesses the route to a specific storage pool so properties can be set. Select the pool to be modified and edit its properties. For CLI properties for unencrypted storage pools, see Configuring Storage in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*. For encrypted pool CLI properties, see Creating an Encrypted Pool (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
PUT /api/storage/v2/pools/p0/edit HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json

{
         "keystore": "LOCAL",
          "keyname": "MyKey",
   "scrub_schedule": "45 days"
}
```

Example Result:

```
HTTP/1.1 202 Accepted
```

Import Pool

This command imports a single storage pool. HTTP status 200 (OK) is returned for a successful command. Error code 400 is returned for an invalid argument or if an import operation is already in progress.

To view the available pools and their properties, use the following commands:

- **View All Pools' Properties:** Use either the `GET` import command or use the `POST` import command with the `props` query parameter set to `true`. The properties include the GUID.
- **View an Individual Pool's Properties:** Use the `POST` import command, specify the pool's name at the end of the route, and set the `props` parameter to `true`.

Set the appropriate arguments when importing a storage pool:

- **Unique Pool Name:** Specify the pool's name.
- **Pool Name Not Unique:** Specify the pool's GUID; the pool's name is not necessary.
- **Pool Name Change:** Specify both the new pool name and the old GUID. The new name must be 1 to 64 characters in length. The name cannot begin with a period (.) and cannot include spaces. Allowable characters are alphanumeric characters and special characters _ **- . :**

Example Request to List Properties for All Pools:

```
GET /api/storage/v2/import HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "pools": [
      {
        "guid": "599033b292ea0da1",
        "name": "p1",
        "state": "online",
        "profile": "raidz1",
        "disks": {
          "data": 4,
          "spare": 0,
          "log": 0,
          "cache": 0,
          "meta": 0
      }
    },
    {
        "guid": "600144c303fb1eb2",
        "name": "p2",
        "state": "online",
        "profile": "raidz1",
        "disks": {
          "data": 2,
          "spare": 0,
          "log": 0,
          "cache": 0,
          "meta": 0
      }
    }
  ]
}
```

Example Request to Import a Pool with a Unique Pool Name:

```
POST /api/storage/v2/import HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

{
    "name": "p1"
}
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "pool": {
        "name": "p1",
        "state": "online",
        "asn": "0433018a-6810-44ec-b740-8d49fff31118",
        "owner": "owner-name",
        "peer": "peer-hostname",
        "profile": "raidz1",
        "errors": [],
        "encryption": "off",
        "guid": "599033b292ea0da1",
        ...
    }
}
```

Example Request to Import a Pool without a Unique Pool Name:

```
POST /api/storage/v2/import HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

{
    "guid": "711255d414gc2fc3"
}
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "pool": {
        "name": "p1",
        "state": "online",
        "asn": "0433018a-6810-44ec-b740-8d49fff31118",
        "owner": "owner-name",
        "peer": "peer-hostname",
        "profile": "raidz1",
        "errors": [],
        "encryption": "off",
        "guid": "711255d414gc2fc3",
        ...
    }
}
```

Example Request to Import a Pool and Rename It:

```
POST /api/storage/v2/import HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

{
    "name": "p3",
    "guid": 599033b292ea0da1"
}
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "pool": {
        "name": "p3",
        "state": "online",
        "asn": "0433018a-6810-44ec-b740-8d49fff31118",
        "owner": "owner-name",
        "peer": "peer-hostname",
        "profile": "raidz1",
        "errors": [],
        "encryption": "off",
        "guid": "599033b292ea0da1",
        ...
    }
}
```

Pool Scrub

Sending a *pool*`/scrub PUT` request starts a pool scrub operation. Sending a *pool*/scrub `DELETE` request stops a running scrub operation. For more information about pool scrubbing, see Scrubbing a Storage Pool – Manual (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

By default, scheduled storage pool scrubbing is enabled and set to every 30 days. The `scrub_schedule` property specifies the number of days between scheduled pool scrubbing operations, or disables scheduled pool scrubbing. The default value of `scrub_schedule` is `30`.

- To disable scheduled scrubbing, for example if you prefer to perform manual scrubbing, set the value of the `scrub_schedule` property to `off`.
- To change the number of days between scheduled scrubbing operations, set the value of the `scrub_schedule` property to `15`, `30`, `45`, `60`, `75`, or `90`.

For more information about scheduled pool scrubbing, see Scrubbing a Storage Pool – Scheduled (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

The `scrub` object reports on the most recent pool scrub, either scheduled or manual.

- If the `complete` property is `false`, then the scrub is still running. The number of errors found (`errors`) and number of errors repaired (`repaired`) are shown.
- If the `complete` property is `true`, then the scrub has finished. Additional information shown includes the time the scrub started (`op_start`) and the time the scrub ended (`last_end`). Times are in GMT.

Note that `scrub_started` and `scrub_finished` are events of the `zfs_pool` alert action event category, and you could specify a custom action for those events. See RESTful API Alert Service.

The following example shows a partial pool list after a scrub.

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "pool": {
        "status": "online",
        "profile": "mirror:log_stripe:cache_stripe",
        "scrub": {
            "errors": 0,
            "op_start": "20190520T16:09:41",
            "complete": true,
            "seq_resilver": 0,
            "type": "everything",
            "examined": 403968,
            "repaired": 0,
            "last_end": "20190520T16:17:59"
        },
        "scrub_schedule": "30 days",
        "name": "p0",
        "peer": "peer-hostname",
        "owner": "system-hostname",
        "asn": "appliance-serial-number"
    }
}
```

List LBP Threshold

This command lists the logical block provisioning (LBP) threshold limit setting for thin provisioned LUNs within the specified storage pool on the Oracle ZFS Storage Appliance system. When the limit is set to the default value of 0, the LBP threshold is disabled.

Example Request:

```
GET /api/storage/v1/pools/p1/LBPthreshold HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
{
    "LBPthreshold": {
        "href": "/api/storage/v1/pools/p1/LBPthreshold",
        "limit": 100
    }
}
```

Set LBP Threshold

This command sets the logical block provisioning (LBP) threshold limit for thin provisioned LUNs within the specified storage pool on the Oracle ZFS Storage Appliance system.

To set the `threshold` parameter, specify the average storage consumption rate, which must be in the range of 1 to 100. When set to the default value of 0, the LBP threshold is disabled.

When the threshold value is exceeded, an error message is returned, and you can manage the space accordingly.

Example Error Message:

```
{  "fault": {
    "code": 413,
    "name": "ERR_OVER_LIMIT",
    "message": "input request too large to handle (threshold value out of range)"
  }
}
```

Example Request:

```
PUT /api/storage/v1/pools/p1/LBPthreshold?threshold=99 HTTP/1.1
Host: zfs-storage.example.com
Content-Type: application/json
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "LBPthreshold": {
        "href": "/api/storage/v1/pools/p1/LBPthreshold",
        "limit": 99
    }
}
```

Unconfigure Pool

This command removes a pool from the Oracle ZFS Storage Appliance system.

Request to Delete a Pool:

```
DELETE /api/storage/v1/pools/p1 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Example Result:

```
HTTP/1.0 204 No Content
Date: Fri, 02 Aug 2013 22:31:06 GMT
X-Zfssa-Nas-Api: 1.0
Content-Length: 0
```

Project Operations

All project operations can be scoped to a given pool. Commands that operate across all projects append `/projects` to the URI, and commands that operate on a single project append `/projects/`*project*.

Table 13-3 Project Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/projects	List all projects
GET	/pools/*pool*/projects	List projects
GET	/pools/*pool*/projects?snaps=true	List all projects, including snapshots
GET	/pools/*pool*/projects/*project*	Get project details
POST	/pools/*pool*/projects	Create a project
PUT	/pools/*pool*/projects/*project*	Modify a project
DELETE	/pools/*pool*/projects/*project*	Destroy a project
GET	/pools/*pool*/projects/*project*/usage/groups	Get project group usage
GET	/pools/*pool*/projects/*project*/usage/groups/*group*	Get project usage for the specified group
GET	/pools/*pool*/projects/*project*/usage/users	Get project user usage
GET	/pools/*pool*/projects/*project*/usage/users/*user*	Get project usage for the specified user

The following table shows the list of editable properties within a project resource.

Table 13-4 Project Properties

Property	Type	Description
`aclinherit`	string	ACL inheritance behavior ("discard", "noallow", "restricted", "passthrough", "passthrough-x", "passthrough-mode-preserve")
`aclmode`	string	ACL behavior on mode change ("discard", "mask", "passthrough")
`atime`	boolean	Update access time on read flag
`canonical_name`	string	Canonical name
`checksum`	string	Block checksum ("fletcher2", "fletcher4", "sha256")
`compression`	string	Data compression setting ("off", "lzjb", "gzip-2", "gzip", "gzip-9")
`copies`	number	Number of additional replication copies
`creation`	datetime	Date and time of project (or LUN, filesystem) creation
`dedup`	boolean	Data deduplication flag

Table 13-4 (Cont.) Project Properties

Property	Type	Description
default_group	string	Project default filesystem group: "other"
default_permissions	string	Project default filesystem permissions "700"
default_sparse	boolean	Project default LUN sparse data flag
default_user	string	Project default filesystem user: "nobody"
default_volblocksize	number	Project default LUN blocksize: 8192
default_volsize	number	Project default LUN Size
exported	boolean	Exported flag
logbias	string	Synchronous write bias ("latency", "throughput")
mountpoint	string	Share mountpoint default "/export/proj-01"
name	string	Project name
nbmand	boolean	Non-blocking mandatory locking flag
nodestroy	boolean	Prevent destruction flag
quota	number	Project quota size in bytes
origin	string	Clone origin
pool	string	Pool names
readonly	boolean	Data is read only if set to true
recordsize	string	Database record size "128k"
reservation	number	Data reservation size
retention_policy	string	File retention: File retention policy ("disabled", "mandatory", "privileged")
retention_policy_changeacl	boolean	File retention: Determines if a retained file's ACL/permissions can be changed ("off", "on")
retention_policy_onexpiry	string	File retention: Determines behavior when file retention expires ("off", "delete", "hold")
retention_period_deletegrace	number	File retention: Number of seconds/hours/days/years that automatic file deletion is delayed when retention_policy_onexpiry is set to "delete." Default value: 0 days.

Table 13-4 (Cont.) Project Properties

Property	Type	Description
retention_period_default	number	File retention: Default amount of time for which a file is retained if it is automatically retained, or retained manually without first changing the file's access time attribute. Must be set if retention_policy is not "disabled."
retention_period_min	number	File retention: Minimum amount of time for file retention. Must be set if retention_policy is not "disabled."
retention_period_max	number	File retention: Maximum amount of time for file retention. Must be set if retention_policy is not "disabled."
retention_period_grace	number	File retention: Amount of time a file must remain unmodified before it is automatically retained at the default file retention period value
retention_status_expiry	string	File retention: Expiration date and time for a file
retention_status_files	string	File retention: File status for expiration date, time, and if expired
rstchown	boolean	Restrict ownership change flag
secondarycache	string	Secondary cache usage ("all", "metadata", "none")
sharedav	string	HTTP share ("off", "rw", "ro")
shareftp	string	FTP share ("off", "rw", "ro")
sharenfs	string	NFS share ("off", "on", "ro", "rw")
sharesftp	string	SFTP share ("off", "rw", "ro")
sharesmb	string	SMB/CIFS share ("off", "rw", "ro")
sharetftp	string	TFTP share ("off", "rw", "ro")
snapdir	string	.zfs/snapshot visibility ("hidden", "visible")
snaplabel	string	Scheduled snapshot label
vscan	boolean	Virus scan flag

List Projects

This command lists all of the projects in a given pool. The request takes a single URI parameter, which is the storage pool name. Each returned project contains the list of modifiable properties listed above as well as the pool name, creation time, loading state, replication actions, and data usage.

Note:

The `depth` query parameter and the `match_*property-name*=*value*` query parameter are not supported.

Request parameters: `filter` – A simple string match filter that requires a property within the project to contain the same filter string within its value.

Example Request:

```
GET /api/storage/v1/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

On a successful `get`, an HTTP code 200 (OK) is returned along with an array of project properties in JSON format.

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "projects": [{
        "name": "proj-01",
        ...
    }, {
        "name": "proj-02",
        ...
    }
}
```

A list of all projects across all pools is also supported; the URI would contain only the / projects path.

Example Request to get all projects with `backup` as part of its properties:

```
GET /projects?filter=backup HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Get Project Properties

This command lists the properties for a single project in a given pool. A successful `get` returns HTTP Code 200 (OK) along with the project properties in JSON format.

Example Request to list the project named `proj-01` in the `zfs-storage-1` pool:

```
GET /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "project": {
        "default_volblocksize": 8192.0,
        "logbias": "latency",
        "creation": "20130411T20:02:35",
        "nodestroy": false,
        "dedup": false,
        "sharenfs": "on",
        "sharesmb": "off",
        "default_permissions": "700",
        "mountpoint": "/export",
        "snaplabel": "",
        "id": "042919bb-0882-d903-0000-000000000000",
        "readonly": false,
        "rrsrc_actions": [],
        "compression": "off",
        "sharetftp": "",
        "default_sparse": false,
        "snapdir": "hidden",
        "aclmode": "discard",
        "copies": 1,
        "aclinherit": "restricted",
        "shareftp": "",
        "canonical_name": "zfs-storage-1/local/default",
        "recordsize": 131072.0,
        "usage": {
            "available": 1758424767306.0,
            "loading": false,
            "quota": 0.0,
            "snapshots": 0.0,
            "compressratio": 100.0,
            "child_reservation": 0.0,
            "reservation": 0.0,
            "total": 45960.0,
            "data": 45960.0
        },
        "default_volsize": 0.0,
        "secondarycache": "all",
        "collection": "local",
        "exported": true,
        "vscan": false,
        "reservation": 0.0,
        "atime": true,
        "pool": "p1",
        "default_user": "nobody",
        "name": "default",
        "checksum": "fletcher4",
        "default_group": "other",
        "sharesftp": "",
        "nbmand": false,
        "sharedav": "",
        "rstchown": true
    }
}
```

Create Project

The create project command creates a project with a given name residing in the given storage pool. The request takes a single URI parameter, which is the storage pool name. The new project with default properties is returned.

JSON body request parameters:

- name – The project name must be supplied to create a project
- Project properties – Any of the project properties can be set as the new project's initial values

Example request to create a project named proj-01:

```
POST /api/storage/v1/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "name": "proj-01",
    "sharenfs": "ro"
}
```

Successful creation returns HTTP status 201 (Created) with the location header containing the URI of the new project. The body contains all of the project properties in JSON format.

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-01

{
    "project": {
        "name": "proj-01",
        "href": "/api/storage/v1/pools/p1/projects/proj-01",
        "mountpoint": "/export/acme/zfs-storage-1",
        ...
    }
}
```

Modify Project

The modify project command changes the attributes of an existing project. The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name

Request parameters: project properties – Any of the project properties can be set as the new project's initial values.

Example request to change a project's name from proj-01 to new-name:

```
POST /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "name": "new-name",
    "sharenfs": "rw",
    "compression": "gzip-9"
}
```

Successful response returns HTTP status 202 (Accepted) and lists all project properties.

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/new-name

{
    "project": {
        "name": "new-name",
        "sharenfs": "rw",
        "compression: "gzip-9",
         ...
    }
}
```

Delete Project

The delete project command removes a single project in a given pool. The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name

To monitor the amount of space to be reclaimed in the storage pool if deferred update Asynchronous Dataset Deletion (OS8.7.0 or later) has been accepted, enter the GET command for `pools/`*pool*. Note the amount of space for property *async_destroy_reclaim_space*. When the operation has completed, 0 (zero) is displayed.

Example Request:

```
DELETE /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Project Usage

Get requests project usage resources can be used to get usage data per user or per group for the project.

Filesystem Operations

Filesystem operations list and manage filesystem shares. All commands are scoped to a given storage pool or project.

`service_uri``/pools/`*`pool`*`/project/`*`project`*

Table 13-5 Filesystem Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/filesystems	List all filesystems
GET	/pools/*pool*/projects/*project*/filesystems	List specified filesystems

Table 13-5 (Cont.) Filesystem Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/pools/*pool*/projects/*project*/filesystems?snaps=true	List all filesystems, including snapshots
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*	Get filesystem details
POST	/pools/*pool*/projects/*project*/filesystems	Create a filesystem
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*	Modify a filesystem
DELETE	/pools/*pool*/projects/*project*/filesystems/*filesystem*	Destroy a filesystem
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/groups	Get filesystem group usage
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/groups/*group*	Get filesystem usage for the specified group
POST	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/groups	Create a filesystem group quota
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/groups/*name*	Modify a filesystem group quota
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/users	Get filesystem user usage
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/users/*user*	Get filesystem usage for the specified user
POST	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/users	Create a filesystem user quota
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*/usage/users/*name*	Modify a filesystem user quota
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/shadow/errors	List Shadow Migration Errors

Each filesystem contains properties from the project and has the following filesystem-specific properties.

Table 13-6 Filesystem Properties

Property	Type	Description
`casesensitivity`	string	Case Sensitivity setting: `mixed`, `sensitive`, or `insensitive`
`group`	string	The group name
`normalization`	string	Normalization
`permissions`	string	The filesystem permissions
`project`	string	The project name
`quota_snap`	boolean	Flag to include snapshots in the quota
`reservation_snap`	boolean	Flag to include snapshots in the reservation
`retention_policy`	string	File retention: File retention policy: `disabled`, `privileged`, or `mandatory`

Table 13-6 (Cont.) Filesystem Properties

Property	Type	Description
`retention_policy_changeacl`	boolean	File retention: Determines if a retained file's ACL/permissions can be changed: `off` or `on`
`retention_policy_onexpiry`	string	File retention: Determines behavior when file retention expires: `off`, `delete`, or `hold`
`retention_period_deletegrace`	number	File retention: Number of seconds/ hours/days/years that automatic file deletion is delayed when `retention_policy_onexpiry` is set to "delete." Default value: 0 days.
`retention_period_default`	number	File retention: Default amount of time for which a file is retained if it is automatically retained, or retained manually without first changing the file's access time attribute. Must be set if `retention_policy` is not "disabled."
`retention_period_min`	number	File retention: Minimum amount of time for file retention. Must be set if `retention_policy` is not "disabled."
`retention_period_max`	number	File retention: Maximum amount of time for file retention. Must be set if `retention_policy` is not "disabled."
`retention_period_grace`	number	File retention: Amount of time a file must remain unmodified before it is automatically retained at the default file retention period value
`retention_status_expiry`	string	File retention: Expiration date and time for a file
`retention_status_files`	string	File retention: File status for expiration date, time, and if expired
`shadow`	string	Data migration source
`errors`	string	Data migration errors
`sharesmb_name`	string	Name of SMB share
`source`	object	Project inheritance properties
`usage`	object	File system usage information
`user`	string	The user name that owns the share
`utf8only`	boolean	Flag to reject non-UTF-8

List Filesystems

The list filesystems command shows all filesystems in a given pool or project.

> **Note:**
>
> The `depth` query parameter and the `match_property-name=value` query parameter are not supported.

Request parameters: `filter` – A simple string match filter that requires a property within the project to contain the same filter string within its value.

The list filesystems command uses the following URI parameters:

- *pool* - Storage pool name
- *project* - Project name

Example Request:

```
GET /api/storage/v1/pools/p1/projects/proj-01/filesystems HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful request returns HTTP status 200 (OK) along with an array of filesystem properties in JSON format.

