ORACLE

Configuring ZFS Storage as an Object Storage Target for Cloud Snapshot Backups

Configuration Best Practices

May, 2024 | Version [5.0]

DISCLAIMER

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

TABLE OF CONTENTS

INTRODUCTION

Object storage is the primary method of storing large amounts of data in the cloud. With a flat namespace, it scales better than file system storage and supports extended metadata attributes that allow comprehensive search operations. Billions of objects can be stored when needed.

Oracle ZFS Storage, both the on-premises appliance and the virtualized ZFS-HA appliance in OCI, provides the abilities to both back up ZFS snapshots to Oracle Cloud Infrastructure (OCI) object storage and to provide OCI-compatible object storage itself that can be a target for other ZFS Storage cloud backups.

This paper details the process of configuring ZFS Storage as an OCI-compatible object store target for cloud backups. Configuring cloud backups is the topic of the paper "Configuring ZFS Storage for Cloud Snapshot Backups to OCI Object Storage". The paper "Oracle ZFS Storage Appliance: On-premises Object Storage Best Practices and Recommended Use," available at the ZFS Storage NAS Documentation web page, also gives more information on the general use and best practices using ZFS Storage for OCI-compatible object storage.

Cloud Snapshot Backups

A snapshot in Oracle ZFS Storage provides an immutable "point-in-time" view of a ZFS project or share. These snapshots can be mounted for read-only access or cloned. Clones may be mounted with read/write access without affecting the original snapshot.

The cloud snapshot backup feature of Oracle ZFS Storage allows you to back up full and incremental snapshots from Oracle ZFS Storage to OCI object storage targets both in OCI and on ZFS Storage.

Use Cases

- Provides low-cost off-site storage for snapshot backups
    - Snapshot backups can be scheduled through workflows
    - Provides recovery by restoring snapshots and rolling back
    - Provides recovery from accidental overwrite or malware like ransomware
    - Supports less-expensive Archive storage for backup objects
    - Provides off-site archive storage for long-term data, such as business compliance requirements. OCI objects can optionally have retention rules set, ensuring that the objects cannot be removed accidentally or by a bad actor
- Provides a straightforward way to migrate data from an on-premises Oracle ZFS Storage Appliance to OCI storage

Advantages

Review the following advantages for storing or migrating data in a hybrid cloud environment with the ZFS Storage Appliance:

- Provides native cloud integration with OCI object storage
- Cloud snapshot backups are fully supported in the ZFS Appliance BUI, CLI, and REST interfaces and integrated with analytics, alerts, logs, and authorization roles
- File system snapshots can be backed up to another on-premises ZFS Appliance or directly to OCI object storage
- Cloud snapshot backups using the ZFS format can be restored back to either an on-premises ZFS Storage Appliance or a ZFS-HA cluster running in OCI
- Cloud snapshot backups using the TAR format can be restored back to ZFS systems or to any system able to process .tar files
- Snapshot backups can be encrypted and compressed and if required, replicated to worldwide on-premises data centers or up to OCI cloud storage

OVERVIEW

This document details the features and benefits of using OCI object storage to back up Oracle ZFS Storage snapshots.