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "filesystems": [{
        "name": "filesystem-01",
        "project": "proj-01",
        "pool": "p1",
        ...
    }, {
        "name": "filesystem-02",
        "project": "proj-01",
        "pool": "p1",
        ...
    }]
}
```

A list of all filesystems across all pools and projects is also supported. In that case, the URI would be `/api/storage/v{1|2}/filesystems`.

Example Request to get all filesystems with the "abcd" string as part of its properties:

```
GET /api/storage/v1/filesystems?filter=abcd HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Get Filesystem

The get filesystem command returns the properties of a single filesystem in a given pool or project. The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name

- *filesystem* - Filesystem name

Example request to list project named `proj-01`:

```
GET /api/storage/v1/pools/p1/projects/proj-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful `get` returns HTTP status 200 (OK) along with the filesystem properties in JSON format.

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "filesystem": {
        "logbias": "latency",
        "creation": "20130423T21:30:34",
        "nodestroy": false,
        "dedup": false,
        "sharenfs": "on",
        "sharesmb": "off",
        "mountpoint": "/export/mnt1",
        "snaplabel": "",
        "id": "424ca2ec-b3fa-df86-0000-000000000000",
        "readonly": false,
        "rrsrc_actions": [],
        "compression": "off",
        "sharetftp": "",
        "source": {
            "logbias": "default",
            "dedup": "default",
            "sharenfs": "inherited",
            "sharesmb": "off",
            "mountpoint": "inherited",
            "rrsrc_actions": "local",
            "compression": "default",
            "sharetftp": "inherited",
            "snapdir": "default",
            "aclmode": "default",
            "copies": "default",
            "aclinherit": "default",
            "shareftp": "inherited",
            "readonly": "default",
            "secondarycache": "default",
            "exported": "inherited",
            "vscan": "default",
            "reservation": "local",
            "atime": "default",
            "recordsize": "default",
            "checksum": "inherited",
            "sharesftp": "inherited",
            "nbmand": "default",
            "rstchown": "default"
        },
        "snapdir": "hidden",
        "aclmode": "discard",
        "copies": 1,
        "aclinherit": "restricted",
        "shareftp": "",
        "canonical_name": "p1/local/default/mnt1",
        "recordsize": 131072.0,
        "usage": {
            "available": 880395477504.0,
            "loading": false,
            "quota": 0.0,
            "snapshots": 18432.0,
            "compressratio": 100.0,
            "reservation": 0.0,
            "total": 50176.0,
            "data": 31744.0
        },
        "secondarycache": "all",
        "collection": "local",
        "exported": true,
        "vscan": false,
        "reservation": 0.0,
        "shadow": "none",
        "atime": true,
        "pool": "p1",
        "quota_snap": true,
        "name": "mnt1",
        "checksum": "fletcher4",
        "project": "default",
        "sharesftp": "",
        "nbmand": false,
        "reservation_snap": true,
        "sharedav": "",
        "rstchown": true,
        "root_acl": {
             "owner@:cC:fd:deny",
             "everyone@:rw:fd:allow",
             "user:admin1:rw:allow",
        }
        "smbshareacl": {
             "owner@:cC:fd:deny",
             "everyone@:rw:fd:allow",
             "user:admin1:rw:allow",
        }
}
```

Create Filesystem

The create filesystem command creates a filesystem with a given name residing in the given storage pool or project. The new filesystem with default properties is returned.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name
- *filesystem* - Filesystem name

Request parameters:

- `name` – The filesystem name must be supplied to create a new filesystem
- Filesystem properties – Any of the properties listed in filesystem properties or project properties can be set as initial values

Example Request (to create a filesystem named `share-01` and owned by the user `admin1`):

```
POST /api/storage/v1/pools/p1/projects/proj-01/filesystems HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "name": "share-01",
    "root_user": "admin1"
}
```

Successful creation returns HTTP status 201 (Created) with the Location header containing the URI of the new filesystem. The body contains all filesystem properties in JSON format.

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01

{
    "filesystem": {
        "name": "share-01",
        "pool": "p1",
        "collection": "local",
        "project": "proj-01",
        "root_user": "admin1"
        ...
    }
}
```

Modify Filesystem

The modify filesystem command changes the attributes of an existing filesystem. Successful response returns HTTP status 202 (Accepted) and lists all filesystem properties.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name
- *filesystem* - Filesystem name

Request parameters: filesystem properties – Any of the filesystem or project properties can be modified.

Example Request (to change a filesystem name from `share-01` to `new-name` and change the owner to `nobody`):

```
PUT /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "name": "new-name",
    "root_user": "nobody"
}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-01/filesystems/
share-01

{
    "filesystem": {
        "name": "new-name",
        "pool": "p1",
        "collection": "local",
        "project": "proj-01",
        "root_user": "nobody"
        ...
    }
}
```

Delete Filesystem

The delete filesystem command removes a single filesystem in a given pool or project.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name
- *filesystem* - Filesystem name

To monitor the amount of space to be reclaimed in the storage pool, enter the GET command for `pools/`*pool*. Note the amount of space for property *async_destroy_reclaim_space*. When the operation has completed, 0 (zero) is displayed.

Example Request:

```
DELETE /api/storage/v1/pools/p1/projects/proj-01/filesystems/share-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful delete returns HTTP status 204 (No Content).

Example Result:

```
HTTP/1.1 204 No-Content
```

Filesystem Quota and Usage

User or group quotas can be created or modified with `POST` or `PUT` requests, respectively. `GET` requests to filesystem use resources are used to get usage data per user or per group for the project.

LUN Operations

All LUN or volume operations are scoped to a given pool or project. The following LUN commands are available.

Table 13-7 Volume Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/luns	List all LUNs
GET	/pools/*pool*/projects/*project*/luns	List LUNs
GET	/pools/*pool*/projects/*project*/luns?snaps=true	List all LUNs, including snapshots
GET	/pools/*pool*/projects/*project*/luns/*lun*	Get LUN details
POST	/pools/*pool*/projects/*project*/luns	Create a LUN
PUT	/pools/*pool*/projects/*project*/luns/*lun*	Modify a LUN
DELETE	/pools/*pool*/projects/*project*/luns/*lun*	Destroy a LUN

The following table lists the LUN properties. Volumes can also inherit or override project properties.

Table 13-8 Volume Properties

Property	Type	Description
`assignednumber`	number or list of numbers	The assigned LU number. If presented to multiple initiator groups, the type is a list of numbers. If presented to multiple initiator groups, the ordering of `assignednumber` and `initiatorgroups` are aligned. For example, the first item in the `assignednumber` list pertains to the first item in the `initiatorgroups` list.
`fixednumber`	boolean	Flag to fix LU number at current value
`initiatorgroups`	list of strings	The initiator group. If the LUN is presented to multiple initiator groups, the ordering of `assignednumber` and `initiatorgroups` are aligned. For example, the first item in the `assignednumber` list pertains to the first item in the `initiatorgroups` list.
`lunguid`	string	STMF GUID
`lunumber`	number or string	The LU number. Either a number or `auto`
`project`	string	The project name (immutable)
`source`	object	Lists source of properties: `local` or `inherited`
`sparse`	boolean	Flag to enable thin provisioning
`status`	string	Logical unit status: `online` or `offline`
`targetgroup`	string	The target group
`usage`	object	Lists LUN usage statistics
`volblocksize`	number	Volume block size

Table 13-8 (Cont.) Volume Properties

Property	Type	Description
`volsize`	number	Volume size
`writecache`	boolean	Flag to enable write cache

Some properties can be inherited from the project. The source object lists each of these properties and identifies whether the property is local to the LUN or is inherited from the project. By default these properties are inherited by the project. Once set, they are local to the LUN. The source object is immutable. To change the source back to inherited, the properties can be unset.

Example JSON request to unset compression:

```
{"unset": ["compression"]}
```

List LUNs

The list LUNs command returns a list of LUNs available in a given pool or project.

> **Note:**
>
> The `depth` query parameter and the `match_`*`property-name`*`=`*`value`* query parameter are not supported.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name
- *filesystem* - Filesystem name

Example request to list LUNs within project `proj-01`:

```
GET /api/storage/v1/pools/p1/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful `get` returns HTTP status 200 (OK) along with the LUN properties in JSON format.

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "luns": [{
        "id": "fa4ac6fb-0bcc-d2e3-0000-000000000000",
        "name": "vol-01"
        ...
    }, {
         "id": "690ae407-7c4d-b5d2-0000-000000000000",
         "name": "vol-01",
         ....
    }]
}
```

Get LUN

The get LUN command returns the properties of a single LUN in a given pool or project.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name
- *lun* - LUN name

Example Request (to get a LUN named "vol-01"):

```
GET /api/storage/v1/pools/p1/projects/proj-01/lun/vol-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful `get` returns HTTP status 200 (OK) along with the LUN properties in JSON format.

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "lun": {
        "logbias": "latency",
        "creation": "20130423T21:31:17",
        "nodestroy": false,
        "dedup": false,
        "rrsrc_actions": [],
        "id": "e3045406-319b-cf7a-0000-000000000000",
        "writecache": false,
        "compression": "off",
        "copies": 1,
        "stmfguid": "600144F0D8E0AE4100005176FDA60001",
        "source": {
            "compression": "default",
            "checksum": "inherited",
            "logbias": "default",
            "dedup": "default",
            "copies": "default",
            "exported": "inherited",
            "rrsrc_actions": "inherited",
            "secondarycache": "default"
        },
        "canonical_name": "p1/local/default/disk1",
        "snaplabel": "",
        "usage": {
            "available": 881469214720.0,
            "loading": false,
            "snapshots": 0.0,
            "compressratio": 100.0,
            "total": 1073758208.0,
            "data": 1073758208.0
        },
        "secondarycache": "all",
        "collection": "local",
        "exported": true,
        "volsize": 1073741824.0,
        "pool": "p1",
        "volblocksize": 8192,
        "checksum": "fletcher4",
        "project": "default",
        "sparse": false
    }
}
```

Create a New LUN

This command creates a new LUN. You must supply a size or a cloning source for the new LUN.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name

Request Parameters:

- `name` – The LUN name must be supplied to create a new LUN.
- `volsize` - Size of the new LUN (mandatory), either as an exact number of bytes, or using units, such as `1M`. `volsize` must be a multiple of `blocksize` and it must be at least 1M (1048576).
- Volume properties – Any of the properties listed in LUN properties or project properties can be set as initial values.

Example Request:

```
POST /api/storage/v1/pools/p1/projects/proj-01/luns HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json

{
        name : "vol-001",           // Volume name (required)

        volsize : "1M",             // New Volume size (required)
        blocksize : 8192,           // New Volume block size
        sparse : true,              // New Volume sparse data flag

        initiatorgroup : 'default', // Initiator group name
        targetgroup : 'default',    // Target group name
        lunumber : 'auto',          // Volume LU number
        status : 'online',          // Initial Status ('online', 'offline')
        fixednumber : false,

        "source": {
            "snapshot_id" : "76b8950a-8594-4e5b-8dce-0dfa9c696358",
            "snapshot": "/pool-001/local/proj-001/snap-001"
        }
}
```

Successful creation returns HTTP status 201 (Created) with the Location header containing the URI of the new LUN. The body contains all of the LUN properties in JSON format.

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-01/luns/
vol-001

{
    "lun": {
        "name": "vol-001",
        ...
    }
}
```

Modify LUN

The modify LUN command changes the attributes of an existing LUN.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name
- *lun* - LUN name

Request parameters: volume properties – Any of the LUN or project properties can be modified.

Example request to change a LUN name from `vol-01` to `new-name`:

```
POST /api/storage/v1/pools/p1/projects/proj-01/luns/vol-01 HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "name": "new-name"
}
```

Successful response returns HTTP status 202 (Accepted) and lists all LUN properties.

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /api/storage/v1/pools/p1/projects/proj-01/luns/new-name

{
    "lun": {
        "name": "new-name",
        "pool": "p1",
        "collection": "local",
        "project": "proj-01",
        ...
    }
}
```

Delete LUN

The delete LUN command removes a single LUN in a given pool or project.

The following URI parameters are used:

- *pool* - Storage pool name
- *project* - Project name
- *lun* - LUN name

To monitor the amount of space to be reclaimed in the storage pool, enter the GET command for pools/*pool*. Note the amount of space for property *async_destroy_reclaim_space*. When the operation has completed, 0 (zero) is displayed.

Example Request:

```
DELETE /pools/p1/projects/proj-01/luns/lun-01 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Successful `get` returns HTTP status 204 (No Content).

Example Result:

```
HTTP/1.1 204 No-Content
```

Snapshot and Clone Operations

All snapshot operations are scoped to a given pool or project. Snapshot operations can also be scoped to the filesystem or LUN level.

- The URI for all project-based snapshot operations begins with: `/api/storage/v{1|2}/pools/`*`pool`*`/projects/`*`project`*
- The URI for all filesystem-based snapshot operations begins with: `/api/storage/v{1|2}/pools/`*`pool`*`/projects/`*`project`*`/filesystems/`*`filesystem`*
- The URI for all LUN-based snapshot operations begins with: `/api/storage/v{1|2}/pools/`*`pool`*`/projects/`*`project`*`/luns/`*`lun`*

To back up a snapshot to the cloud, or restore a snapshot backup to Oracle ZFS Storage Appliance as a new share, see RESTful API Cloud Service.

Table 13-9 Snapshot and Clone Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/snapshots	List all local snapshots
GET	/pools/*pool*/projects?snaps=true	List all projects, including snapshots
GET	/pools/*pool*/projects/*project*/filesystems?snaps=true	List all filesystems, including snapshots
GET	/pools/*pool*/projects/*project*/luns?snaps=true	List all LUNs, including snapshots
GET	/pools/*pool*/projects/*project*/snapshots	List all snapshots for a project
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots	List all snapshots for a filesystem
GET	/pools/*pool*/projects/*project*/luns/*lun*/snapshots	List all snapshots for a LUN
GET	/pools/*pool*/projects/*project*/snapshots/*snapshot*	Get project snapshot details
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Get filesystem snapshot details

Table 13-9 (Cont.) Snapshot and Clone Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/pools/*pool*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Get LUN snapshot details
POST	/pools/*pool*/projects/*project*/snapshots	Create a project snapshot
POST	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots	Create a filesystem snapshot
POST	/pools/*pool*/projects/*project*/luns/*lun*/snapshots	Create a LUN snapshot
PUT	/pools/*pool*/projects/*project*/snapshots/*snapshot*	Modify a project snapshot
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Modify a filesystem snapshot
PUT	/pools/*pool*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Modify a LUN snapshot
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*/clone	Clone a filesystem snapshot
PUT	/pools/*pool*/projects/*project*/luns/*lun*/snapshots/*snapshot*/clone	Clone a LUN snapshot
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*/rollback	Rollback data to the given filesystem snapshot
PUT	/pools/*pool*/projects/*project*/lun/*lun*/snapshots/*snapshot*/rollback	Rollback data to the given LUN snapshot
DELETE	/pools/*pool*/projects/*project*/snapshots/*snapshot*	Destroy a project snapshot
DELETE	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Destroy a filesystem snapshot
DELETE	/pools/*pool*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Destroy a LUN snapshot
GET	/pools/*pool*/projects/*project*/snapshots/*snapshot*/dependents	List project snapshot dependents
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*/dependents	List filesystem snapshot dependents
GET	/pools/*pool*/projects/*project*/lun/*lun*/snapshots/*snapshot*/dependents	List LUN snapshot dependents
POST	/pools/*pool*/projects/*project*/automatic	Create a new project automatic snapshot object
POST	/pools/*pool*/projects/*project*/automatic?convert=true	Create a new project automatic snapshot object. Optionally, set a retention hold. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
GET	/pools/*pool*/projects/*project*/automatic/*automatic*	Get the specified project automatic snapshot properties
GET	/pools/*pool*/projects/*project*/automatic	List all project automatic snapshot objects

Table 13-9 (Cont.) Snapshot and Clone Commands

Request	Append to Path /api/storage/v{1|2}	Description
PUT	/pools/*pool*/projects/*project*/automatic/*automatic*	Modify the specified project automatic snapshot object
PUT	/pools/*pool*/projects/*project*/automatic/*automatic*?convert=true	Modify the specified project automatic snapshot schedule object. Optionally, modify the retention hold setting. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
DELETE	/pools/*pool*/projects/*project*/automatic/*automatic*	Destroy the specified automatic object
DELETE	/pools/*pool*/projects/*project*/automatic/*automatic*?convert=true	Destroy the specified automatic snapshot schedule object. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
POST	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic	Create a new filesystem automatic snapshot object
POST	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic?convert=true	Create a new filesystem automatic snapshot object. Optionally, set a retention hold. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic/*automatic*	Get the specified filesystem automatic snapshot properties
GET	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic	List all filesystem automatic snapshot objects

Table 13-9 (Cont.) Snapshot and Clone Commands

Request	Append to Path /api/storage/v{1|2}	Description
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic/*automatic*	Modify the specified filesystem automatic snapshot object
PUT	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic/*automatic*?convert=true	Modify the specified filesystem automatic snapshot schedule object. Optionally, modify the retention hold setting. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
DELETE	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic/*automatic*	Destroy the specified automatic snapshot schedule object
DELETE	/pools/*pool*/projects/*project*/filesystems/*filesystem*/automatic/*automatic*?convert=true	Destroy the specified filesystem automatic snapshot schedule object. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
POST	/pools/*pool*/projects/*project*/luns/*lun*/automatic	Create a new LUN automatic snapshot
POST	/pools/*pool*/projects/*project*/luns/*lun*/automatic?convert=true	Create a new LUN automatic snapshot schedule. Optionally, set a retention hold. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
GET	/pools/*pool*/projects/*project*/luns/*lun*/automatic/*automatic*	Get the specified LUN automatic snapshot properties

Table 13-9 (Cont.) Snapshot and Clone Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/pools/*pool*/projects/*project*/luns/*lun*/automatic	List all LUN automatic snapshot objects
PUT	/pools/*pool*/projects/*project*/luns/*lun*/automatic/*automatic*	Modify the specified LUN automatic snapshot object
PUT	/pools/*pool*/projects/*project*/luns/*lun*/automatic/*automatic*?convert=true	Modify the specified LUN automatic snapshot schedule object. Optionally, modify the retention hold setting. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.
DELETE	/pools/*pool*/projects/*project*/luns/*lun*/automatic/*automatic*	Destroy the specified LUN automatic object
DELETE	/pools/*pool*/projects/*project*/luns/*lun*/automatic/*automatic*?convert=true	Destroy the specified LUN automatic snapshot schedule object. Existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the convert property causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.

List Snapshots

Lists available snapshots on an Oracle ZFS Storage Appliance system. Depending on the request URI, the list contains project, filesystem, or LUN snapshots.

Table 13-10 List Snapshot Command Forms

Command	Append to Path /api/storage/v{1|2}/pools/*pool*/projects/*project*
List project snapshots	/snapshots
List filesystem snapshots	/filesystems/*share*/snapshots
List LUN snapshots	/lun/*share*/snapshots

Example Request:

```
GET /api/storage/v1/pools/p1/projects/default/snapshots
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "snapshots": [{
        "canonical_name": "p1/local/default@snap-001",
        "collection": "local",
        "creation": "20211104T11:00:00",
        "href": "/api/storage/v1/pools/p1/projects/default/snapshots/snap-001",
        "id": "1bc742f1-6a56-bf7c-0000-000000000000",
        "isauto": false,
        "name": "snap-001",
        "numclones": 0,
        "pool": "p1",
        "project": "default",
        "retentionpolicy": "off",
        "shadowsnap": false,
        "space_data": 31744,
        "space_unique": 0,
        "type": "snapshot"
    }, {

        "canonical_name": "p1/local/default@snap-002",
        "collection": "local",
        "creation": "20211104T11:00:07",
        "href": "/api/storage/v1/pools/p1/projects/default/snapshots/snap-002",
        "id": "24927817-ac89-5071-0000-000000000000",
        "isauto": false,
        "name": "snap-002",
        "numclones": 0,
        "pool": "p1",
        "project": "default",
        "retentionpolicy": "unlocked",
        "shadowsnap": false,
        "space_data": 31744,
        "space_unique": 0,
        "type": "snapshot"
    }]
}
```

Get Snapshot

View all information about a single snapshot. Returns HTTP status 200 (OK) on success.

Example Request:

```
GET /api/storage/v1/pools/p1/projects/default/snapshots/snap-001
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

```
{
    "snapshot": {
        "canonical_name": "p1/local/default@snap-001",
        "collection": "local",
        "creation": "20211104T11:00:00",
        "href": "/api/storage/v1/pools/p1/projects/default/snapshots/snap-001",
        "id": "1bc742f1-6a56-bf7c-0000-000000000000",
        "isauto": false,
        "name": "snap-001",
        "numclones": 0,
        "pool": "p1",
        "project": "default",
        "retentionpolicy": "off",
        "shadowsnap": false,
        "space_data": 31744,
        "space_unique": 0,
        "type": "snapshot"
    }
}
```

Create Snapshot

The create snapshot command creates snapshots for projects, filesystems, or LUNs.

- Create Project Snapshot – `POST /pools/`*`pool`*`/projects/`*`project`*`/snapshots`
- Create Filesystem Snapshot – `POST /pools/`*`pool`*`/projects/`*`project`*`/filesystems/`*`share`*`/snapshots`
- Create Volume Snapshot – `POST /pools/`*`pool`*`/projects/`*`project`*`/luns/`*`lun`*`/snapshots`

Optionally, you can set a retention policy:

- If the snapshot is a project snapshot, the retention settings also apply to all of its shares.
- For a manual snapshot object, you can set the `retention` policy property to `off` or `unlocked`.
- For an automatic snapshot schedule object, you can set the `retentionpolicy` property to `off` or `locked`. If `locked`, set the corresponding `retentionhold` value, which must be the same or less than the `keep` value.
- When creating a new automatic snapshot schedule object and setting the `convert` property to `true`, existing auto-generated snapshots that do not fit new schedules are converted to manual snapshots. Excluding the `convert` property or setting it to `false` causes existing auto-generated snapshots to be destroyed. If the snapshots have a retention hold, the `convert` property setting does not change the retention hold nor can the snapshots be converted to manual snapshots.

To use the snapshot retention hold feature, apply deferred update "Support for Snapshot Retention." For information about deferred updates, see Deferred Updates in *Oracle ZFS Storage Appliance Customer Service Manual, Release OS8.8.x*. To understand the required user role authorizations, see Taking a Snapshot (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x* and Scheduling Snapshots (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

Route `retention` can be used instead of route `retentionpolicy`. Both are the same.

```
POST /api/storage/v1/pools/p1/projects/default/snapshots
Content-Type: application/json

{
    "name": "initial-backup",
    "retentionpolicy": "unlocked"
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: /pools/p1/projects/default/snapshot/initial-backup

{
    "snapshot": {
        "name": "initial-backup",
        "numclones": 0,
        "creation": "20130610T21:00:49",
        "collection": "local",
        "project": "default",
        "canonical_name": "zfs-storage-1/local/default@initial-backup",
        "usage": {
            "unique": 0.0,
            "loading": false,
            "data": 145408.0
        },
        "type": "snapshot",
        "id": "a26abd24-e22b-62b2-0000-000000000000",
        "pool": "p1",
        "retention": "unlocked"
    }
}
```

Rename Snapshot

Renames an existing snapshot.

- **Request URI** – Snapshot, the current snapshot name
- **Request Body** – JSON object with name parameter containing new snapshot name

Optionally, you can modify the retention policy setting while in this same path:

- If the snapshot is a project snapshot, the retention settings also apply to all of its shares.
- For a manual snapshot object, you can modify the `retention` policy property to `off` or `unlocked`, as appropriate.
- If automatic snapshots containing a retention hold have been generated with this schedule, the `retentionhold` property must be set to a higher value to prevent early lock removal, but not higher than the `keep` property. If no automatic snapshots have been generated with this schedule, the retention hold can be set to a lower value.

To use the snapshot retention hold feature, apply deferred update "Support for Snapshot Retention." For information about deferred updates, see Deferred Updates in *Oracle ZFS Storage Appliance Customer Service Manual, Release OS8.8.x*. To understand the required user role authorizations, see Renaming a Snapshot (CLI) in

Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x and Editing a Snapshot Retention Policy (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
PUT /api/storage/v1/pools/p1/projects/default/snapshots/initial-snapshot
Content-Type: application/json
Accept: application/json

{
    "name": "old-snapshot"
}
```

Example Result:

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Location: /pools/p1/projects/default/snapshot/initial-backup
```

Clone Snapshot

Makes a new filesystem or LUN from an existing snapshot.

The following URI parameters are used:

- *pool* - Source pool name
- *project* - Source project name
- *filesystem* - Source share name for filesystem snapshot
- *lun* - Source share name for LUN snapshot
- *snapshot* - Source snapshot name

Clone a filesystem:

```
PUT /pools/pool/projects/project/filesystems/share/snapshots/snapshot/clone
```

Clone a volume:

```
PUT /pools/pool/projects/project/luns/lun/snapshots/snapshot/clone
```

Request body contains a JSON object with the following properties.

The clone will have the same retention hold setting as the original snapshot. To apply or remove a retention hold for the clone, make a snapshot of the clone and specify a new retention hold setting. To use the snapshot retention hold feature, apply deferred update "Support for Snapshot Retention." For information about deferred updates, see Deferred Updates in *Oracle ZFS Storage Appliance Customer Service Manual, Release OS8.8.x*.

Table 13-11 Clone Snapshot Properties

Property	Type	Description
`pool`	string	Destination clone pool name
`project`	string	Destination clone project name
`lun`	string	Destination LUN name for LUN snapshot

Example Request:

```
PUT /api/storage/v1/pools/p1/projects/default/filesystems/fs01/snapshots/snap01/
clone

{
    "project":"rest",
    "share":"snap01clone01",
    "compression": "gzip-9"
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Length: 2035
X-Zfssa-Storage-Api: 1.0
Location: /api/storage/v1/pools/p1/projects/rest/filesystem/snap01clone01
Content-Type: application/json; charset=utf-8

{
    "filesystem": {
        "origin": {
            "project": "default",
            "share": "fs01",
            "snapshot": "snap01",
            "pool": "p1",
            "collection": "local"
        },
        "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/
snap01clone01",
        "mountpoint": "/export/snap01clone01",
        "compression": "gzip-9",
        "source": {
            "compression": "local",
            ...
        },
        ...
       "canonical_name": "zfs-storage-1/local/rest/snap01clone01"
   }
}
```

Rollback Snapshot

The rollback snapshot causes the source file system or LUN to be modified back to its state when the snapshot was taken. Successful response returns HTTP status 202 (Accepted), as well as the snapshot properties in JSON format.

The rollback is not allowed if the rollback would remove recent snapshots with a retention hold.

The following URI parameters are used:

- *pool* - Source pool name
- *project* - Source project name
- *filesystem* - Source filesystem name for filesystem snapshot
- *lun* - Source LUN name for LUN snapshot
- *snapshot* - Source snapshot name

Rollback a filesystem snapshot:

```
PUT /pools/pool/projects/project/filesystems/filesystem/snapshots/snapshot/rollback
```

Rollback a LUN snapshot:

```
PUT /pools/pool/projects/project/luns/lun/snapshots/snapshot/rollback
```

Example Request:

```
PUT /api/storage/v1/pools/p1/projects/default/filesystems/fs-01/snapshots/initial-
backup/rollback
```

Example Result:

```
HTTP/1.1 202 Accepted
Location: /pools/p1/projects/default/filesystems/fs-01/snapshot/fs-01-initial-clone
Content-Type: application/json

{
    "snapshot": {
        "name": "fs-01-initial-clone",
        "numclones": 0,
        "creation": "20130610T21:00:49",
        "filesystem": "fs-01",
        "collection": "local",
        "project": "default",
        "canonical_name": "zfs-storage-1/local/default/
               fs-01@fs-01-initial-clone",
        "usage": {
            "unique": 0.0,
            "loading": false,
            "data": 31744.0
        },
        "type": "snapshot",
        "id": "5c9bda07-21c1-2238-0000-000000000000",
        "pool": "p1"
    }
}
```

Delete a Snapshot

The `DELETE` snapshot command deletes a project, filesystem, or LUN snapshot from the Oracle ZFS Storage Appliance system.

The following URI parameters are used:

- *pool* - Source pool name
- *project* - Source project name
- *filesystem* - Source filesystem name
- *lun* - LUN name
- *snapshot* - Source snapshot name

Delete a project snapshot:

```
DELETE /api/storage/v1/pools/pool/projects/project/snapshots/snapshot
```

Delete a filesystem snapshot:

```
DELETE /api/storage/v1/pools/pool/projects/project/filesystems/filesystem/
snapshots/snapshot
```

Delete a filesystem LUN:

```
DELETE /api/storage/v1/pools/pool/projects/project snapshot
```

If the snapshot has an NDMP hold on it, add `?confirm=true` to the `DELETE` command. Note, however, that this could adversely affect NDMP operations. For more information, see NDMP Configuration in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Before a manual snapshot with a retention hold can be deleted, the hold type must be `off`. To modify a manual snapshot from `unlocked` to `off`, use the PUT operation.

An error message might warn that existing automatic snapshots could be destroyed. If the snapshot or its children are actively changing, an error message indicates that the snapshot schedule cannot be removed. Also, if the schedule contains locked automatic snapshots, the schedule cannot be removed until the retention holds expire. If the automatic snapshot schedule has a retention hold but no snapshots have been generated, the schedule can be removed. If the snapshot is a project snapshot, the schedule will also be removed from its shares.

To use the snapshot retention hold feature, apply deferred update "Support for Snapshot Retention." For information about deferred updates, see Deferred Updates in *Oracle ZFS Storage Appliance Customer Service Manual, Release OS8.8.x*. To understand the required user role authorizations, see Removing a Snapshot Schedule (CLI) in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
DELETE /pools/p1/projects/default/filesystems/fs-01/snapshots/initial-backup?
confirm=true
```

Example result if `?confirm=true` is not added:

If `?confirm=true` is not added when an NDMP hold exists on the snapshot, then the command will fail with the following output (lines are artificially broken for readability):

```
HTTP/1.1 409 Conflict

{"fault": {"message": "request requires confirm=true to complete (confirmation
needed for scripted command (scripted commands must be prefixed with \"confirm\"
to automatically confirm or \"deny\" to automatically deny) (encountered while
attempting to run command \"confirm destroy snap\"))", "code": 409, "name":
"ERR_CONFIRM_REQUIRED"}}
```

List Snapshot Dependents

Lists dependents for a filesystem or volume. The following URI parameters are used:

- *pool* - System storage pool name
- *project* - Project name
- *filesystem* - Filesystem name
- *lun* - LUN name
- *snapshot* - Snapshot name

List filesystem dependents:

```
GET /api/storage/v1/pools/pool/projects/project/filesystems/filesystem/snapshots/
snapshot/dependents
```

List volume dependents:

```
GET /api/storage/v1/pools/pool/projects/project/lun/lun/snapshots/snapshot/dependents
```

Example Request:

```
GET /api/storage/v1/pools/p1/projects/default/filesystems/fs01/snapshots/snap01/
dependents
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Storage-Api: 1.0
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 1.0

{
    "dependents": [
        {
            "project": "rest",
            "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone01",
            "share": "snap01clone01"
        },
        {
            "project": "rest",
            "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone02",
            "share": "snap01clone02"
        },
        {
            "project": "rest",
            "href": "/api/storage/v1/pools/p1/projects/rest/filesystems/snap01clone03",
            "share": "snap01clone03"
        }
    ]
}
```

Schema

Manages custom schema properties.

Table 13-12 Schema Commands

Request	Append to Path /api/storage/v{1|2}	Description
GET	/schema	List all NAS schema property objects
GET	/schema/*property*	Get the specified NAS schema property properties
POST	/schema	Create a new NAS schema property
PUT	/schema/*property*	Modify the specified NAS schema property object

Table 13-12 (Cont.) Schema Commands

Request	Append to Path /api/storage/v{1|2}	Description
DELETE	/schema/*property*	Delete the specified NAS schema property object

Each custom schema property can be set on projects, filesystems, and LUNs by adding the prefix `custom:` to the custom property name.

For example, the following PUT body modifies a custom `int` property named `priority`:

```
{"custom:priority": 5}
```

Table 13-13 Schema Parameters

Parameter	Description
`property`	Name of property (immutable)
`description`	Property description (for browser interface)
`type`	Type (“String”, “Integer”, “PositiveInteger”, “Boolean”, “EmailAddress”, “Host”)

List Properties

Lists schema properties.

Example Request:

```
GET /api/storage/v1/schema
```

Example Result:

```
{
    "properties": [{
        "description": "bob",
        "href": "/api/storage/v1/schema/bob",
        "property": "bob",
        "type": "String"
    },{
        "description": "pat",
        "href": "/api/storage/v1/schema/pat",
        "property": "pat",
        "type": "String"
    }]
}
```

Get Property

Gets a schema property.

Example Request:

```
GET /api/storage/v1/schema/priority
```

Example Result:

```
{
    "property": {
        "description": "priority",
        "href": "/api/storage/v1/schema/priority",
        "property": "bob",
        "type": "Integer"
    }
}
```

Create Property

Creates a new schema property.

Example Request:

```
POST /api/storage/v1/schema HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Content-Length: 64

{"property":"priority", "type":"Integer", "description":"Oh my"}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Length: 89
X-Zfssa-Nas-Api: 1.0
Content-Type: application/json
Location: /api/storage/v1/schema/priority

{
    "property": {
        "href": "/api/storage/v1/schema",
        "type": "Integer",
        "description": "Oh my"
    }
}
```

Modify Property

Modifies a schema property.

Example Request:

```
PUT /api/storage/v1/schema/priority

{"description":"My custom priority level"}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Nas-Api: 1.0
Content-Type: application/json
Content-Length: 90

{
    "property": {
        "href": "//api/storage/v1/schema/priority",
        "type": "Integer",
        "description": "My custom priority level"
    }
}
```

Delete Property

This command deletes a schema property.

Example Request:

```
DELETE /api/storage/v1/schema/me HTTP/1.1
```

Example Result:

```
HTTP/1.1 204 No Content
```

Replication

Replication facilitates replication of projects and shares between Oracle ZFS Storage Appliance systems.

> **Note:**
>
> Replication is a licensed feature for certain models of Oracle ZFS Storage Appliance, and the replication RESTful API manages that feature. The service is available from the following URI: `https://`*`hostname`*`:215/api/storage/v{1|2}/replication`. For license details, refer to the Oracle Software License Agreement (SLA) and Entitlement for Hardware Systems with Integrated Software Options, and the Licensing Information User Manual for the software release.

The Replication RESTful API manages the following resources:

- **Replication Service** – The service that manages replication tasks.
- **Replication Target** – An Oracle ZFS Storage Appliance peer that receives and stores data replicated from another appliance peer (the source). This term also refers to a configuration object on the Oracle ZFS Storage Appliance system that enables it to replicate to another appliance.
- **Replication Action** – A configuration object on a source Oracle ZFS Storage Appliance system specifying a project or share, a target appliance, and policy options (including how often to send updates, whether to encrypt data on the wire, and so on).
- **Replication Package** – The target-side analog of an action; the configuration object on the target Oracle ZFS Storage Appliance that manages the data replicated as part of a particular action from a particular source. Each action on a source appliance is associated with exactly one package on a target appliance and vice versa. Loss of either object requires creating a new action/package pair (and a full replication update).

The API supplies replication operations for replication actions and replication packages. The service API is used to manage the replication service and replication sources and targets.

Table 13-14 Replication Service Commands

Request	Append to Path /api/service/v{1|2}/services	Description
GET	/replication	Get replication service state properties
PUT	/replication/enable	Enable the replication service
PUT	/replication/disable	Disable the replication service

List Replication Service Properties

Gets the state of the replication service.

Example Request:

```
GET /api/service/v2/services/replication HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK

{
    "service": {
        "href": "/api/service/v2/services/replication",
        "<status>": "online",
        "enable_start_finish_alerts": true
    }
}
```

Modify Replication Service Properties

The replication service state can be modified like any other service. See the Service RESTful API for more information.

Depending on the number of projects that are replicating and the frequency of the replication schedule, the number of start and finish alerts for scheduled updates can obscure other important alerts. To disable start and finish alerts for scheduled updates, set the `enable_start_finish_alerts` property to `false`:

```
PUT /api/service/v2/services/replication
Host: zfs-storage.example.com:215
Content-Type: application/json

{
    "enable_start_finish_alerts":false
}
```

Replication Targets

The following table shows the available replication target commands.

Table 13-15 Replication Target Commands

Request	Append to Path /api/storage/v{1|2}	Description
POST	/replication/targets	Create a new replication target
GET	/replication/targets/*target*	Get the specified replication target properties
GET	/replication/targets	List all replication target objects
PUT	/replication/targets/*target*	Modify the specified replication target object
DELETE	/replication/targets/*target*	Destroy the specified target object

The following table shows the properties of a replication target.

Property	Description
`label`	The target name to display
`hostname`	The fully qualified domain name or IPv4 address of the target Oracle ZFS Storage Appliance
`host_match`	Perform or bypass hostname verification; see Verify the Target Certificate
`auto_accept_cert`	Automatically accept the target's certificate; see Verify the Target Certificate

List Replication Targets

Lists all of the available replication targets on an Oracle ZFS Storage Appliance system.

Example Request:

```
GET /api/storage/v2/replication/targets HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 529

{
        "targets": [{
            "address": "ipaddr-1",
            "label": "zfs-storage-1",
            "hostname": "ipaddr-2",
            "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
            "actions": 0,
            "target": "target-000",
            "href": "/api/storage/v2/replication/targets/zfs-storage-1"
        }, {
            "address": "ipaddr-3",
            "label": "zfs-storage-2",
            "hostname": "ipaddr-4",
            "asn": "16a4c82c-26c1-4a50-e317-ac53181f2e86",
            "actions": 0,
            "target": "target-001",
            "href": "/api/storage/v2/replication/targets/zfs-storage-2"
        }]
}
```

List a Specified Replication Target

This command lists the details of a single replication target. The target is accessed by its hostname.

Example Request:

```
GET /api/storage/v2/replication/targets/zfs-storage-1 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 337

{
        "target": {
            "href": "/api/storage/v2/replication/targets/zfs-storage-1",
            "address": "ipaddr-1",
            "label": "zfs-storage-1",
            "hostname": "ipaddr-2",
            "asn": "9d7a7543-ca83-68f5-a8fc-f818f65e1cfc",
            "actions": 0
        }
}
```

Create a Replication Target

The `targets` command creates a new replication target for replication.

If you need to ensure that the replication traffic goes over a particular network interface, set up a static route for the target that specifies that interface as shown in Add Route.

Example Request:

See Verify the Target Certificate for information about the `hostname` and `auto_accept_cert` properties.