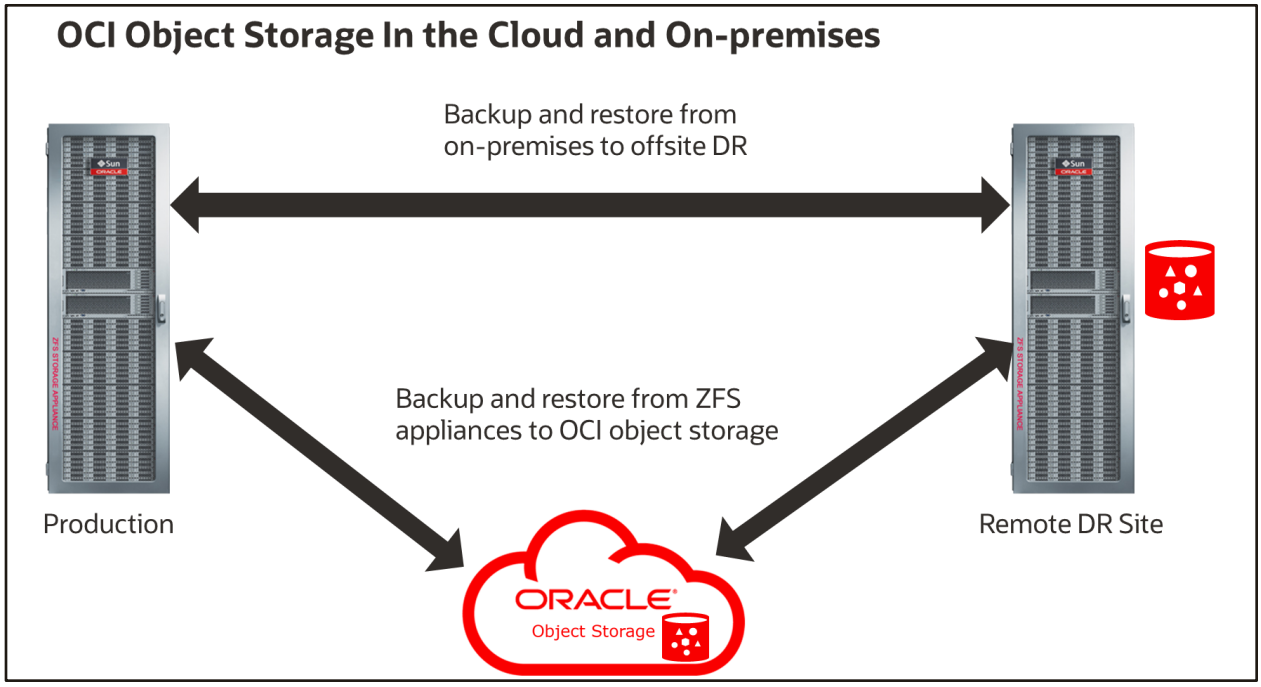
It also details the following processes:

- Configuring a ZFS Storage node as a cloud snapshot backup source system
- Configuring an OCI cloud target for archiving cloud snapshot backups
- Sending a ZFS cloud snapshot backup to OCI object storage
- Restoring a ZFS cloud snapshot from OCI object storage

The source ZFS Storage node is defined as the host for data in the form of filesystems and LUNs from which a snapshot can be generated and subsequently pushed to the target. These snapshots can be in either ZFS or tar format.

The target ZFS Storage node is defined as the recipient of snapshots sent from the source ZFS Storage node. The target node is seen as an OCI bucket by the source node.

The following is a context diagram of the environment.



The ZFS Appliance on the left takes on the role of the source from which shares and LUNs can be exposed to clients. The source can initiate a snapshot and then send that snapshot to the target ZFS Appliance shown on the right, which is another node over a defined network, or it may send the snapshot directly to object storage in OCI.

FEATURES OF CLOUD SNAPSHOT BACKUPS AND OCI OBJECT STORAGE

Cloud Snapshot Backup Formats (ZFS or Tar)

ZFS Storage Appliance cloud snapshot backups support both ZFS format and tar format. The ZFS format can be used when restoring cloud snapshot backups back to ZFS Storage, either on-premises or in OCI. The tar format provides more data recovery flexibility because you can restore back to any system that can process tar files, usually Solaris or Linux systems. Both ZFS and tar format can be backed up to the same object storage bucket. ZFS will automatically select the proper format upon restoration back to ZFS Storage.

The ZFS format is block based. Incremental backups will only transfer the blocks that have changed since the last snapshot. Compression applied at the share level will be reflected in the blocks sent to OCI object storage.

The tar format is file based. A change of a single byte in a large file will cause the entire file to be sent in an incremental snapshot. Tar-formatted cloud snapshots do not reflect any compression that may be applied to the ZFS share but will be compressed within the snapshot.

The following table summarizes the features of the ZFS and tar formats.