```
POST /api/replication/v2/targets HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54

{
    "label":"zfs-storage-3",
    "hostname":"zfs-storage-3.example.com",
    "root_password":"root-password",
    "auto_accept_cert":true
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Length: 135
Content-Type: application/json
Location: /service/v2/services/replication/targets/target-000
X-Zfssa-Replication-Api: 1.0
{
    "target": {
        "actions": 0,
        "address": "123.45.78.9:216",
        "asn": "fa5bf303-0dcb-e20d-ac92-cd129ccd2c81",
        "auto_accept_cert": true,
        "hostname": "zfs-storage-3.example.com",
        "href": "/service/v2/services/replication/targets/target-000",
        "label": "zfs-storage-3"
    }
}
```

Verify the Target Certificate

When you create a replication target, certificate verification is performed. Certificate verification consists of the following steps:

1. Certificate hostname check
2. Certificate trust check

If either the hostname check or the certificate trust check fails, the target is not created.

Hostname Check

The value of the `hostname` property can be a fully qualified domain name or an IPv4 address. The recommended value to use is the target's fully qualified domain name.

The hostname check verifies that the hostname specified in the `hostname` property for the target matches a host specified in the certificate. If you specify an IP address or an unqualified domain name for `hostname`, and the certificate only has fully qualified domain names, the hostname check fails and the target is not created.

If the target is using an ASN-based certificate, specify the target's fully qualified domain name for the value of the `hostname` property.

The hostname check is performed by default. If you set the `host_match` property to `false`, the hostname check is not performed.

For stronger security, set the value of the `hostname` property to the target's fully qualified domain name, and make sure the `host_match` property is set to `true`.

Certificate Trust Check

The certificate trust check verifies that one of the following certificates has been added to the source's trusted certificate list and is enabled for peer use:

- The target appliance's certificate

- The certificate for the certificate authority that issued the target appliance's certificate

If the certificate is not trusted, HTTP status 409 (Conflict) is returned and the target is not created. In the following example, the message line is broken for readability:

```
{
    "fault": {
        "code": 409,
        "name": "ERR_ILLEGAL_STATE",
        "message": "operation failed due to illegal state (Certificate is not trusted
                    (encountered while attempting to run command \"commit\"))"
    }
}
```

The first time you create this target for this source, the target host's certificate is not known to be trusted. Because the RESTful API cannot prompt you to confirm the certificate, set the `auto_accept_cert` property to `true` to automatically accept the target's certificate.

After the target is created, its certificate can become untrusted. For example, the source's administrator could remove the certificate from the list of trusted certificates, or the target's administrator could replace the certificate. Modify the target to set the `auto_accept_cert` property to `true` to tell the source to accept this certificate as trusted.

The certificate trust check is performed for every peer and replication connection. If the certificate is not trusted, the source rejects the connection.

To check the properties of a certificate such as fingerprint or issuer `commonName` or `SubjectAltName`, to upload a certificate, or to delete a certificate, see Managing Certificates.

Modify a Replication Target

This command modifies the properties of a replication target.

Example Request:

```
PUT /api/replication/v2/targets/target-001 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 78

{
    "hostname": "zfs-storage-3.example.com"
}
```

Delete a Replication Target

This command deletes an existing replication target.

Example Request:

```
DELETE /service/v2/services/replication/targets/target-000 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Successful delete returns HTTP status 204 (No Content).

Replication Actions

Replication actions define the rules for replicating data to replication targets. The following commands manage the replication actions.

Using the Flat Action Interface

Requests to manage replication actions can be made directly to an Oracle ZFS Storage Appliance system, without specifying a project or share.

The following table lists the base commands for managing replication actions.

Table 13-16 Base Action Interface

Request	Append to Path /api/storage/v{1|2}	Description
GET	/replication/actions	List all replication action objects
GET	/replication/actions/*ra_id*	Get the specified replication action properties
PUT	/replication/actions/*ra_id*	Modify the specified replication action object
DELETE	/replication/actions/*ra_id*	Delete the specified replication action object
PUT	/replication/actions/*ra_id*/sendupdate	Start the selected replication action
PUT	/replication/actions/*ra_id*/cancelupdate	Stop the selected replication action

The following table lists the commands for managing replication action schedules.

Table 13-17 Accessing Action Schedules

Request	Append to Path /api/storage/v{1|2}	Description
GET	/replication/actions/*ra_id*/schedules	List all replication action schedule objects
GET	/replication/actions/*ra_id*/schedules/*ra_schedule*	Get the specified replication action schedule properties
POST	/replication/actions/*ra_id*/schedules	Create a new replication action schedule
PUT	/replication/actions/*ra_id*/schedules/*ra_schedule*	Modify the specified replication action schedule object
DELETE	/replication/actions/*ra_id*/schedules/*ra_schedule*	Delete the specified replication action schedule object

The following table lists the commands for replication automatic snapshot management.

Note:

Share-level auto-snapshot schedules configured within project-level replication actions cannot be accessed with the following commands. Project-level actions can have multiple auto snapshot schedules in multiple shares, and this interface does not provide a definitive way to identify all the combinations.

Table 13-18 Accessing Replication Automatic Snapshot Configuration

Request	Append to Path /api/storage/v{1|2}	Description
GET	/replication/actions/*ra_id*/autosnaps	Retrieve auto-snapshot configurations for the selected replication action
GET	/replication/actions/*ra_id*/autosnaps/*autosnaps_id*	Get the specified replication action auto-snapshot object
PUT	/replication/actions/*ra_id*/autosnaps	Modify the specified replication action auto-snapshot properties
PUT	/replication/actions/*ra_id*/autosnaps/*autosnaps_id*	Modify the specified replication action auto-snapshot object
DELETE	/replication/actions/*ra_id*/autosnaps/*autosnaps_id*	Delete the specified replication action auto-snapshot object

The following table lists the command for Replication Action Statistics.

Table 13-19 Accessing Replication Action Statistics

Request	Append to Path /api/storage/v{1|2}	Description
GET	/replication/actions/*ra_id*/stats	Retrieve read-only replication statistics for the selected replication action

Replication Actions in Project, Filesystem, or LUN Context

Requests to manage replication actions can also be made in the context of a specific project, filesystem, or LUN.

The following table lists the base commands for managing replication actions.

- Project-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project`

- Filesystem-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem`

- LUN-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project/luns/lun`

Append the following base commands to the desired context URI listed above to manage replication actions.

Table 13-20 Project, Filesystem, or LUN Base Replication Action Interfaces

Request	Append to Project, Filesystem, or LUN URI Listed Above	Description
GET	/replication/actions	List all replication action objects
GET	/replication/actions/*ra_id*	Get the specified replication action properties
POST	/replication/actions	Create a new replication action
PUT	/replication/actions/*ra_id*	Modify the specified replication action object
DELETE	/replication/actions/*ra_id*	Delete the specified replication action object
PUT	/replication/actions/*ra_id*/sendupdate	Start the selected replication action
PUT	/replication/actions/*ra_id*/cancelupdate	Stop the selected replication action

The following table lists the base commands for managing replication schedules.

- Project-based operations URIs begin with:

  /api/storage/v{1|2}/pools/*pool*/projects/*project*

- Filesystem-based operations URIs begin with:

  /api/storage/v{1|2}/pools/*pool*/projects/*project*/filesystems/*filesystem*

- LUN-based operations URIs begin with:

  /api/storage/v{1|2}/pools/*pool*/projects/*project*/luns/*lun*

Append the following base commands to the desired context URI listed above to manage replication schedules.

Table 13-21 Project, Filesystem, or LUN Replication Action Schedules

Request	Append to Project, Filesystem, or LUN URI Listed Above	Description
GET	/replication/actions/*ra_id*/schedules	List all replication action schedule objects
GET	/replication/actions/*ra_id*/schedules/*ra_schedule*	Get the specified replication action schedule properties
POST	/replication/actions/*ra_id*/schedules	Create a new replication action schedule
PUT	/replication/actions/*ra_id*/schedules/*ra_schedule*	Modify the specified replication action schedule object
DELETE	/replication/actions/*ra_id*/schedules/*ra_schedule*	Delete the specified replication action schedule object

The following table lists the base commands for managing replication automatic snapshot configuration.

- Project-based operations URIs begin with:

`/api/storage/v{1|2}/pools/pool/projects/project`

- Filesystem-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem`

- LUN-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project/luns/lun`

Append the following base commands to the desired context URI listed above to manage replication automatic snapshot configuration.

> **Note:**
>
> Share-level auto-snapshot schedules configured within project-level replication actions cannot be accessed with the following project-based operations. Project-level actions can have multiple auto-snapshot schedules in multiple shares, and this interface does not provide a definitive way to identify all combinations.

Table 13-22 Project, Filesystem, or LUN Replication Automatic Snapshot Configuration

Request	Append to Project, Filesystem, or LUN URI Listed Above	Description
GET	/replication/actions/*ra_id*/autosnaps	Retrieve auto-snapshot configurations for a project/share's selected replication action
GET	/replication/actions/*ra_id*/autosnaps/*autosnaps_id*	Get a project/share's specified replication action auto-snapshot configuration
POST	/replication/actions/*ra_id*/autosnaps	Create a new project/share's level replication action auto-snapshot object
PUT	/replication/actions/*ra_id*/autosnaps	Modify a project/share's specified replication action's target auto-snapshot retention policy.
PUT	/replication/actions/*ra_id*/autosnaps/*autosnaps_id*	Modify the specified replication action auto-snapshot object
DELETE	/replication/actions/*ra_id*/autosnaps/*autosnaps_id*	Delete the specified replication action auto-snapshot object

The following table lists the base command for accessing replication action statistics.

- Project-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project`

- Filesystem-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project/filesystems/filesystem`

- LUN-based operations URIs begin with:

  `/api/storage/v{1|2}/pools/pool/projects/project/luns/lun`

Append the following base commands to the desired context URI listed above to access replication action statistics.

Table 13-23 Accessing Replication Action Statistics

Request	Append to Project, Filesystem, or LUN URI Listed Above	Description
GET	/replication/actions/*ra_id*/stats	Retrieve read-only replication statistics for the selected replication action

List Replication Actions

Gets a list of all available replication actions.

Example Request:

```
GET /api/storage/v2/replication/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 529

{
    "actions": [{
        "href": ""
        ...
    }, {
        "href": "",
        ...
    }]
}
```

Get Replication Action

The `get replication action status` command returns the status of a single replication action given by its ID.

Example Request:

```
GET /api/storage/v2/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 529

{
    "action": {
        "average_throughput": 0.0,
        "bytes_sent": 0.0,
        "collection": "local",
        "compression": true,
        "continuous": false,
        "enabled": true,
        "estimated_size": 0.0,
        "estimated_time_left": 0.0,
        "href": "/api/storage/v2/replication/actions",
        "id": "8373d331-de60-e590-90e8-9ad69fcb4aec",
        "include_clone_origin_as_data": false,
        "include_snaps": true,
        "last_sync": "20130916T21:36:50",
        "last_try": "20130916T21:36:50",
        "max_bandwidth": 0,
        "pool": "p1",
        "project": "proj-01",
        "retain_user_snaps_on_target": false,
        "share": "fs1",
        "state": "sending",
        "target": "38094753-6c90-49ed-aa92-995a296d432a",
        "use_ssl": true
    }
}
```

Example Request:

The following replication action response shows an example recovery point objective (RPO) and related replica lag warning and alerts.

```
GET /api/storage/v2/replication/actions HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type:application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 529

{
    "action": {"id": "12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
        "target_id": "4fd305ac-4af5-c34a-87c3-88203207305b",
        ...
        "replica_lag": "42:25:31",
        "recovery_point_objective": 0,
        "replica_lag_warning_alert": 0,
        "replica_lag_error_alert": 0,
        "replica_lag_over_warning_limit": false,
        "replica_lag_over_error_limit": false,
        "project": "default"
    }
}
```

Create Replication Action

Creates a new replication action.

Create Properties:

```
Initial values:
                          target = cleo
                         enabled = true
                      continuous = false
                   include_snaps = true
     retain_user_snaps_on_target = false
                           dedup = true
    include_clone_origin_as_data = false
                   max_bandwidth = unlimited
                      bytes_sent = 0
                  estimated_size = 0
             estimated_time_left = 0
              average_throughput = 0
                         use_ssl = true
                     compression = on
```

Example Request:

```
POST /api/storage/v2/replication/actions HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Content-Length: 121
Accept: application/json

{
    "pool": "p1",
    "project": "proj-01",
    "share": "fs1",
    "target_pool": "pool1",
    "target": "38094753-6c90-49ed-aa92-995a296d432a"
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Length: 506
Content-Type: application/json
Location: /api/storage/v2/replication/action/8373d331-de60-e590-90e8-9ad69fcb4aec
X-Zfssa-Replication-Api: 1.0

{
    "action": {
        "project": "blue1",
        "target": "38094753-6c90-49ed-aa92-995a296d432a",
        "bytes_sent": 0.0,
        "compression": true,
        "continuous": false,
        "enabled": true,
        "dedup": false,
        "max_bandwidth": 0,
        "collection": "local",
        "estimated_size": 0.0,
        "state": "idle",
        "href": "/api/storage/v2/replication/pools/p1/projects/blah1/shares/fs1/
                  actions/8373d331-de60-e590-90e8-9ad69fcb4aec",
        "average_throughput": 0.0,
        "use_ssl": true,
        "estimated_time_left": 0.0,
        "retain_user_snaps_on_target": false,
        "share": "fs1",
        "id": "8373d331-de60-e590-90e8-9ad69fcb4aec",
        "pool": "p1",
        "include_clone_origin_as_data": false,
        "include_snaps": true
    }
}
```

Creates a schedule for a replication action.

Example Request:

```
POST /api/storage/v2/replication/actions/b77bd8cd-17ed-69da-9e4b-aebe3cc63755/
schedules HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: */*
Content-Type:application/json
Content-Length: 65

{
    "frequency":"month",
    "day":"5th",
    "hour":"auto",
    "minute":"auto"
}
```

Example Result:

```
HTTP/1.1 201 Created
Date: Thu, 12 Jan 2017 17:35:48 GMT
Server: TwistedWeb/192.0.2
Content-Length: 0
X-Zfssa-Storage-Api: 1.1
Content-Type: application/json; charset=utf-8
X-Zfssa-Api-Version: 1.0
X-Zfssa-Version: user/generic@2016.12.08,1-0
```

Modify Replication Action

Modifies an existing replication action.

Example Request:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

{
    "use_ssl": false
}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 620

{
    "action": {
        "target_id": "407642ae-91b5-681c-de5e-afcd5cbf2974",
        "compression": true,
        "continuous": false,
        "enabled": true,
        "max_bandwidth": 0,
        "dedup": false,
        "retain_user_snaps_on_target": false,
        "use_ssl": false,
        "id": "c141d88d-ffd2-6730-d489-b71905f340cc",
        "include_clone_origin_as_data": false,
        "include_snaps": true
    }
}
```

Example Request:

```
PUT /api/storage/v2/replication/actions/action_id -d
'{"recovery_point_objective": 60}HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
```

Example Result:

```
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 620

{
    "action": {
        "state_description": "Idle (no update in progress)",
        "recovery_point_objective": 60,
        "replica_lag_over_warning_limit": false,
        "bytes_sent": "0",
        "last_try": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
        "max_bandwidth": 0,
        "estimated_size": "0",
        "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-
a6b8263a0f6a",
        "estimated_time_left": 0,
        "use_ssl": true,
        "id": "12d981c3-b098-c65a-e1e9-a6b8263a0f6a",
        "stats": {"total_logical_bytes": 40656,
        "last_dd_table_build": 9169029,
        "total_after_dedup": 18476,
        "last_try": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
        "dd_total_updates": 1,
        "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-
a6b8263a0f6a/stats",
        "dd_total_duration": 47149245470,
        "last_logical_bytes": 40656,
        "dd_total_table_mem": 2097152,
        "last_result": "success",
        "last_after_dedup": 18476,
        "last_duration": 47149245470,
        {"dd_total_logical_bytes": 40656,
        "total_updates": 1,
        "total_duration": 47149245470,
        "replica_data_timestamp": "Mon Nov 21 2016 23:25:12 GMT+0000 (UTC)",
        "total_to_network": 9623,
        "dd_total_table_build": 9169029,
        "dd_total_phys_bytes": 16800,
        "last_to_network": 9623,
        "total_phys_bytes": 16800,
        "last_phys_bytes": 16800,
        "last_sync": "Mon Nov 21 2016 23:25:59 GMT+0000 (UTC)",
        "last_dd_table_mem": 2097152,
        "dd_total_after_dedup": 18476,
        "dd_total_to_network": 9623},
        "compression": "on",
        "dedup": true,
        "replica_lag_warning_alert": 0,
        "last_result": "success",
        "include_clone_origin_as_data": false,
        "state": "idle",
        "offline": false,
        "export_path": "",
        "export_pending": false,
        "autosnaps": {"autosnaps_retention_policies":
        "synchronized",
        "href": "/api/storage/v2/replication/actions/12d981c3-b098-c65a-e1e9-
a6b8263a0f6a/autosnaps"},
        "replica_data_timestamp": "Mon Nov 21 2016 23:25:12 GMT+0000 (UTC)",
        "continuous": false,
        "target_id": "4fd305ac-4af5-c34a-87c3-88203207305b",
        {"average_throughput": "0B/s",
        "next_update": "Sync now",
        "pool": "p1",
        "replica_lag_over_error_limit": false,
        "target": "pool1",
        "replica_lag": "42:28:24",
        "retain_user_snaps_on_target": false,
        ...
    }
}
```

Monitor Replication Action Progress

The `get replication action status` command returns the status of a single replication action given by its ID. Examine `state` and `state_description` to determine replication progress.

`state` property values:

- `sending`
- `idle`

`state_description` property values:

- `Connecting to replication target`
- `Receiving checkpoint from target`
- `Estimating size of update`
- `Building deduplication tables`

  This property value is only for deduplicated replication streams.

Example Request:

```
GET /api/storage/v2/replication/actions/1438ed7f-aad3-c631-d869-9e85cd7f15b4
HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 529

{
    "action": {
       "id": "1438ed7f-aad3-c631-d869-9e85cd7f15b4",
       "target_id": "4fd3483e-b1f5-4bdc-9be3-b3a4becd0c42",
       "target": "cleo",
       "pool": "p0",
       "replication_of": "testproj",
       "enabled": true,
       "continuous": false,
       "include_snaps": true,
       "retain_user_snaps_on_target": false,
       "dedup": true,
       "include_clone_origin_as_data": false,
       "max_bandwidth": -1,
       "bytes_sent": 0,
       "estimated_size": 0,
       "estimated_time_left": 0,
       "average_throughput": 0,
       "use_ssl": true,
       "compression": "on",
       "export_path": "",
       "state": "sending",
       "state_description": "Receiving checkpoint from target",
       "export_pending": false,
       "offline": false,
       "next_update": "Sync now",
       "replica_data_timestamp": "Thu Apr 28 2016 22:38:03 GMT+0000 (UTC)",
       "last_sync": "<unknown>",
       "last_try": "<unknown>",
       "last_result": "<unknown>",
       "replica_lag": "00:00:18",
       "recovery_point_objective": 0,
       "replica_lag_warning_alert": 0,
       "replica_lag_error_alert": 0,
       "replica_lag_over_warning_limit": false,
       "replica_lag_over_error_limit": false,
       "project": "testproj"
    }
}
```

Cancel Update

Cancels an in-progress replication update.

Example Request:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/
cancelupdate HTTP/1.1
```

```
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

Send Update

Schedules a replication update to start as soon as possible.

Example Request:

```
PUT /api/storage/v2/replication/actions/c141d88d-ffd2-6730-d489-b71905f340cc/
sendupdate HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

Delete a Replication Action

Deletes an existing replication action.

Example Request:

```
DELETE /api/storage/v2/replication/actions/e7e688b1-ff07-474f-d5cd-cac08293506e
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Successful delete returns HTTP status 204 (No Content).

Example Result:

```
HTTP/1.1 204 No-Content
X-Zfssa-Replication-Api: 1.0
```

Replication Packages

This section details replication package and source commands.

Table 13-24   Replication Package Commands

Request	Append to Path /api/storage/v{1|2}/replication	Description
GET	/packages	List all replication packages
GET	/packages/*package*	Get the specified replication package
PUT	/packages/*package*	Modify the specified replication package
DELETE	/packages/*package*	Destroy the specified replication package
PUT	/packages/*package*/cancelupdate	Run cancelupdate on the specified package
PUT	/packages/*package*/sever	Run sever on the specified package

Table 13-24 (Cont.) Replication Package Commands

Request	Append to Path /api/storage/v{1|2}/replication	Description
PUT	/packages/*package*/pkgreverse	Run reverse on the specified package
PUT	/packages/*package*/clone	Clone the specified package
GET	/packages/*package*/clone/conflicts	List share property conflicts
GET	/packages/*package*/projects	List package projects
GET	/packages/*package*/projects/*project*	Get package project
PUT	/packages/*package*/projects/*project*	Modify package project
GET	/packages/*package*/projects/*project*/usage/groups	Get package project group usage
GET	/packages/*package*/projects/*project*/usage/users	Get package project users usage
GET	/packages/*package*/projects/*project*/snapshots	List all snapshot objects
GET	/packages/*package*/projects/*project*/snapshots/*snapshot*	Get the specified snapshot properties
DELETE	/packages/*package*/projects/*project*/snapshots/*snapshot*	Destroy the specified snapshot object
PUT	/packages/*package*/projects/*project*/snapshots/*snapshot*	Rename the package project snapshot
GET	/packages/*package*/projects/*project*/automatic	List all package project automatic snapshot objects
GET	/packages/*package*/projects/*project*/automatic/*automatic*	Get the specified package project automatic snapshot properties
GET	/packages/*package*/projects/*project*/filesystems	List package filesystems
GET	/packages/*package*/projects/*project*/filesystems/*filesystem*	Get package filesystem
PUT	/packages/*package*/projects/*project*/filesystems/*filesystem*	Modify package filesystem
GET	/packages/*package*/projects/*project*/filesystems/*filesystem*/usage/groups	Get package filesystem group usage
GET	/packages/*package*/projects/*project*/filesystems/*filesystem*/usage/users	Get package filesystem users usage
GET	/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Get the specified snapshot properties
GET	/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots	List all snapshot objects
DELETE	/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Destroy the specified snapshot object
PUT	/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Rename the package filesystem snapshot
GET	/packages/*package*/projects/*project*/filesystems/*filesystem*/automatic	List all package filesystem automatic snapshot objects
GET	/packages/*package*/projects/*project*/filesystems/*filesystem*/automatic/*automatic*	Get the specified package filesystem automatic snapshot properties
GET	/packages/*package*/projects/*project*/luns	List package LUNs
GET	/packages/*package*/projects/*project*/luns/*lun*	Get package LUN

Table 13-24 (Cont.) Replication Package Commands

Request	Append to Path /api/storage/v{1|2}/replication	Description
PUT	/packages/*package*/projects/*project*/luns/*lun*	Modify package LUN
GET	/packages/*package*/projects/*project*/luns/*lun*/usage/groups	Get package LUN group usage
GET	/packages/*package*/projects/*project*/luns/*lun*/usage/users	Get package LUN users usage
GET	/packages/*package*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Get the specified snapshot properties
GET	/packages/*package*/projects/*project*/luns/*lun*/snapshots	List all snapshot objects
DELETE	/packages/*package*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Destroy the specified snapshot object
PUT	/packages/*package*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Rename the package LUN snapshot
GET	/packages/*package*/projects/*project*/luns/*lun*/automatic	List all package LUN automatic snapshot objects
GET	/packages/*package*/projects/*project*/luns/*lun*/automatic/*automatic*	Get the specified package LUN automatic snapshot properties

Replication sources and their corresponding packages can also be accessed using the following commands.

Table 13-25 Replication Source Commands

Request	Append to Path /api/storage/v{1|2}/replication/sources	Description
GET	Use only /api/storage/v{1|2}/replication/sources	List replication sources
GET	/*source*	List replication source details
GET	/*source*/packages/*package*	Get the specified replication package
PUT	/*source*/packages/*package*	Modify the specified replication package
DELETE	/*source*/packages/*package*	Destroy the specified replication package
PUT	/*source*/packages/*package*/cancelupdate	Run `cancelupdate` on the specified package
PUT	/*source*/packages/*package*/sever	Run sever on the specified package
PUT	/*source*/packages/*package*/pkgreverse	Run reverse on the specified package
PUT	/*source*/packages/*package*/clone	Clone the specified package
GET	/*source*/packages/*package*/clone/conflicts	List share property conflicts
GET	/*source*/packages/*package*/projects	List package projects
GET	/*source*/packages/*package*/projects/*project*	Get package project
PUT	/*source*/packages/*package*/projects/*project*	Modify package project
GET	/*source*/packages/*package*/projects/*project*/usage/groups	Get package project group usage
GET	/*source*/packages/*package*/projects/*project*/usage/users	Get package project users usage

Table 13-25 (Cont.) Replication Source Commands

Request	Append to Path /api/storage/v{1|2}/replication/ sources	Description
GET	/*source*/packages/*package*/projects/*project*/snapshots/*snapshot*	Get the specified snapshot properties
GET	/*source*/packages/*package*/projects/*project*/snapshots	List all snapshot objects
DELETE	/*source*/packages/*package*/projects/*project*/snapshots/*snapshot*	Destroy the specified snapshot object
PUT	/*source*/packages/*package*/projects/*project*/snapshots/*snapshot*	Rename the package project snapshot
GET	/*source*/packages/*package*/projects/*project*/automatic	List all package project automatic snapshot objects
GET	/*source*/packages/*package*/projects/*project*/automatic/*automatic*	Get the specified package project automatic snapshot properties
GET	/*source*/packages/*package*/projects/*project*/filesystems	List package filesystems
GET	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*	Get package filesystem
PUT	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*	Modify package filesystem
GET	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/usage/groups	Get package filesystem group usage
GET	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/usage/users	Get package filesystem users usage
GET	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Get the specified snapshot properties
GET	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots	List all snapshot objects
DELETE	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Destroy the specified snapshot object
PUT	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/snapshots/*snapshot*	Rename the package filesystem snapshot
GET	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/automatic	List all package filesystem automatic snapshot objects
GET	/*source*/packages/*package*/projects/*project*/filesystems/*filesystem*/automatic/*automatic*	Get the specified package filesystem automatic snapshot properties
GET	/*source*/packages/*package*/projects/*project*/luns	List package LUNs
GET	/*source*/packages/*package*/projects/*project*/luns/*lun*	Get package LUN
PUT	/*source*/packages/*package*/projects/*project*/luns/*lun*	Modify package LUN
GET	/*source*/packages/*package*/projects/*project*/luns/*lun*/usage/groups	Get package LUN group usage
GET	/*source*/packages/*package*/projects/*project*/luns/*lun*/usage/users	Get package LUN users usage

Table 13-25 (Cont.) Replication Source Commands

Request	Append to Path /api/storage/v{1|2}/replication/sources	Description
GET	*/source*/packages/*package*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Get the specified snapshot properties
GET	*/source*/packages/*package*/projects/*project*/luns/*lun*/snapshots	List all snapshot objects
DELETE	*/source*/packages/*package*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Destroy the specified snapshot object
PUT	*/source*/packages/*package*/projects/*project*/luns/*lun*/snapshots/*snapshot*	Rename the package LUN snapshot
GET	*/source*/packages/*package*/projects/*project*/luns/*lun*/automatic	List all package LUN automatic snapshot objects
GET	*/source*/packages/*package*/projects/*project*/luns/*lun*/automatic/*automatic*	Get the specified package LUN automatic snapshot properties

List Replication Sources

Lists all available replication sources.

Example Request:

```
GET /api/storage/v2/replication/sources HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Output:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 529

{
    "sources": [{
        "asn": "314d252e-c42b-e844-dab1-a3bca680b563",
        "href": "/api/storage/v2/replication/sources/zfs-repl-host",
        "ip_address": "ipaddr-1",
        "name": "zfs-repl-host",
        "source": "source-000"
    }]
}
```

List Replication Packages

Lists all of the replication packages.

Example Request:

```
GET /api/storage/v2/replication/packages HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 529

{
  "packages": [
    {
      "href": "/api/storage/v2/replication/packages/0efaab49-7b22-4d4a-
def8-813c27780894",
      "id": "0efaab49-7b22-4d4a-def8-813c27780894",
      "source_name": "sourceA",
      "source_asn": "8a22f6e0-4ee4-4b85-f141-e152f5fac961",
      "source_ip": "ipaddr-1",
      "source_pool": "poolA",
      "target_pool": "poolA",
      "replica_of": "projTest",
      "enabled": true,
      "state": "idle",
      "state_description": "Idle (no update in progress)",
      "offline": false,
      "import_path": "",
      "data_timestamp": "2017-03-09T22:36:12Z",
      "last_sync": "2017-03-09T22:36:22Z",
      "last_try": "2017-03-09T22:36:22Z",
      "last_result": "success"
    }
  ]
}
```

Modify Package

Modifies the package properties.

Property	Type	Description
`enabled`	boolean	Current state of replication updates

Example Request:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

{
    "enabled": false
}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

Example Request:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/
pkgreverse HTTP/1.1
Host: zfs-storage.example.com:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json

{
    "new_project_name":"restrev",
    "enable_action_upon_reversal":"true"
}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

Delete Package

Destroys a replication package.

Example Request:

```
DELETE /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Successful delete returns HTTP status 204 (No Content).

Example Result:

```
HTTP/1.1 204 No-Content
X-Zfssa-Replication-Api: 1.0
```

Cancel Update

Cancels an ongoing update for this package.

Example Request:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/
cancelupdate HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

If no update is in progress, HTTP status 409 (Conflict) is returned.

Example Result:

```
HTTP/1.1 409 Conflict
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 137

{
    "cancelupdate": {
        "AKSH_ERROR": "EAK_NAS_REPL_BADSTATE",
        "message": "operation illegal for state"
    }
}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

Clone Package

Clones the package project.

Example Request:

```
PUT /api/v2/storage/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/
clone HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

Successful clone returns HTTP status 202 (Accepted). A helper command can be used to determine whether there are conflicts with the clone operation.

Example Clone Conflicts Request:

```
GET /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/
clone/conflicts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Clone/conflicts Returns Conflicts:

```
HTTP/1.1 200 OK
X-Zfssa-Replication-Api: 1.0
Content-Type: application/json
Content-Length: 58

{
    "conflicts": "There are no conflicts."
}
```

Properties:

```
Default settings:
            target_project = (unset)
       original_mountpoint = /export
       override_mountpoint = false
                mountpoint =
```

Sever Package

Severs a replication connection and moves the package contents into new project. This action permanently severs this package and its replicated shares from the source system, making them local projects on this system. Subsequent replication updates in either direction requires defining new actions and sending a full update.

Example Request:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/
sever HTTP/1.1
Host: zfs-storage.example.com:215
```

```
Authorization: Basic Tm8gcGVla2luZyE=

{
    "projname":"restsev"
}
```

Success Response:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

Reverse Package

Reverses the direction of replication. This action disables replication for this package and moves the contents of this package into a new local project configured to replicate back to the source. Any metadata or data changes made on the source since the last successful update are lost when the new project is first replicated back to the source.

Example Request:

```
PUT /api/storage/v2/replication/packages/8373d331-de60-e590-90e8-9ad69fcb4aec/reverse
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=

{
    "projname":"restrev"
}
```

Success Response:

```
HTTP/1.1 202 Accepted
X-Zfssa-Replication-Api: 1.0
```

14 Storage Encryption

Pools, projects, and shares can be encrypted. If a pool is encrypted, then each child project and share is encrypted and inherits encryption property values from the pool. If encryption property values are specified for a project, then the inherited values are not used, and child shares inherit the local values specified for the project. An encrypted project can be created in an unencrypted pool. An unencrypted project cannot be created in an encrypted pool.

If a project is encrypted, then each child share is encrypted and inherits encryption property values from the project. If encryption property values are specified for a share, then the inherited values are not used. An encrypted share can be created in an unencrypted project. An unencrypted share cannot be created in an encrypted project.

Create an Encrypted Pool, Project, or Share

To create an encrypted pool, project, or share, specify values for the `encryption`, `keystore`, and `keyname` properties, in addition to the required properties described in Configure Pool, Create Project, Create Filesystem, and Create a New LUN.

The following table describes encryption properties for a particular pool, project, or share.

Table 14-1 Encryption Properties for Pools, Projects, and Shares

Property	Type	Description
`encryption`	string	AES encryption type and key length
`keystore`	string	Type of keystore: `local`, `okm`, or `kmip`
`keyname`	string	A specific key name
`keylastchanged`	string	(Read-only) The date that the key was last changed. This value is in ISO-8601 datetime format in v2 and in Javascript datetime format in v1. If the value is empty, then this key has not been changed since it was created.
`keystatus`	string	(Read-only) Either `available` or `unavailable`, or `none`. If the value of this property is `unavailable`, then the key has been deleted.

Example Request:

```
POST /api/storage/v2/pools/p1/projects HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "name": "proj-enc",
    "encryption": "aes-128-ccm",
    "keystore": "local",
    "keyname": "Key-0"
}
```

Example Result:

```
HTTP/1.1 201 Created
Content-Type: application/json
Location: http://zfs-storage.example.com:215/pools/p1/projects/proj-enc

{
    "project": {
        "name": "proj-enc",
        "href": "/api/storage/v2/pools/p1/projects/proj-enc",
        ...
        "encryption": "aes-128-ccm",
        "keystore": "local",
        "keychangedate": "",
        "keystatus": "available",
        "keyname": "Key-0",
        ...
    }
}
```

Manage Encryption Keys

> **Note:**
>
> Encryption is a licensed feature for certain models. For details, refer to the "Oracle Software License Agreement ("SLA") and Entitlement for Hardware Systems with Integrated Software Options" and the Licensing Information User Manual for the software release.

Oracle ZFS Storage Appliance offers transparent data encryption for pools, projects, and individual shares (filesystems and LUNs). The appliance includes a built-in Local keystore, and also supports Oracle Key Manager (OKM), and Key Management Interoperability Protocol (KMIP) encryption. Each encrypted project or share requires a wrapping key from the Local, OKM, or KMIP keystore. The data encryption keys are managed by the storage appliance and are stored persistently encrypted by the wrapping key.

The encryption key must be created before you can create an encrypted pool, project, or share.

- Because the keystore must be configured before the pool is created, you cannot create an encrypted pool at initial system configuration or after factory reset.
- Before setting up replication for a share or project in a encrypted pool, ensure that the encryption key used at the source is also available at the target.

The following table describes the RESTful API requests available to manage encryption keys. In the table, the value of *keystore* is `local`, `okm`, or `kmip`. The value of *key* is the value of the `keyname` property. Key properties are described in List Encryption Keys.

Table 14-2 Encryption Key Operations

Request	Append to Path /api/storage/v{1|2}	Description
GET	/encryption/*keystore*	List all *keystore* properties
PUT	/encryption/*keystore*	Modify *keystore* properties
GET	/encryption/*keystore*/keys	List all *keystore* keys
GET	/encryption/*keystore*/keys/*key*	Get details about the specified key
POST	/encryption/*keystore*/keys	Create a key
DELETE	/encryption/*keystore*/keys/*key*	Destroy a key
GET	/encryption/*keystore*/keys/*key*/dependents	List the shares that are dependent on this key

Configure a Local Keystore

To configure a Local keystore, set the master passphrase. For a Local keystore, `master_passphrase` is the only property that you can list or modify.

Check whether `master_passphrase` is set:

```
GET /api/storage/v2/encryption/local HTTP/1.1
```

Output:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "keystore": {
        "href": "/api/storage/v2/encryption/local",
        "master_passphrase": false,
        "keys": []
    }
}
```

Specify a value for `master_passphrase`:

```
PUT /api/storage/v2/encryption/local HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "master_passphrase": "passphrase"
}
```

Confirm that `master_passphrase` is set:

```
GET /api/storage/v2/encryption/local HTTP/1.1
```

Output:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "keystore": {
        "href": "/api/storage/v2/encryption/local",
        "master_passphrase": true,
        "keys": []
    }
}
```

Configure an OKM Keystore

The following table describes the properties that must be set to configure an OKM keystore.

Table 14-3 OKM Keystore Properties

Property	Type	Description
`agent_id`	string	Agent ID
`registration_pin`	string	This value is supplied by your OKM security officer
`server_addr`	string	IP address of your OKM server

Check whether the above properties are set:

```
GET /api/storage/v2/encryption/okm HTTP/1.1
```

Output:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "keystore": {
        "href": "/api/storage/v2/encryption/okm",
        "agent_id": "",
        "registration_pin": false,
        "server_addr": "",
        "keys": []
    }
}
```

Specify values for `agent_id`, `registration_pin`, and `server_addr`:

```
PUT /api/storage/v2/encryption/okm HTTP/1.1
Host: zfs-storage.example.com:215
Content-Type: application/json
Accept: application/json

{
    "agent_id": "agent-id",
    "registration_pin": "reg-pin",
    "server_addr": "ipaddr"
}
```

Configure a KMIP Keystore

The KMIP keystore is used in conjunction with KMIP-compliant servers, including Oracle Key Vault. Oracle Key Vault is a software appliance that is installed on a dedicated server and that supports the OASIS KMIP standard.

To configure encryption using KMIP, upload the key and certificates that you received from your KMIP administrator as described in Upload a Key or Certificate.

After you have uploaded the key and certificates, specify the KMIP server, client certificate, and a key name.

The following table describes the properties to set to configure a KMIP keystore.

Table 14-4   KMIP Keystore Properties

Property	Type	Description
`server_list`	list	IP address or host name of a KMIP server; this property can have multiple values
`client_cert`	string	Certificate that you created from files provided by your KMIP server administrator
`host_match`	boolean	Validate the server hostname against the server's identity in the server certificate
`destroy_key_on_remove`	boolean	Destroy or preserve a key on the KMIP server when that key is deleted key on the appliance

For more information about `host_match` and `destroy_key_on_remove`, see Key Management Interoperability Protocol (KMIP) Keystore in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Check whether the above properties are set:

```
GET /api/storage/v2/encryption/kmip HTTP/1.1
```

Output:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
  "keystore": {
    "href": "/api/storage/v2/encryption/kmip",
    "server_list": [
      "ipaddr-or-hostname"
    ],
    "client_cert": "134a9138-29a0-4720-80bb-ec2b13457c39",
    "host_match": false,
    "destroy_key_on_remove": true,
    "keys": [],
... detailed information about the private key, certificate, and certificate
authority ...
```

```
  }
}
```

Create an Encryption Key

After the keystore is configured, to create a key, simply set the key name. The following example creates a new KMIP key. See List Encryption Keys for example results.

Example Request:

```
POST /api/storage/v2/encryption/kmip/keys HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Type: application/json
Accept: application/json
{
        "keyname": "atz-1-27-2021"
}
```

List Encryption Keys

This command lists the properties of all encryption keys. HTTP status 200 (OK) is returned for a successful command. The HTTP body contains an array of keys in JSON format. The following table describes encryption key properties.

Table 14-5 Encryption Key Properties

Property	Type	Description
cipher	string	AES encryption type
key	string	(Local only) Hex-encoded raw 256-bit key, stored in an encrypted form; this value is generated automatically if you do not specify a value
keyname	string	A specific key
href	string	The path to the key

The following example lists all Local, OKM, and KMIP keys.