ZFS or Tar Format Feature Comparison

Feature Description	ZFS Format	Tar Format
Restore on any system?	ZFS Storage Appliance, ZFS-HA cluster, or Oracle Solaris server	Any system regardless of OS
Supports both filesystem and LUN snapshots?	Yes	Filesystem snapshots
Preserves filesystem or LUN properties?	Yes	Filesystem properties
Supports full and incremental backups	Yes	Yes
Supported within the same Oracle Cloud Infrastructure bucket?	Yes	Yes
Supports high-efficiency compression? • If underlying share is compressed, less data is transferred and the backup is faster. • Incremental backups are performed at the block level.	Yes	No
Supports mid-efficiency compression? • Files are read and compressed during the backup operation. • Incremental backups are performed at the file level. Especially not efficient if a large file is modified because the full file will be part of the incremental backup.	No	Yes

OCI Object Storage Retention Policy Summary

OCI object storage retention policies offer administrators the ability to meet regulatory requirements for records management and retention. For an independent assessment of the OCI object storage retention rules feature's ability to meet regulatory requirements for record management and retention, see Cohasset Associate's SEC 17a-4(f), FINRA 4511(c), CFTC 1.31(c)-(d) and MiFID II Compliance Assessment.

Different types of retention policies are available to cover a wide range of use cases.

POLICY	USE CASE	COMPLIANCE TYPE
Time-bound – specify a time duration	• Your industry might require data retention for a defined time duration • Data retention regulations might also require locked retention settings • Locked retention settings mean, the only change allowed is to increase the retention duration	Regulatory
Time-bound – specify a time duration that might change	• Your internal business requirements might require data retention • While data retention is required, that time period could change	Data Governance
Indefinite – object modification is prevented until retention rule is removed	• Requirement is to preserve certain business data in response to potential or on-going litigation • A legal hold does not have a defined retention period and remains in effect until removed	Legal Hold
Retention lock	• Your company data retention regulations might also require that you lock the retention settings	Data Governance

OCI Archive Object Storage Support

OCI object storage supports different tiers of object storage, including the inexpensive archive object storage. The ZFS Storage cloud backup supports writing to both the standard and archive object tier.

The use of Object Lifecycle Policies or auto-tiering to Infrequent Access object storage in OCI is not supported by the cloud backup feature. If less expensive object storage is desired, use an archive bucket for the backup's data target.

When using archive storage, separate buckets must be used for metadata and the backup objects. This is detailed in the section Using OCI Object Retention Policies With ZFS Storage Cloud Backups.

PROCESS OVERVIEW

A summary of the configuration steps are as follows:

- **CLI installation and Certification Creation**
    - Install CLI
    - Create certificates
    - Generate fingerprint for the certificate
    - Create configuration file
- **ZFS Storage Appliance (both) – OCI Account Creation**
    - Create OCI account on both target and source
- **Local System and ZFS Storage Appliance (target) – OCI Target Creation**
    - Create encrypted project and share with compression enabled for the target destination
    - Create the target and bucket
    - List the bucket
- **ZFS Storage Appliance (source) – OCI Cloud Service and Target Configuration**
    - Enable cloud service
    - Configure HTTP service
    - Create OCI-compatible target

The systems referred to in this document are referred to as noted in the following table.

Component Description	Shortened Name	Purpose
Local system or laptop	local	External system on which the oci command line tool has been installed
ZFS Storage Appliance #1	source appliance or source	ZFS Storage Appliance that is the source of the cloud snapshot backup
ZFS Storage Appliance #2	target appliance or target	ZFS Storage Appliance that provides the OCI-compatible object storage target that is the destination of the cloud snapshot backups

CLI INSTALLATION AND CERTIFICATE CREATION

On the local system, install the CLI.

Specific configuration steps and visibility into the OCI-compatible object store must be completed with the OCI command interface that are installed on a local system like a server or laptop.

The oci command line interface is installed on a local system, running at least Python version 3.5+. The local system can be either Linux, Windows, or MacOS. The CLI can be downloaded manually or using an installer script.

Install CLI with installer script method.

```
local% bash -c "$(curl -L \
https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh)"
```

See appendix A for sample dialog output of this command.

Install CLI with manual method.

The manual process will use curl to download the install script after which, the installer is run with the options noted.

```
local% curl -L -O \
https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh
```

```
local% ./install.sh --accept-all-defaults
```

On the local system, create public and private keys (oci_api_key.pem and oci_api_key_public.pem).

When creating a cloud target in OCI, certificates are provided as part of your OCI account and are accessible from the user profile section.