Example Request:

```
GET /api/storage/v2/encryption/local/keys /api/storage/v2/encryption/okm/
keys /api/storage/v2/encryption/kmip/keys HTTP/1.1

GET /api/storage/v2/encryption/local/keys /api/storage/v2/encryption/okm/keys
HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Accept: application/json
```

Example Result:

This result shows no Local or OKM keys exist on this Oracle ZFS Storage Appliance system, and one KMIP key exists.

```
{
    "keys": {
    }
}
{
    "keys": {
    }
}
{
    "keys": [{
        "cipher": "AES",
        "keyname": "atz-1-27-2021",
        "href": "/api/storage/v2/encryption/kmip/keys/key-000"
    }]
}
```

List Storage that is Encrypted with the Specified Key

The `dependents` query lists any shares, projects, or pools that are encrypted with the specified key.

The following example shows that only filesystem `fs-enc` in the default pool is encrypted with the `atz-1-27-2021` KMIP key.

Example Request:

```
GET /api/storage/v2/encryption/kmip/keys/atz-1-27-2021/dependents HTTP/1.1
```

Example Result:

```
{
  "dependents": [
    "pool-0/local/default/fs-enc"
  ]
}
```

Delete a Key

To delete a key, use the value of the `href` property of the key (not the key name). Successful deletion returns HTTP Status 204 (No Content).

When a key is deleted, all of the data in all of the pools and shares that use the key becomes inaccessible. This is equivalent to secure data destruction and is permanent and irrevocable unless you have prepared for key restoration by backing up the key.

Example Request:

```
DELETE /api/storage/v2/encryption/kmip/keys/key-000 HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

15 System Commands

System commands are used to obtain Oracle ZFS Storage Appliance system identity information, and to perform top-level system management commands. The following table lists the available system commands.

Appliance System Commands

The following system commands are available.

Table 15-1 Oracle ZFS Storage Appliance System Commands

Request	Append to Path /api/system/v{1|2}	Description
GET	/version	List the Oracle ZFS Storage Appliance hardware and software version information
PUT	/reboot	Reboot Oracle ZFS Storage Appliance; any queued platform updates will be applied during this reboot
PUT	/reboot?skip_update=true	Reboot Oracle ZFS Storage Appliance without applying any queued platform updates
PUT	/reboot?diag=true	Diagnostic reboot: Reboot Oracle ZFS Storage Appliance, gathering additional diagnostic information in the process
PUT	/poweroff	Turn off Oracle ZFS Storage Appliance
PUT	/restart	Restart the management interface and gather diagnostic information
PUT	/factoryreset	Reset the Oracle ZFS Storage Appliance configuration back to factory settings
GET	/disks	List all system disks
GET	/disks/*disk*	List the specified system disk properties
GET	/memory	System memory status report

Get Version

This command returns a system structure that contains system identity information. HTTP status 200 (OK) is returned for a successful command.

Example Request:

```
GET /api/system/v1/version HTTP/1.1
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
Content-Type: application/json

{
    "version": {
        "hw_csn": "1211FM2009",
        "updated": "20130528T16:21:17",
        "fw_vendor": "American Megatrends Inc.",
        "os_isa": "i386",
        "os_boot": "20130528T16:25:44",
        "hw_product": "Sun Netra X4270 M3",
        "http_version": "Apache/2.2.24 (Unix)",
        "hw_asn": "2f4aeeb3-b670-ee53-e0a7-d8e0ae410749",
        "ssl_version": "OpenSSL 1.0.0k 5 Feb 2013",
        "os_machine": "i86pc",
        "os_nodename": "admin1",
        "os_version": "nas/generic@2013.05.16,1-0",
        "ak_product": "SUNW,iwashiG2",
        "fw_version": "21000208",
        "os_release": "5.11",
        "installed": "20130411T19:50:16",
        "sp_version": "3.1.2.0",
        "os_platform": "i86pc",
        "fw_release": "10/22/2012"
    }
}
```

Power Off System

This command performs a clean shutdown of Oracle ZFS Storage Appliance. All data services become permanently unavailable unless the appliance is part of a cluster. To power the system back on requires either service processor access or physical access to the power switch. This command runs asynchronously and returns an HTTP status of 202 (Accepted). The appliance must be monitored to follow the status of the actual command.

Example Request:

```
PUT /api/system/v1/poweroff HTTP/1.1
Host: zfs-storage.example.com:215
```

Reboot System

This command performs a clean power cycle of Oracle ZFS Storage Appliance. All services are temporarily unavailable. This command runs asynchronously and returns HTTP status 202 (Accepted). The appliance must be monitored to follow the status of the actual command.

> **Note:**
>
> If a pending platform update is available to the appliance, it will be applied during this reboot. To perform a reboot without applying the pending platform update, use the `/reboot?skip_update=true` command instead.

Example Request:

```
PUT /api/system/v1/reboot HTTP/1.1
Host: zfs-storage.example.com:215
```

Example Request:

```
PUT /api/system/v1/reboot?skip_update=true HTTP/1.1
Host: zfs-storage.example.com:215
```

Restart System Management

This command restarts the management interface and gathers diagnostic information. This command runs asynchronously and returns HTTP status 202 (Accepted). The Oracle ZFS Storage Appliance system must be monitored to follow the status of the actual command.

Example Request:

```
PUT /api/system/v1/restart HTTP/1.1
Host: zfs-storage.example.com:215
```

Diagnostic Reboot

This command reboots Oracle ZFS Storage Appliance, gathering additional diagnostic information in the process. This command runs asynchronously and returns HTTP status 202 (Accepted). The appliance must be monitored to follow the status of the actual command.

> **Note:**
>
> If there is a pending platform update available to the appliance, it will be not applied during this diagnostic reboot.

Example Request:

```
PUT /api/system/v1/reboot?diag=true HTTP/1.1
Host: zfs-storage.example.com :215
```

Factory Reset

This command restores the Oracle ZFS Storage Appliance configuration to the original factory settings. All configuration changes are lost, and the appliance must be taken through initial setup as when first installed. This command runs asynchronously and returns HTTP status 202 (Accepted). The appliance must be monitored to follow the status of the actual command. Because this command can result in a loss of all configuration data, the query parameter `?confirm=true` must be added for the command to succeed.

Example Request:

```
PUT /api/system/v1/factoryreset?confirm=true HTTP/1.1
Host: zfs-storage.example.com:215
```

System Support Bundles

The following support bundle commands are available.

Table 15-2 Support Bundle Commands

Request	Append to Path /api/system/v{1|2}	Description
GET	/bundles	List all support bundles
GET	/bundles/*bundle*	Get the specified bundle data or properties
POST	/bundles	Build a support bundle and upload it to Oracle Support
PUT	/bundles/*bundle*/retry	Retry upload of the specified bundle
PUT	/bundles/*bundle*/cancel	Cancel upload of the specified bundle
PUT	/bundles/*bundle*/send	Upload the specified bundle to Oracle Support with an optional Service Request (SR) number
DELETE	/bundles/*bundle*	Destroy the specified bundle

Create Support Bundle

Creates a new support bundle to help resolve a service request. A Service Request (SR) number must be supplied to associate the support bundle with the open service request and send it to Oracle Support. The SR number must be in `3-`*`nnnnnnnnnn`* format. For the support bundle to be automatically uploaded to Oracle Support, the Phone Home settings must be registered with valid My Oracle Support (MOS) credentials that have upload permissions.

Example Request:

```
POST /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23

{"srn": "3-0123456789"}
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
```

If a Service Request Number (SRN) is not provided, the system will build a local bundle instead.

Example Request:

```
POST /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 23
```

Example Result:

```
{
  "bundle": {
    "status": "",
    "uuid": "d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b",
    "associated_bundle": [
      {
        "href": "/api/system/v1/bundles/4050963a-4082-663f-99c0-fee915f2839c"
      }
    ],
    "srn": null,
    "filename": "ak.d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b.tar.gz",
    "href": "/api/system/v1/bundles/d4431d57-ba4f-4f37-fa1e-a09fcbf3e56b",
    "date": "Thu Mar 10 2016 19:38:58 GMT+0000 (UTC)",
    "type": "User initiated"
  }
}
```

List Support Bundles

This command lists all support bundles being processed or collected by the Oracle ZFS Storage Appliance system. After a support bundle is uploaded to Oracle Support, the support bundle is removed from the system.

Example Request:

```
GET /api/system/v1/bundles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

Example Result:

```
{
    "bundles": [{
        "status": "building",
        "step_progress": 6.25,
        "srn": "3-0123456789",
        "filename": "/upload/issue/3-0123456789/3-0123456789_ak.ba8ebd55-2349-c31c-
cde3-acf3fb0c3389.tar.gz",
        "href": "/api/system/v1/bundles/ba8ebd55-2349-c31c-cde3-acf3fb0c3389",
        "date": "Wed Apr 30 2014 19:31:06 GMT+0000 (UTC)",
        "type": "User initiated",
        "uuid": "ba8ebd55-2349-c31c-cde3-acf3fb0c3389"
    }],
}
```

Get Support Bundle

Gets properties from a single bundle.

Example Request:

```
GET /api/system/v1/bundles/9604155c-928b-cf97-c826-cda9fc17ac57 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 165

{
    "bundle": {
        "status": "building",
        "step_progress": 62.5,
        "srn": "3-0123456789",
        "filename": "/upload/issue/3-0123456789/3-0123456789_ak.ba8ebd55-2349-
c31c-cde3-acf3fb0c3389.tar.gz",
        "href": "/api/system/v1/bundles/ba8ebd55-2349-c31c-cde3-acf3fb0c3389",
        "date": "Wed Apr 30 2014 19:31:06 GMT+0000 (UTC)",
        "type": "User initiated",
        "uuid": "ba8ebd55-2349-c31c-cde3-acf3fb0c3389"
    }
}
```

Cancel Support Bundle

This command cancels automatic upload of a support bundle.

Example Request:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/cancel HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
```

Retry Support Bundle Upload

This command creates a new bundle upload job that attempts to upload a bundle to Oracle Support. The get bundle command can be used to monitor the status of the support bundle upload.

Example Request:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/retry HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
```

To retry a bundle upload using a different Service Request (SR) number, use the `send` command. If an SR number is not provided, the system will retry the upload using the original SR number.

> **Note:**
>
> An SR number is required when running `send` on a locally generated bundle, or else an error will be thrown.

Example Request:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/send HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215

{"srn": "3-0123456789"}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
```

Upload Support Bundle

A support bundle that is not automatically uploaded to Oracle Support can be uploaded manually.

> **Note:**
>
> A Service Request (SR) number is required when running `send` on a locally generated bundle, or else an error will be thrown.

Example Request:

```
PUT /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3/send HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215

{"srn": "3-0123456789"}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
```

Delete Support Bundle

This command removes a support bundle from Oracle ZFS Storage Appliance.

Example Request:

```
DELETE /api/system/v1/bundles/9aef7c38-073c-603f-f35c-be64e26e90e3 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-Api: 1.0
```

System Updates

These commands manage Oracle ZFS Storage Appliance system update images.

Table 15-3 Update Commands

Request	Append to Path /api/system/v{1|2}	Description
GET	/updates	List all system updates
GET	/updates/*update*	Get the specified system update properties
GET	/update/platform	Show the update status of platform firmware (refers to the service processor (SP) and system board firmware on the controller)
GET	/update/firmware	Show the update status of component firmware (refers to disk and SSD firmware, and also Oracle Storage Drive Enclosure IOM firmware)
PUT	/updates/*update*	Modify update settings
PUT	/updates/*update*/upgrade	Upgrade to the specified update image
PUT	/updates/*update*/check	Run upgrade health checks for the specified update image
PUT	/updates/*update*/rollback	Rollback to the specified update image
PUT	/updates-apply	Apply deferred incompatible updates
DELETE	/updates/*update*	Destroy the specified system update
POST	/updates	Load an update image onto Oracle ZFS Storage Appliance

Table 15-4 Oracle ZFS Storage Appliance System Update Properties

Property	Type	Description
`version`	String	Update media version
`release_date`	DateTime	Update release date

Table 15-4 (Cont.) Oracle ZFS Storage Appliance System Update Properties

Property	Type	Description
`install_date`	DateTime	Update latest installation date; if not installed, date of download to Oracle ZFS Storage Appliance
`status`	String	Update media status (immutable)
`update_deferred`	ChooseOne	Deferred setting: `onreboot` or `onrequest`

Deferred updates notice:

```
The following updates enable features that are incompatible with earlier
software versions. As these updates cannot be reverted once committed, and
peer system resources are updated across a cluster, verifying first that the
system upgrade is functioning properly before applying deferred updates is
advised.
```

List System Updates

Example request to get Oracle ZFS Storage Appliance system updates:

```
GET /api/system/v1/updates HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Length: 541
Content-Type: application/json

{
    "updates": [
    {
        "release_date": "Tue Aug 13 2013 17:47:32 GMT+0000 (UTC)",
        "install_date": "Wed Aug 14 2013 12:33:08 GMT+0000 (UTC)"
        "href": "/api/system/v1/updates/nas@2013.08.13,1-0",
        "status": "previous",
        "version": "2013.08.13,1-0"
    },
    {
        "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
        "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)"
        "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
        "status": "current",
        "version": "2013.08.24,1-0"
    },
    {
        "release_date": "Sun Aug 25 2013 12:56:57 GMT+0000 (UTC)",
        "install_date": "Mon Aug 26 2013 18:50:33 GMT+0000 (UTC)"
        "href": "/api/system/v1/updates/nas@2013.08.25,1-0",
        "status": "waiting",
        "version": "2013.08.25,1-0"
    }]
}
```

Get System Update

Gets properties for a single update image.

Example Request:

```
GET /api/system/v1/updates/nas@2013.08.25,1-0 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Length: 541
Content-Type: application/json

{
    "update": {
        "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
        "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)"
        "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
        "status": "current",
        "version": "2013.08.24,1-0",
        "update_deferred", "on_request"
    }
}
```

Get Platform Firmware Update Status

Gets the update status for pending platform firmware updates. Platform firmware is a collective term that refers to the service processor (SP) and system board firmware on the Oracle ZFS Storage Appliance controller.

Example Request:

```
GET /api/system/v1/update/platform HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Length: 541
Content-Type: application/json

{
    "platform": {
       "href": "/api/system/v1/update/platform",
       "power_down_needed": true,
       "update_needed": "true"
    }
}
```

Get Component Firmware Update Status

Gets the number of pending, failed, and in-progress component firmware updates. Component firmware a collective term that refers to disk and SSD firmware, and also Oracle Storage Drive Enclosure disk shelf IOM firmware.

Example Request:

```
GET /api/system/v1/update/firmware HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Length: 541
Content-Type: application/json

{
    "firmware": {
        "href": "/api/system/v1/update/firmware",
        "upgrades_pending": 0,
        "upgrades_failed": 0,
        "upgrades_in_progress": 0
    }
}
```

Upload System Update

This command uploads a new Oracle ZFS Storage Appliance system update image.

Example Upload Command Using `curl`:

```
$ curl --user root:root-password -k --data-binary @nas@2013.08.24,1-0.pkg.gz \
         --header "Content-Type: application/octet-stream" \
         https://zfs-storage.example.com/api/system/v1/updates
```

After the image is uploaded and is unpacked, the properties of the update image are returned. The HTTP status is set to 201 (Created) on success, and the relative location of the new image is returned in the location header.

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Content-Length: 541
Content-Type: application/json
Location: /api/system/v1/updates/nas@2013.08.24,1-0

{
    "update": {
        "release_date": "Sat Aug 24 2013 17:54:23 GMT+0000 (UTC)",
        "install_date": "Sun Aug 25 2013 11:30:14 GMT+0000 (UTC)"
        "href": "/api/system/v1/updates/nas@2013.08.24,1-0",
        "status": "current",
        "version": "2013.08.24,1-0",
        "update_deferred", "on_request"
    }
}
```

Upgrade

This command loads the update image and reboots Oracle ZFS Storage Appliance to the specified update image. The specified image status should be equal to "waiting" or the command fails.

Example Request:

```
PUT /api/system/v1/updates/nas@2013.08.25,1-0/upgrade?force=true
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Length: 0
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
```

Rollback

Rollback reboots Oracle ZFS Storage Appliance to a previous update image.

Example Request:

```
PUT /api/system/v1/updates/nas@2013.08.24,1-0/rollback
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
Content-Length: 0
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
```

Delete Update Image

Removes an unused update image from Oracle ZFS Storage Appliance.

Example Request:

```
DELETE /api/system/v1/updates/nas@2013.08.13,1-0 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-Api: 1.0
```

16 RESTful API User Service

The RESTful API User service configures users on Oracle ZFS Storage Appliance.

User Service Commands

The following user service commands are available.

Table 16-1 User Service Commands

Request	Append to Path /api/user/v{1|2}	Description
GET	Use only /api/user/v{1|2}	List the user service commands
GET	/users	List all Oracle ZFS Storage Appliance users
GET	/users/*user*	List the properties of the specified user
POST	/users	Create a new Oracle ZFS Storage Appliance user
PUT	/users/*user*	Modify the properties of the specified user
DELETE	/users/*user*	Remove the specified user from Oracle ZFS Storage Appliance
GET	/users/*user*/preferences	List preference properties for the specified user
PUT	/users/*user*/preferences	Modify preference properties for the specified user
GET	/users/*user*/exceptions	List all authorization exceptions for the specified user
GET	/users/*user*/exceptions/*auth*	List the *auth* authorizations for the specified user
POST	/users/*user*/exceptions	Create new authorization exceptions for the specified user
PUT	/users/*user*/exceptions/*auth*	Modify the specified authorizations for the specified user
DELETE	/users/*user*/exceptions/*auth*	Destroy the specified authorizations for the specified user
GET	/users/*user*/preferences/keys	List all SSH keys for the specified user
GET	/users/*user*/preferences/keys/*key*	List the properties of the specified SSH key for the specified user
POST	/users/*user*/preferences/keys	Create a new SSH key for the specified user
PUT	/users/*user*/preferences/keys/*key*	Modify the specified SSH key for the specified user

Table 16-1 (Cont.) User Service Commands

Request	Append to Path /api/user/v{1|2}	Description
DELETE	/users/*user*/preferences/keys/*key*	Destroy the specified SSH key for the specified user
GET	/users/*user*/preferences/tokens	List all REST login tokens for the specified user
GET	/users/*user*/preferences/tokens?token=*token*	List a REST login token by its token value
GET	/users/*user*/preferences/tokens/*token-id*	List a REST login token by its token ID
POST	/users/*user*/preferences/tokens	Create a REST login token for the specified user
DELETE	/users/*user*/preferences/tokens?token=*token*	Remove a REST login token by its token value
DELETE	/users/*user*/preferences/tokens/*token-id*	Remove a REST login token by its token ID

User Service Properties

In addition to username and password, user service properties define characteristics such as which authorizations are granted to the user, what restrictions are placed on the user, and what is the user's locale.

User Properties

Users can have the following properties. Some properties are available only for users of a particular type.

Table 16-2 User Properties

Property	Type	Description
`logname`	string	Unique username. The `logname` is immutable after the user is created.
`type`	string	Type of user: `local`, `directory`, `data`, `nologin`. The `type` is immutable after the user is created. Default: `local`.
`uid`	number	User ID. You can specify the user ID or allow the system to assign the user ID. If you specify the user ID, the user ID cannot be less than 100, cannot be greater than 2147483646, and cannot be equal to 60001, 60002, or 65534. The `uid` is immutable after the user is created.
`fullname`	string	Full name or real name for the user. In the BUI, the full name is shown to the left of the **Logout** button at the top of the dashboard, and might also be shown on the browser tab. Default: same as `logname`.
`initial_password`	string	Password for local and data users.