When creating an OCI-compatible object store target on a ZFS Storage Appliance, the certificates must be created as shown below. The example is shown for Linux;

A PEM (Privacy Enhanced Mail) file contains an X.509 certificate (whose structure is defined using Abstract Syntax Notation One, or ASN.1), encoded using the ASN.1 DER (Distinguished Encoding Rules), which has been encoded using Base64. The certificate is then anchored between two plain text lines (BEGIN CERTIFICATE and END CERTIFICATE).

More detail on generating the private and public keys can be found in the online OCI documentation at https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm#How

Key creation steps run on the local system

- Ensure that you are in the home directory for the user.
  ```
  local% cd ~
  ```

- Create the .oci directory.
  ```
  local% mkdir .oci
  ```

- Generate an RSA private key, optionally providing arguments to the command to specify the cipher used and the size of the private key in bits. If a passphrase is desired for the key, enter it when prompted.
  ```
  local% openssl genrsa -out ~/.oci/oci_api_key.pem -aes128 2048
  ```

  Using the default values:
  ```
  local% openssl genrsa -out ~/.oci/oci_api_key.pem
  ```

- Reduce permissions on the private key.
  ```
  local% chmod go-rwx ~/.oci/oci_api_key.pem
  ```

- Generate the public key.
  ```
  local% openssl rsa -pubout -in ~/.oci/oci_api_key.pem -out /.oci/oci_api_key_public.pem
  ```

- Generate a fingerprint of the public key. If a passphrase was used when creating the private key, enter it when prompted.

```
local% openssl rsa -pubout -outform DER -in ~/.oci/oci_api_key.pem | openssl md5 -c
```

  This command will return the fingerprint for the public key. Copy it for use in the next step.

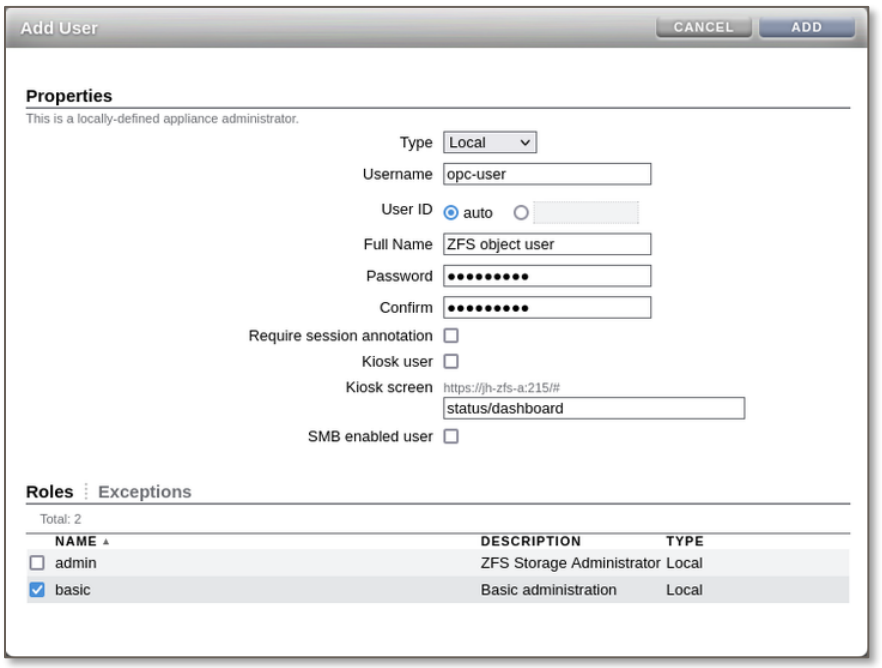
- Using the information from the above steps, use a text editor to edit or create the `~/.oci/config` and optional `~/.oci/oci_cli_rc` files which define a profile with default values to pass to the oci command line tool. An example "`zfssa`" profile is shown here:

```
[zfssa]
user=ocid1.user.oc1..opc-user
fingerprint= eb:5c:e1:c1:8a:57:26:de:ad:be:ef:30:28:b0:af:92
key_file~/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.oc1..nobody
```

  For more detail on the use of profiles and the format required for some field values, see Customization Tips later in this document. The rest of this document will assume that there is a `[zfssa]` profile defined in the `~/.oci/config` and `~/.oci/oci_cli_rc` files for all oci commands.

ACCOUNT CREATION ON THE ZFS STORAGE APPLIANCES

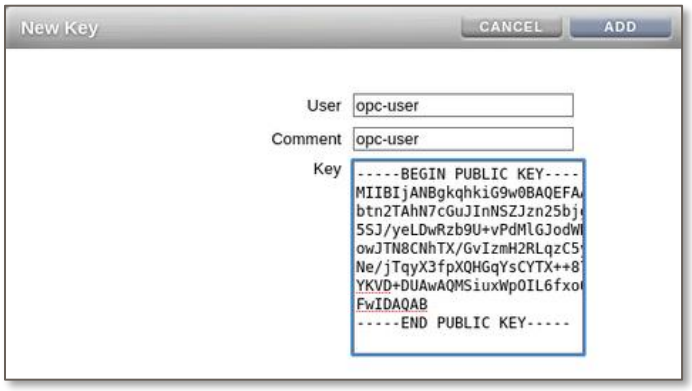
On both the source and target ZFS Storage Appliances, navigate to the Configuration→Users screen and create a local user account named "opc-user" and assign it the "basic" administration role. You may use other usernames but future examples in this paper will use the "opc-user" name.



ENABLE OBJECT STORAGE ON THE TARGET APPLIANCE

On the target appliance, the HTTP service must be configured and enabled for object storage access. Navigate to the Configuration→Services→HTTP screen and click the OCI tab. Click the "On/Off" icon (highlighted in the top circle in the image below) to enable HTTP. Then click the "Enable OCI" box and click the "Apply" button.

Next, click the "Plus" icon to add a new authorization key for the target appliance's object storage access. Enter "opc-user" in the User field and the contents of the `~/.oci/oci_api_key_public.pem` file in the Key field then click Add.



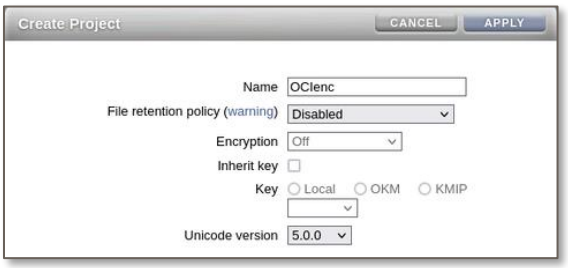
SHARE CREATION ON THE TARGET ZFS STORAGE APPLIANCE

- **Create a project for the backups on the target appliance**

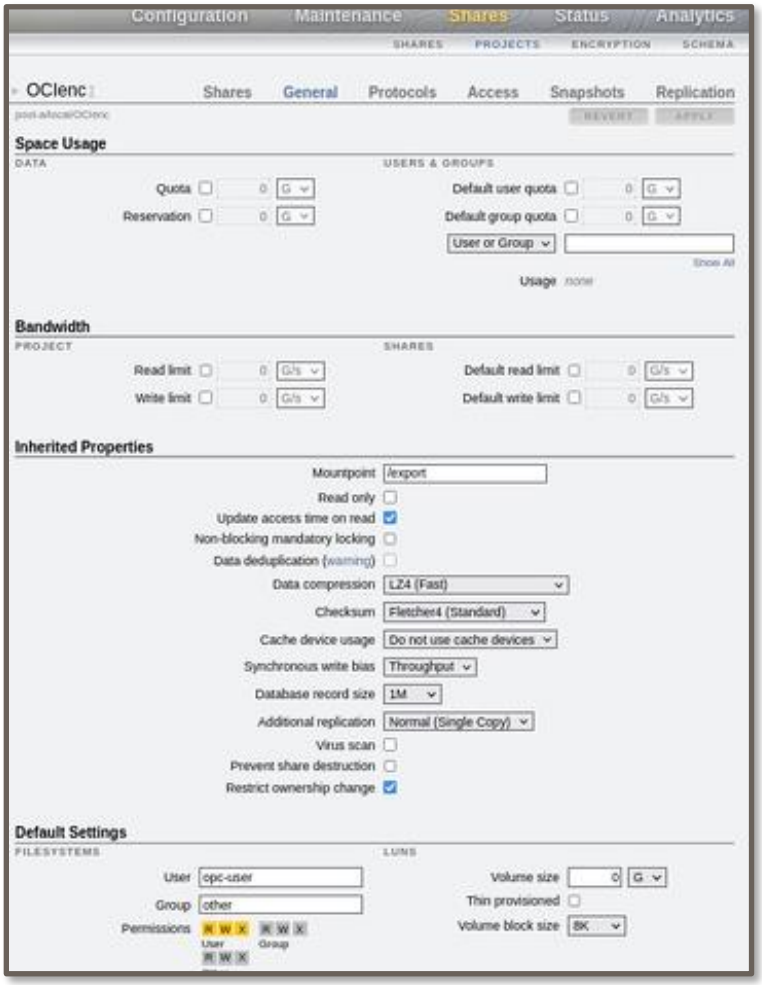
  Create a project for the shares that will hold the buckets that will contain the backup objects. This example uses the project name OClenc. See the ZFS Storage Appliance documentation on Projects and Shares for more information on the relationship between projects and the shares they contain.