Table 16-2 (Cont.) User Properties

Property	Type	Description
`require_annotation`	boolean	When true: • BUI – Require the user to enter a comment prior to displaying the initial BUI page. • CLI – Require the user to enter a comment prior to displaying the CLI prompt. • REST – Requests fail as unauthorized. The session annotation appears in the audit log.
`roles`	list	The roles assigned to a directory or local user.
`kiosk_mode`	boolean	When true, this user is a kiosk user: • BUI – The user is restricted to viewing only the screen that is the value of the `kiosk_screen` property. • CLI – Login fails. • REST – Requests fail as unauthorized.
`kiosk_screen`	string	The BUI screen that this user is restricted to if `kiosk_mode` is true. Default: `status/dashboard`.
`exceptions`	list	Additional authorizations assigned to a directory or local user, or limitations on authorizations that are assigned in a role.
`preferences`	list	User environment preferences such as locale, BUI start page, timeouts, SSH public keys, and REST login tokens.

For further descriptions of these properties, see the following documentation:

- Understanding Users and Roles in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*
- Managing User Properties in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*

User Roles and Exceptions

Use the `roles` property to assign existing roles to a user. To create a role, see RESTful API Role Service.

Use the `exceptions` property to add authorizations for a user. You can also use this mechanism to remove authorizations. For example, if the user is assigned a role that grants certain authorizations, you can set individual authorizations to `false` in the exceptions list, while keeping other authorizations that are included in the role.

The following request adds the authorization to restart the Kerberos service for user `user1`.

```
POST /api/user/v1/users/user1/exceptions HTTP/1.1

{
    "scope": "svc",
    "service": "kerberos",
    "allow_restart": true
}
```

The following result shows all authorizations that are available for the Kerberos service for `user1`. In this example, other authorizations still have their default value of `false`.

```
{
    "auth": {
        "href": "/api/user/v1/users/user1/exceptions/auth-001",
        "scope": "svc",
        "service": "kerberos",
        "allow_administer": false,
        "allow_configure": false,
        "allow_restart": true
    }
}
```

User Preferences Properties

You can set preferences for the user that you are logged in as or for any user for which you have the `allow_changePreferences` authorization. To gain the `allow_changePreferences` authorization, see User Roles and Exceptions.

Table 16-3 User Preferences Properties

Property	Type	Description
`locale`	string	Locality; default: C
`login_screen`	string	The BUI page that is presented upon successful login if a page is not specified in the URL. Default: `status/dashboard`.
`cli_idle_timeout`	integer	The length of time in seconds that the CLI can be idle before the session is killed. The default value, `-1`, means the CLI will not automatically close when idle.
`advanced_analytics`	string	Make available advanced analytics statistics
`keys`	list	RSA/DSA public keys
`tokens`	list	REST login tokens

See the following sections for more information about these properties:

- CLI Timeout
- SSH Keys
- REST Login Tokens

CLI Timeout

By default, there is no limit on the length of time that the command-line interface (CLI) can be idle (the value of `cli_idle_timeout` is -1). To set a limit on the length of time that the CLI can be idle, set a positive integer value for `cli_idle_timeout` seconds. If the timeout limit is reached, the CLI is closed.

The following example sets the CLI timeout to 1 hour.

```
PUT /api/user/v1/users/user1/preferences HTTP/1.1

{ "cli_idle_timeout": 3600 }

{
    "preferences": {
        "href": "/api/user/v1/users/user1/preferences",
        "locale": "C",
        "login_screen": "configuration/preferences",
        "session_timeout": 15,
        "cli_idle_timeout": 3600,
        "advanced_analytics": false,
        "keys": [],
        "tokens": []
    }
}
```

To disable the timeout, set the value of `cli_idle_timeout` to `-1` or use `unset`, as shown in the following example.

```
PUT /api/user/v1/users/user1/preferences HTTP/1.1

{ "<unset>": ["cli_idle_timeout"] }
```

Enclosing `unset` in angle brackets avoids the problem of having a property named `unset`.

SSH Keys

SSH public keys can be used to allow SSH connections without the use of passwords.

Table 16-4 SSH Key Properties

Property	Type	Description
`type`	string	The type of SSH key: either RSA or DSA
`key`	string	The contents of the SSH key
`comment`	string	A comment associated with this SSH key

REST Login Tokens

You can create a persistent or non-persistent REST login token, and access the token via its returned token value or token ID.

Table 16-5 REST Login Token Properties

Property	Type	Description
`name`	string	The token name
`token_username`	string	The name of the user that can use this login token; this value is set in the request path

Table 16-5 (Cont.) REST Login Token Properties

Property	Type	Description
`preserve`	boolean	If `false`, use the default `expiration` value. If `true`, set a custom `expiration` value. Default: `false`.
`expiration`	number	When creating a token, this value is seconds until the token expires. This property is required to be set if `preserve` is set to `true`. If `preserve` is set to `false`, the value of `expiration` is 900. When listing token properties, this value is the date and time that this token will expire.

REST Login Token Query Parameter

Use the `token=`*`token`* query parameter with the `GET` command to show the property values for that token. Use the `token=`*`token`* query parameter with the `DELETE` command to remove that token. The *token* is the value of `X-Auth-Session` in the create token response.

The *token* is displayed when the token is created and is not shown again. Be sure to copy and keep the *token*. See example "Creating a REST Login Token" in section Manage Tokens.

List Users

In the following example, the `root` user is local to this Oracle ZFS Storage Appliance system, and the `user1` user is an LDAP, NIS, or AD user. For a `directory` type user, the user ID, full name, and password are pulled from the directory service, and the password property is not listed. See more information about user types in Understanding Users and Roles in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
GET /api/user/v1/users HTTP/1.1
```

Example Result:

```
{
    "users": [
    {
        "logname": "root",
        "type": "local",
        "uid": 0,
        "fullname": "Super-User",
        "initial_password": true,
        "require_annotation": false,
        "href": "/api/user/v2/users/root"
    },{
        "logname": "user1",
        "type": "directory",
        "uid": user1uid,
        "fullname": "Real Name",
        "require_annotation": false,
        "roles": ["basic"],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard",
        "href": "/api/user/v2/users/user1"
    }]
}
```

List a Specific User

Information about a specific user includes user preferences and authorization exceptions. In this example, `user1` has an authorization exception named `auth-000` that grants `user1` the ability to configure and post alerts. Preferences for `user1` are all default values.

Example Request:

```
GET /api/user/v1/users/user1 HTTP/1.1
```

Example Result:

```
{
    "user": {
        "href": "/api/user/v2/users/user1",
        "logname": "user1",
        "type": "directory",
        "uid": user1uid,
        "fullname": "Real Name",
        "require_annotation": false,
        "roles": ["basic"],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard",
        "exceptions": [
            "scope": "alert",
            "allow_configure": true,
            "allow_post": true,
            "href": "/api/user/v2/users/user1/exceptions/auth-000"
        ],
        "preferences": {
            "href": "/api/user/v2/users/user1/preferences",
            "locale": "C",
            "login_screen": "status/dashboard",
            "session_timeout": 15,
            "cli_idle_timeout": -1,
            "advanced_analytics": false,
            "keys": [],
            "tokens": []
        }
    }
}
```

Create a User

To create a new user, you must provide at least a username (`logname`). If you do not specify a type, then the new user will be type `local`. Other properties are required depending on the type of the user, as described in the following list of user types:

- `directory` – The username must be an existing NIS, LDAP, or AD user. The UID, password, and full name are managed automatically by the directory service.
- `local` and `data` – Specify username and password. You can specify the UID, or a UID will be automatically assigned.

- `nologin` – Specify a username. You can specify the UID, or a UID will be automatically assigned.

To understand more about users and user types, see Understanding Users and Roles in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example 16-1 Creating a Local User

Example Request:

```
POST /api/user/v1/users HTTP/1.1

{
    "logname": "test_user",
    "initial_password": "password"
}
```

Example Result:

```
{
    "user":
    {
        "href": "/api/user/v1/users/test_user",
        "logname": "test_user",
        "type": "local",
        "uid": 2000000002,
        "fullname": "test_user",
        "initial_password": true,
        "require_annotation": false,
        "roles": ["basic"],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard",
        "exceptions": [],
        "preferences": {
            "href": "/api/user/v1/users/test_user/preferences",
            "locale": "C",
            "login_screen": "status/dashboard",
            "session_timeout": 15,
            "cli_idle_timeout": -1,
            "advanced_analytics": false,
            "keys": [],
            "tokens": []
        }
    }
}
```

Example 16-2 Cloning a User

To create a new user of the same type as an existing user and with the same roles and authorizations assigned, specify the following properties:

- `user` – Username of the user that you are cloning
- `clonename` – Username of the new cloned user
- `password` – Initial password of the new cloned user

Example Request:

```
POST /api/user/v1/users HTTP/1.1

{
    "user": "test_user",
    "clonename": "clone_user",
    "password": "password"
}
```

Example Result:

```
{
    "user":
    {
        "href": "/api/user/v1/users/clone_user",
        "logname": "clone_user",
        "type": "local",
        "uid": 2000000003,
        "fullname": "clone_user",
        "initial_password": true,
        "require_annotation": false,
        "roles": ["basic"],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard",
        "exceptions": [],
        "preferences": {
            "href": "/api/user/v1/users/clone_user/preferences",
            "locale": "C",
            "login_screen": "status/dashboard",
            "session_timeout": 15,
            "cli_idle_timeout": -1,
            "advanced_analytics": false,
            "keys": [],
            "tokens": []
        }
    }
}
```

Modify User Properties

Modifies user properties directly. The `logname`, `type`, and `uid` are immutable.

To add, modify, or delete roles, exceptions, preferences, SSH keys, or login tokens, see the following sections:

- RESTful API Role Service
- User Service Properties

Example Request:

```
PUT /api/user/v1/users/test_user HTTP/1.1

{"fullname": "Test A. User", "require_annotation": true}
```

Example Result:

```
{
    "user": {
        "href": "/api/user/v1/users/test_user",
        "logname": "test_user",
        "type": "local",
        "uid": 2000000002,
        "fullname": "Test A. User",
        "initial_password": true,
        "require_annotation": true,
        "roles": ["basic"],
        "kiosk_mode": false,
        "kiosk_screen": "status/dashboard",
        "exceptions": [],
        "preferences": {
            "href": "/api/user/v2/users/test_user/preferences",
            "locale": "C",
            "login_screen": "status/dashboard",
            "session_timeout": 15,
            "cli_idle_timeout": -1,
            "advanced_analytics": false,
            "keys": [],
            "tokens": []
        }
    }
}
```

Delete a User

Deletes a user from the Oracle ZFS Storage Appliance system.

Example Request:

```
DELETE /api/user/v1/users/clone_user HTTP/1.1
```

Example Result:

```
HTTP/1.1 204 No Content
```

Manage Tokens

You can create persistent and non-persistent REST login tokens, view properties of tokens, and delete tokens. Properties of REST login tokens are read-only after the token is created. You can access a token via its returned token value or token ID.

Example 16-3 Creating a REST Login Token

Username and password are required to create a REST login token.

Set the token `name`. By default the value of `preserve` is `false` and the value of `expiration` is 900. If you set `preserve` to `true`, then you must set `expiration` to a number of seconds.

Be sure to save the value of `X-Auth-Session` from the result. For more information, see REST Login Tokens.

Example Request:

```
POST /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: test_user
X-Auth-Key: password-xxx
Content-Type: application/json

{"name": "Test Token"}
```

Example Result:

```
HTTP/1.1 201 Created
...
```

```
X-Auth-Session: JjZJsZrVQfbZULyAuvSJjTftnBHCcQT

{
    "token": {
        "href":
"/api/user/v2/users/test_user/preferences/tokens/fb65a127-a04c-4f58-bc52-
efa884447efb",
        "name": "Test Token",
        "token_username": "test_user",
        "preserve": false,
        "expiration": "2020-04-30T02:33:44Z",
        "id": "fb65a127-a04c-4f58-bc52-efa884447efb"
    }
}
```

Example Request:

```
POST /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
Host: zfs-storage.example.com:215
X-Auth-User: test_user
X-Auth-Key: password-xxx
Content-Type: application/json

{"name": "Another Token", "preserve": true, "expiration": 3600}
```

Example Result:

```
HTTP/1.1 201 Created
...
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK

{
    "token": {
        "href":
"/api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-
dd5560256dfb",
        "name": "Another Token",
        "token_username": "test_user",
        "preserve": true,
        "expiration": "2020-04-30T03:20:31Z",
        "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb"
    }
}
```

Example 16-4 List All Tokens of the Specified User

Example Request:

```
GET /api/user/v2/users/test_user/preferences/tokens HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

Example Result:

```
{
    "tokens": [{
        "name": "Another Token",
        "token_username": "test_user",
        "preserve": true,
        "expiration": "2020-04-30T03:20:31Z",
        "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb",
        "href": "/api/user/v2/users/testuser1/preferences/tokens/21f981ad-6221-4fb4-
a7d1-dd5560256dfb"
```

```
    },{
        "name": "Test Token",
        "token_username": "test_user",
        "preserve": false,
        "expiration": "2020-04-30T02:33:44Z",
        "id": "fb65a127-a04c-4f58-bc52-efa884447efb",
        "href": "/api/user/v2/users/testuser1/preferences/tokens/fb65a127-
a04c-4f58-bc52-efa884447efb"
    }]
}
```

Example 16-5 List a Specific Token by Its Token Value

Example Request:

```
GET /api/user/v2/users/test_user/preferences/tokens?
token=pviHrthBGQhGZHoPuqxFQrDcCPZgwEK HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

Example Result:

```
{
    "token": {
        "href":
"/api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-
dd5560256dfb",
        "name": "Another Token",
        "token_username": "test_user",
        "preserve": true,
        "expiration": "2020-04-30T03:20:31Z",
        "id": "21f981ad-6221-4fb4-a7d1-dd5560256dfb"
    }
}
```

Example 16-6 List a Specific Token by Its Token ID

```
GET /api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-
dd5560256dfb HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

The result is the same as earlier.

Example 16-7 Delete a Token by Its Token Value

Example Request:

```
DELETE /api/user/v2/users/test_user/preferences/tokens?
token=pviHrthBGQhGZHoPuqxFQrDcCPZgwEK HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

Example Result:

```
HTTP/1.1 204 No Content
```

Example 16-8 Delete a Token by Its Token ID

```
DELETE /api/user/v2/users/test_user/preferences/tokens/21f981ad-6221-4fb4-a7d1-
dd5560256dfb HTTP/1.1
X-Auth-Session: pviHrthBGQhGZHoPuqxFQrDcCPZgwEK
```

17

RESTful API Role Service

A role is a named collection of authorizations that can be assigned to users. You can create roles with different authorizations for different purposes. Oracle ZFS Storage Appliance users can be assigned any necessary roles. Using roles is more secure than sharing administrator passwords. Roles restrict users to necessary authorizations only, and attribute user actions to that username in the audit log.

The "Basic administration" role is preexisting and is assigned to all users by default.

Role Service Commands

The following table lists the role service commands.

Table 17-1 Role Service Commands

Request	Append to Path /role/v{1|2}	Description
GET	Use only /role/v{1|2}	List the role service commands
GET	/roles	List all roles
GET	/roles/*role*	List the properties of the specified role
POST	/roles	Create a new role
PUT	/roles/*role*	Modify the properties of the specified role
PUT	/roles/*role*/revoke	Remove the specified role from all users
DELETE	/roles/*role*	Destroy the specified role
GET	/roles/*role*/authorizations	List all authorizations for the specified role
GET	/roles/*role*/authorizations/*auth*	List the properties of the specified role authorization
POST	/roles/*role*/authorizations	Create a new authorization for the specified role
PUT	/roles/*role*/authorizations/*auth*	Modify the properties of the specified role authorization
DELETE	/roles/*role*/authorizations/*auth*	Destroy the specified role authorization

Role Service Properties

The `name` and `description` properties are required to create a role and are displayed when you list all roles. Authorizations are added after the role is created and are displayed when you list a specific role.

Table 17-2 Role Properties

Property	Type	Description
`name`	string	Name of the role as it will be shown in lists
`description`	string	Verbose description of the role
`authorizations`	list	Authorizations for this role

List Roles

List all defined roles.

Example Request:

```
GET /api/role/v1/roles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
{
    "roles": [{
        "description": "Basic administration",
        "href": "/api/role/v1/roles/basic",
        "name": "basic",
        "role": "basic"
    }, {
        "description": "a",
        "href": "/api/role/v1/roles/rola",
        "name": "rola",
        "role": "rola"
    }]
}
```

Get Role

Retrieves the properties for a single role. To return the property metadata, set the `props` query parameter to `true`.

Example Request:

```
GET /api/role/v1/roles/basic?props=true HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 390

{
    "props": [{
        "immutable": true,
        "label": "Role name",
        "name": "name",
        "type": "String"
    }, {
        "label": "A description of this role",
        "name": "description",
        "type": "String"
    }],
    "role": {
        "authorizations": [],
        "description": "Basic administration",
        "href": "/api/role/v1/roles/basic",
        "name": "basic"
    }
}
```

Create Role

This command creates a new role.

Table 17-3 Create New Role Properties

Property	Type	Description
`name`	string	Name of the new role (required)
`clone`	string	Name of the role to clone original properties (optional)
`description`	string	Role description (required)

Example Request:

```
POST /api/role/v1/roles HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 71

{"name":"role_workflow", "description":"Role to run workflows"}
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 143
Location: /api/role/v1/roles/role_workflow

{
    "role": {
        "authorizations": [],
        "description": "Role to run workflows",
        "href": "/api/role/v1/roles/role_workflow",
        "name": "role_workflow"
    }
}
```

Modify Role

The role properties can be modified after a role is created.

Example Request:

```
PUT /api/role/v1/roles/role_workflow HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 54

{"description":"Role allowing user to run workflows!"}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 158

 {
    "role": {
        "authorizations": [],
        "description": "Role allowing user to run workflows!",
        "href": "/api/role/v1/roles/role_workflow",
        "name": "role_workflow"
    }
}
```

Revoke Role

Revokes a role from all users.

Example Request:

```
PUT /api/role/v1/role_worksheets/revoke HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 0
```

Delete Role

Deletes a role from the Oracle ZFS Storage Appliance system. If the role is still assigned to one or more users, add `?confirm=true` to the `DELETE` command.

Example Request:

```
DELETE /api/role/v1/roles/rola?confirm=true HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-Api: 1.0
```

List Role Authorizations

Lists the authorizations for the selected role.

Example Request:

```
GET /api/role/v1/roles/role_workflow/authorizations HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
{
    "authorizations": [{
        "allow_modify": false,
        "allow_read": true,
        "auth": "auth-000",
        "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
        "owner": "*",
        "scope": "workflow",
        "uuid": "*"
    }]
}
```

Create Role Authorization

Creates a new role authorization. The input properties are the same as defined in the CLI. Each authorization has a defined `scope` property. Other properties can be set based on the input scope. Scope values include:

```
ad         cluster    keystore   role       stmf       user
alert      dataset    nas        schema     svc        workflow
appliance  hardware   net        stat       update     worksheet
```

Example Request:

```
POST /api/role/v1/roles/role_workflow/authorizations HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 41
{"scope": "workflow", "allow_read": true}
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 171
Location: /api/role/v1/roles/role_workflow/authorizations/auth-000
```

```
{
    "auth": {
        "allow_modify": false,
        "allow_read": true,
        "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
        "owner": "*",
        "scope": "workflow",
        "uuid": "*"
    }
}
```

Modify Role Authorization

The role authorization properties can be modified.

Example Request:

```
PUT /api/role/v1/roles/role_workflow/authorizations/auth-000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 29

{"allow_modify": true}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 171

{
    "auth": {
        "allow_modify": true,
        "allow_read": true,
        "href": "/api/role/v1/roles/role_workflow/authorizations/auth-000",
        "owner": "*",
        "scope": "workflow",
        "uuid": "*"
    }
}
```

Delete Role Authorization

Deletes a role authorization.

Example Request:

```
DELETE /api/role/v1/roles/role_workflow/authorizations/auth-000 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-Api: 1.0
```

18 Workflow and Script Commands

This service is used to manage workflows. A workflow is a script that is uploaded to and managed by Oracle ZFS Storage Appliance. Workflows can be parameterized and executed in a first-class fashion from either the browser interface or the command line interface. Workflows can also be executed as alert actions or at a designated time; thus, they can allow the appliance to be extended in ways that capture specific policies and procedures, and they can be used to formally encode best practices for a particular organization or application.

Workflow and Script Service Commands

The following table shows the workflow service commands.

Table 18-1 Workflow Service Commands

Request	Append to Path /api/workflow/v{1|2}	Description
GET	Use only /api/workflow/v{1|2}	List the workflow service commands
POST	/workflows	Upload a new workflow onto Oracle ZFS Storage Appliance
GET	/workflows	List all workflows
GET	/workflows/*workflow*	List the specified workflow properties
PUT	`/workflows/`*`workflow`*	Modify the specified workflow properties
PUT	/workflows/*workflow*/execute	Execute the specified workflow
DELETE	/workflows/*workflow*	Destroy the specified workflow
POST	/scripts	Upload and run a script
GET	/scripts	List all running scripts
GET	/scripts/*script*	Reconnect to a running script
DELETE	/scripts/*script*	Stop a running script

Upload Workflow

Uploads a workflow to Oracle ZFS Storage Appliance.

Example Request:

```
POST /api/workflow/v1/workflows HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/javascript
Content-Length: 290

var workflow = {
    name: 'Echo',
    description: 'Echo bird repeats a song.',
    parameters: {
        song: {
            label: 'Words of a song to sing',
            type: 'String',
        }
    },
    execute: function (params) { return (params.song) }
};
```

Example Result:

```
HTTP/1.1 201 Created
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 268
X-Zfssa-Version: user/generic@2013.09.14,1-0
Location: /api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971

{
    "workflow": {
        "href": "/api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-
cd0c0cce9971",
        "name": "Echo",
        "description": "Echo bird repeats a song.",
        "uuid": "f4fe892f-cf46-4d6a-9026-cd0c0cce9971",
        "owner": "root",
        "origin": "<local>",
        "setid": false,
        "alert": false,
        "version": "",
        "scheduled": false
    }
}
```

List Workflows

Lists all workflows installed on an Oracle ZFS Storage Appliance system. If the query parameter `showhidden=true` is set, the list includes workflows that are normally hidden.

Example Request:

```
GET /api/workflow/v1/workflows HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json; charset=utf-8
Content-Length: 1908

{
    "workflows": [{
        "description": "Clear locks held on behalf of an NFS client",
        "href": "/api/workflow/v1/workflows/10f25f2c-3a56-e733-d9c7-
d4c6fd84e073",
        ...
    },
    {
        "description": "Sets up environment for Oracle Solaris Cluster NFS",
        "href": "/api/workflow/v1/workflows/2793f2dc-72de-eac4-c58b-cfbe527df92d",
        ...
    },
    {
        "description": "Removes the artifacts from the appliance used by Oracle
Solaris Cluster NFS",
        "href": "/api/workflow/v1/workflows/9e2d5eed-cc72-67b0-e913-bf5ffad1d9e1",
        ...
    },
    {
        "description": "Sets up environment to be monitored by Oracle Enterprise
Manager",
        "href": "/api/workflow/v1/workflows/bb5de1b8-b950-6da6-a650-f6fb19f1172c",
        ...
    },
    {
        "description": "Removes the artifacts from the appliance used by Oracle
Enterprise Manager",
        "href": "/api/workflow/v1/workflows/bd7214fc-6bba-c7ad-ed1f-942c0189e757",
        ...
    }]
}
```

Get Workflow

Gets properties for a single workflow. In the header, if `Accept` is specified as `application/javascript`, it returns the content of the workflow, otherwise it returns workflow properties.

Example request where `Accept` is specified as `application/javascript`:

```
GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/javascript
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/javascript; charset=utf-8
Content-Length: 916

    var workflow = {
        name: 'Clear locks',
        description: 'Clear locks held on behalf of an NFS client',
        origin: 'Oracle Corporation',
        version: '1.0.0',
        parameters: {
            hostname: {
                label: 'Client hostname',
                type: 'String'
            },
            ipaddrs: {
                label: 'Client IP address',
                type: 'String'
            }
        },
        validate: function (params) {
            if (params.hostname == '') {
                return ({ hostname: 'Hostname cannot be empty.' });
            }

            if (params.ipaddrs == '') {
                return ({ ipaddrs: 'IP address cannot be empty.' });
            }
        },
        execute: function (params) {
            try {
                nas.clearLocks(params.hostname, params.ipaddrs);
            } catch (err) {
                return ('Failed to clear NFS locks: ' + err.message);
            }

            return ('Clear of locks held for ' + params.hostname +
                ' returned success.' );
        }
    };
```

Example request where `Accept` is specified as `application/json`:

```
GET /api/workflow/v1/workflows/cc574599-4763-4523-9e72-b74e1246d448 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
```

Example Result:

```
HTTP/1.1 200 OK
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json; charset=utf-8
Content-Length: 649

{
    "workflow": {
        "href": "/api/workflow/v1/workflows/cc574599-4763-4523-9e72-
b74e1246d448",
        "name": "Clear locks",
        "description": "Clear locks held on behalf of an NFS client",
        "uuid": "cc574599-4763-4523-9e72-b74e1246d448",
        "checksum":
"695d029224f614258e626fe0b3c449c1233dee119571f23b678f245f7748d13c",
        "installdate": "Wed Apr 01 2015 17:59:44 GMT+0000 (UTC)",
        "owner": "root",
        "origin": "Oracle Corporation",
        "setid": false,
        "alert": false,
        "version": "1.0.0",
        "scheduled": false
    }
}
```

Modify a Workflow

You can modify properties for a single workflow by sending a PUT request to a workflow resource.

Example Request:

```
PUT /api/workflow/v1/workflows/6c2b6545-fa78-cc7b-8cc1-ff88bd628e7d HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28

{"setid": false}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 234

{
    "workflow": {
        "alert": false,
        "description": "Echo bird repeats a song.",
        "href": "/api/workflow/v1/workflows/448b78e1-f219-e8f4-abb5-e01e09e1fac8",
        "name": "Echo",
        "origin": "<local>",
        "owner": "root",
        "scheduled": false,
        "setid": true,
        "uuid": "448b78e1-f219-e8f4-abb5-e01e09e1fac8",
        "version": ""
    }
}
```

Execute a Workflow

Executes a workflow script and return the results. Any workflow parameters must be passed in a JSON object within the body. On success HTTP status 202 (Accepted) is returned along with a JSON object with a single result property containing the workflow output.

Example Request:

```
PUT /api/workflow/v1/workflows/6c2b6545-fa78-cc7b-8cc1-ff88bd628e7d/run HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: application/json
Content-Type: application/json
Content-Length: 28

{"song": "tweet tweet tweet"}
```

Example Result:

```
HTTP/1.1 202 Accepted
X-Zfssa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 34

{
    "result": "tweet tweet tweet\n"
}
```

Delete Workflow

Deletes a workflow script from Oracle ZFS Storage Appliance.

Example Request:

```
DELETE /api/workflow/v1/workflows/f4fe892f-cf46-4d6a-9026-cd0c0cce9971 HTTP/1.1
Authorization: Basic Tm8gcGVla2luZyE=
Host: zfs-storage.example.com:215
Accept: */*
```

Example Result:

```
HTTP/1.1 204 No Content
X-Zfssa-Appliance-Api: 1.0
```

Upload and Run a Script

Uploads and runs a script on Oracle ZFS Storage Appliance.

A root user can view and access all scripts uploaded to the appliance. A non-root user can only view and access their own scripts.

For more information on scripting, see Working with CLI Scripting in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

This script lists all shares on the appliance.

Example Request:

```
$ curl -kv --user root:pw --data-binary @listShares.aksh \
         https://hostname:215/api/workflow/v1/scripts

POST /api/workflow/v1/scripts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
Content-Length: 12
Content-Type: application/x-www-form-urlencoded
```

Example Result:

```
HTTP/1.1 201 Created
Date: Mon, 27 Mar 2017 22:16:38 GMT
X-Zfssa-Workflow-Api: 1.1
X-Zfssa-Version: user/generic@2017.02.27,1-0
X-Zfssa-Api-Version: 1.0
Content-Type: plain/text; charset=utf-8
Transfer-Encoding: chunked

default
share1
share2
fs1
lun1
```

List All Running Scripts

Use the following command to list all running scripts.

A root user can view and access all scripts uploaded to Oracle ZFS Storage Appliance. A non-root user can only view and access their own scripts.

For more information on scripting, see Working with CLI Scripting in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
$ curl -kv --user root:pw https://hostname:215/api/workflow/v1/scripts

GET /api/workflow/v1/scripts HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
```

Example Result:

```
HTTP/1.1 200 OK
Date: Mon, 27 Mar 2017 22:41:06 GMT
Content-Length: 96
X-Zfssa-Workflow-Api: 1.1
X-Zfssa-Api-Version: 1.0
Content-Type: application/json; charset=utf-8

{
    "scripts": [
        {
            "time": 4,
            "href": "/api/workflow/v1/scripts/1",
            "user": "root",
            "script": "1"
        },
        {
              "time": 39,
              "href": "/api/workflow/v1/scripts/9",
              "user": "root",
              "script": "9"
        }
    ]
}
```

Reconnect to a Running Script

A root user can reconnect to any running script uploaded to Oracle ZFS Storage Appliance. A non-root user can only reconnect to their own running scripts.

For more information on scripting, see Working with CLI Scripting in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
$ curl -kv -H "Accept: text/plain" --user root:pw \
         https://hostname:215/api/workflow/v1/scripts/9
```

```
GET /api/workflow/v1/scripts/9 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: text/plain
```

Example Result:

```
{
    "test2": {,
        "str": "this is a string",
        "bool": "True",
        "posint": 994,
        "int": 1123,
        "address": "",
        "host": "192.0.2.0",
        "hostname": "example.com",
        "color": "red",
        "languages": "latin",
        "size": "red",
        "onoff": "off",
        "number": 0,
        "stringlist": "this is another string",
        "emptystringlist": "this is another string",
        "yetanotherstr": "You can't change me",
        "emptystr": "Any string",
        "password": "password",
        "longpassword": "longpassword",
        "permissions": "022",
        "nonnegativeint": 42,
        "port": 21,
        "time": "Thu Jan 01 1970 15:22:30 GMT+0000 (UTC)",
        "date": "Sun Jun 17 2007 00:00:00 GMT+0000 (UTC)",
        "datetime": "Sun Jun 17 2007 15:22:00 GMT+0000 (UTC)",
        "hostport": "ipaddr-1",
        "dn": "uid=root,ou=people,dc=fishpong,dc=com",
        "commalist": "foo,bar"
    }
},
    "utask": []
}
```

Stop a Running Script

A root user can delete any running script uploaded to Oracle ZFS Storage Appliance. A non-root user can only access and delete their own running scripts.

For more information on scripting, see Working with CLI Scripting in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

Example Request:

```
$ curl -kv -X DELETE --user root:l1a \
         https://hostname:215/api/workflow/v1/scripts/9

DELETE /api/workflow/v1/scripts/9 HTTP/1.1
Host: zfs-storage.example.com:215
Authorization: Basic Tm8gcGVla2luZyE=
User-Agent: curl/7.45.0
Accept: */*
```

Example Result:

```
HTTP/1.1 204 No Content
Date: Mon, 27 Mar 2017 22:59:12 GMT
Content-Length: 0
X-Zfssa-Workflow-Api: 1.1
X-Zfssa-Version: build/generic@2017.02.27,1-0
X-Zfssa-Api-Version: 1.0
Content-Type: application/json; charset=utf-8
```

19 RESTful Clients

Any HTTP client can be used as a RESTful client. Even the BUI can return RESTful API `GET` results by typing in a resource URL. Mozilla Firefox has a RESTful client module that can be installed to make RESTful requests (`https://addons.mozilla.org/en-us/firefox/addon/restclient/`). This module allows `PUT`, `POST`, and `DELETE` requests, as well as the normal HTTP `GET` requests.

RESTful clients must use TLS protocols because the SSL v2/3 protocols are no longer supported. Curl clients must use `curl` version 7.34.0 or higher.

This section contains more detailed information about various RESTful clients.

Curl Rest Client

Curl clients must use `curl` version 7.34.0 or higher. Two common CLI-based HTTP clients are `wget` and `curl`. This section shows several examples of using `curl` to make RESTful API calls, and similar functionality can be accomplished using `wget`.

Get Resource Data

This example shows how to use a simple HTTP `GET` request to obtain some JSON data:

```
$ curl --user ${USER}:${PASSWORD} -k \
  -i https://hostname:215/api/storage/v1/pools/p1

 HTTP/1.1 200 OK
 Date: Tue, 23 Jul 2018 12:57:02 GMT
 Server: WSGIServer/0.1 Python/2.6.4
 Content-Length: 284
 Content-Type: application/json
 X-Zfs-Sa-Nas-Api: 1.0

{
   "pool": {
      "profile": "mirror",
      "name": "p1",
      "usage": {
         "available": 895468984832.0,
         "total": 895500681216.0,
         "dedupratio": 100,
         "used": 31696384.0
      },
      "peer": "00000000-0000-0000-0000-000000000000",
      "state": "online",
      "owner": "admin1",
      "asn": "314d252e-c42b-e844-dab1-a3bca680b563"
    }
}
```

Create a New Resource

This example shows how to send JSON data in a request to create a new resource:

```
$ curl --user ${USER}:${PASSWORD} -s -k -i -X POST -d @-  \
         -H "Content-Type: application/json" \
         https://zfs-storage.example.com:215/api/user/v1/users <<JSON
> {"logname": "rest_user",
> "fullname": "REST User",
> "initial_password": "password"}
> JSON

HTTP/1.1 201 Created
Date: Tue, 23 Jul 2018 13:07:37 GMT
Server: WSGIServer/0.1 Python/2.6.4
X-Zfs-Sa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 357

{
   "user": {
      "logname": "rest_user",
      "fullname": "REST User",
      "initial_password": "password",
      "require_annotation": false,
      "kiosk_mode": false,
      "kiosk_screen": "status/dashboard",
      "roles": ["basic"],
      "exceptions": {},
      "preferences": {
         "href": "/api/user/v1/users/admin1/preferences",
         "locale": "C",
         "login_screen": "status/dashboard",
         "session_timeout": 15,
         "cli_idle_timeout": -1,
         "advanced_analytics": false,
         "keys": {}
      }
   }
}
```

Modify an Existing Resource

This example modifies a user's session timeout:

```
$ curl --user admin1:password -3 -s -k -i -X PUT \
         -H "Content-Type: application/json" -d @-  \
         https://zfs-storage.example.com:215/api/appliance/v1/users/admin1/
preferences <<JSON
> {"session_timeout":60}
> JSON

HTTP/1.1 202 Accepted
Date: Wed, 24 Jul 2018 05:43:17 GMT
X-Zfs-Sa-Appliance-Api: 1.0
Content-Type: application/json
Content-Length: 0

{
    "preferences": {
        "href": "appliance/v1/users/admin1/preferences",
        "locale": "C",
        "login_screen": "status/dashboard",
        "session_timeout": 60,
        "cli_idle_timeout": -1,
        "advanced_analytics": false,
        "keys": {}
    }
}
```

Delete an Existing Resource

This command removes a user from the Oracle ZFS Storage Appliance system:

```
$ curl --user ${USER}:${PASSWORD} -s -k -i -X DELETE \
       https://zfs-storage.example.com:215/api/appliance/v1/users/admin1
HTTP/1.1 204 No Content
Date: Tue, 23 Jul 2018 13:21:11 GMT
Server: WSGIServer/0.1 Python/2.6.4
X-Zfs-Sa-Appliance-Api: 1.0
Content-Length: 0
```

Python RESTful Client

A Python RESTful API client is provided along with a REST test library to aid in test development of RESTful services.

Example RESTful Client Program:

```
>>> import urllib2
>>> import json

>>> request = urllib2.Request("https://zfsssa.example:215/api/access/v1", "")
>>> request.add_header("X-Auth-User", "rest_user")
>>> request.add_header("X-Auth-Key", "password")
>>> response = urllib2.urlopen(request)
>>> response.getcode()
201

>>> info = response.info()
>>>
        opener = urllib2.build_opener(urllib2.HTTPHandler)
>>> opener.addheaders = [("X-Auth-Session", info.getheader("X-Auth-Session")),
... ('Content-Type', 'application/json'), ('Accept', 'application/json')]
```

The opener can then be used to open requests that are already pre-authenticated and ready to send/receive JSON data.

Get a Resource

The following Python code can be used to get data from any RESTful API resource.

Example GET:

```
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/network/v1/
routes")
>>> response = opener.open(request)
>>> response.getcode()
```

```
200
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{
    "routes": [

            {
            "destination": "ipaddr-0",
            "family": "IPv4",

            "gateway": "ipaddr-1",
            "href":
            "/api/network/v1/routes/ixgbe0,ipaddr-0,ipaddr-1",

            "interface": "ixgbe0",
            "mask": 0,
            "type": "static"

            }
    ]
}
```

Create a Resource

Example Python code to create a new resource:

```
>>> action = {'category': 'network'}
>>> post_data = json.dumps(action)
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/
actions", post_data)
>>> request.add_header('Content-Type', 'application/json')

>>> response = opener.open(request)
>>> response.getcode()
201
>>> response.info().getheader('Location')
'/api/alert/v1/actions/actions-001'
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{

        "actions": {
        "category": "network",
        "datalink_failed": true,

        "datalink_ok": true,
        "href":
        "/api/alert/v1/actions/actions-001",

        "ip_address_conflict": true,

        "ip_address_conflict_resolved": true,

        "ip_interface_degraded": true,
        "ip_interface_failed":
        true,
        "ip_interface_ok": true,

        "network_port_down": true,
        "network_port_up":
```

```
        true
    }
}
```

Modify a Resource

Example Python code to modify an existing resource:

```
>>> put_data = '{"ip_address_conflict_resolved": false}'
>>>
        request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/
actions/actions-001", put_data)
>>> request.add_header('Content-Type', 'application/json')
>>> request.get_method = lambda: 'PUT'

>>> response = opener.open(request)
>>> response.getcode()
202
>>> body = json.loads(response.read())
>>> print json.dumps(body, sort_keys=True, indent=4)
{

        "actions": {
        "category": "network",
        "datalink_failed": true,

        "datalink_ok": true,
        "href":
        "/api/alert/v1/actions/actions-001",

        "ip_address_conflict": true,

        "ip_address_conflict_resolved": false,

        "ip_interface_degraded": true,
        "ip_interface_failed":
        true,
        "ip_interface_ok": true,

        "network_port_down": true,
        "network_port_up":
        true
    }
}
```

Delete an Existing Resource

Example Python code to delete an existing resource:

```
>>> request = urllib2.Request("https://zfs-storage.example.com:215/api/alert/v1/
actions/actions-001")
>>> request.get_method = lambda: 'DELETE'
>>> response = opener.open(request)
>>> response.getcode()
204
```