  Because projects and shares underneath them can only be encrypted at creation time, set the cipher and key values here if at-rest encryption is to be used.

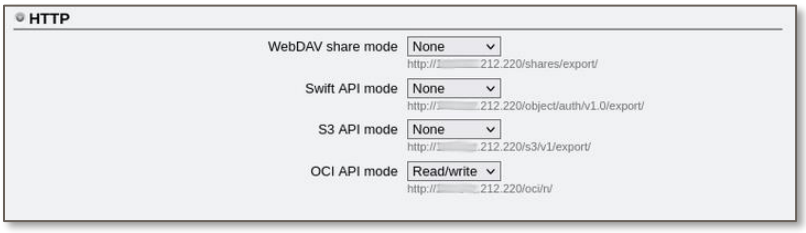
  Do not enable file retention in this project! If retention is desired, object retention can be enabled on the data bucket that will be created later in this process.



- **Define the project's properties**
  On the target appliance, modify the new project's "General" settings so that the shares created under it can inherit these values. The following properties are ones most often changed from the defaults:
  - Data deduplication: enable only if you are on a ZFS Storage Appliance that has the SSD metadata devices installed and included in the storage pool the project will be created under.
  - Select Data compression: LZ4 (optional but recommended).
  - Cache device usage: this setting affects the SSD-based L2ARC only and is ignored if these SSDs are not installed in the system. Even when installed, most backup use cases do not benefit from the use of read cache. In such cases, it is recommended to set this value to "Do not use cache devices".
  - Synchronous write bias: "throughput" is the recommended setting for streaming workloads such as backups.
  - Database record size: a value of 1M is recommended for streaming workloads such as backups.
  - Filesystem User: set to "opc-user" or whatever name was chosen earlier

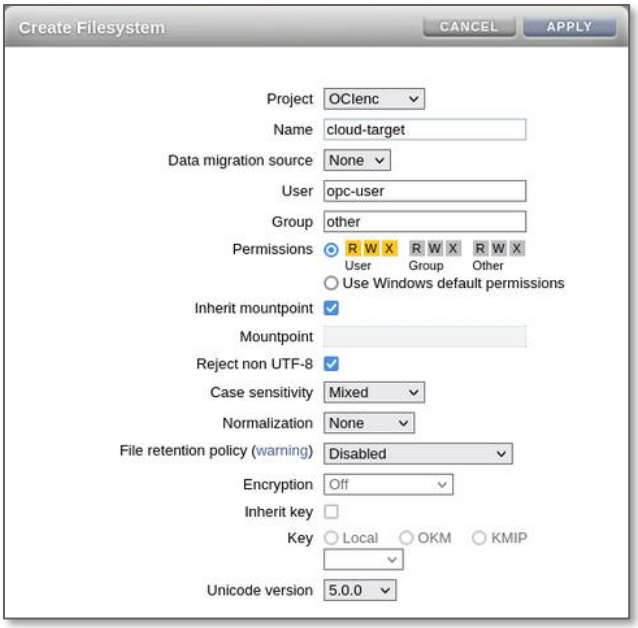


- **Enable OCI-compatible object storage for all shares in the project**
  Click on the "Protocols" tab. Ensure that the share mode for both NFS and SMB is set to "none", then scroll down to the HTTP settings. Set the OCI API mode to "Read/Write" then click the "Apply" button.



- **Create the share for the backup target**
  After creating the project and setting the properties, click on the Shares tab just to the right of the project name, then click on the "plus" icon next to the word "Filesystems". (In ZFS Storage, the words share and filesystem are often used interchangeably.)

  In the Create Filesystem window that appears, enter a name for the share. In the example shown below, "cloud-target" is the name used. It should not be necessary to change any other fields in this window. Click the "Apply" button to create the share.



CREATING THE TARGET BUCKET WITH THE CLI

Buckets and objects cannot be handled directly on ZFS Storage so the command line tool installed earlier will be used to handle that task.

- **Create the bucket on the target appliance.**

  On the local system, create the data bucket to hold the backup objects on the target with a command similar to the following:

```
local% oci os bucket create --name data-bucket --profile zfssa
```

  For information about creating the config file, see OCI Command Line Simplification later in this document.

- **On the local system, list the bucket to confirm the bucket creation.**

```
local% oci os object list –profile zfssa
```

Once the bucket is created, ZFS snapshots can then be written to it as described in the document “Configuring ZFS Storage for Cloud Snapshot Backups to OCI Object Storage”.

CUSTOMIZATION TIPS

OCI Command Line Simplification

The OCI command line tool can be used to manage target buckets and monitor OCI object stores, you might consider setting up profiles to simplify the OCI command line execution.

Profiles can be used to simplify command line input when working with a specific target. Instead defining each parameter separately on the command line to access a target, these parameter values can be grouped into a profile so that they will be specified simply by using the `--profile` argument on the command line.

The profile’s settings are defined under its title (for example, `[zfssa]`). A special name of `[DEFAULT]` may be used and values in the default section will be used unless overridden by the values in the named profile.

Two files are used to define various parts of a profile. These files usually reside in the .oci directory in the local user’s home directory, notated as ~/.oci.

The .oci/config file will contain the values for the `user`, `fingerprint`, `key_file`, and `region` fields, while the .oci/oci_cli_rc file will contain the values for the `compartment_id`, `namespace`, and `endpoint` fields.

- **~/.oci/config**
  Create a profile named zfssa in the config file by adding lines like the following:

```
[zfssa]
user=ocid1.user.oc1..opc-user
fingerprint= eb:5c:e1:c1:8a:57:26:de:ad:be:ef:30:28:b0:af:92
key_file=~/.oci/oci_api_key.pem
tenancy=ocid1.tenancy.oc1..nobody
```

  If any of the four fields shown above are not present in the config file, an error will be shown.

  The values for `key_file` and `fingerprint` fields are from the keys generated as described in the CLI Installation and Certificate Creation section of this document above.

  The user field must be in the Oracle Cloud ID (OCID) format. When connecting to OCI object storage on ZFS Storage, this field takes the format of `ocid1.user.oc1..<zfs_username>`.

  The tenancy value is required but has no equivalent in ZFS Storage. The value shown above is an acceptable dummy value.

- **~/.oci/oci_cli_rc**
  Create a profile named zfssa in the oci_cli_rc file by adding lines like the following:

```
[zfssa]
endpoint=http://ZFSSA-name-or-IP/oci
compartment_id=cloud target
namespace=cloud-target
```

  Any values for the three fields shown above not included in the oci_cli_rc file must be provided on the command line.

  The endpoint field requires a URL that uses the IP address or a DNS-defined name for the target ZFS storage followed by `/oci`.

  The `compartment_id` and `namespace` fields both have a value of the ZFS share created to be the backup target in the Share Creation On the Target ZFS Storage Appliance section of this document.

Using a Profile on the Command Line

To use a defined profile, call it by name using the `–profile <profile_name>` argument.

The following two command lines creating a bucket in the cloud-target share are equivalent when the `zfssa` profile is defined as above:

- `local%` **`oci os bucket create --endpoint "http://zfssa.example.com/oci" -ns cloud-target -c cloud-target –name data-bucket`**
- `local%` **`oci os bucket create --profile zfssa --name data-bucket`**

May 2024