

**Oracle® ZFS Storage Appliance Object
API Guide for Amazon S3 Service
Support, Release OS8.8.x**

ORACLE®

Part No: F13774-01
March 2019

Part No: F13774-01

Copyright © 2019, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Référence: F13774-01

Copyright © 2019, Oracle et/ou ses affiliés. Tous droits réservés.

Ce logiciel et la documentation qui l'accompagne sont protégés par les lois sur la propriété intellectuelle. Ils sont concédés sous licence et soumis à des restrictions d'utilisation et de divulgation. Sauf stipulation expresse de votre contrat de licence ou de la loi, vous ne pouvez pas copier, reproduire, traduire, diffuser, modifier, accorder de licence, transmettre, distribuer, exposer, exécuter, publier ou afficher le logiciel, même partiellement, sous quelque forme et par quelque procédé que ce soit. Par ailleurs, il est interdit de procéder à toute ingénierie inverse du logiciel, de le désassembler ou de le décompiler, excepté à des fins d'interopérabilité avec des logiciels tiers ou tel que prescrit par la loi.

Les informations fournies dans ce document sont susceptibles de modification sans préavis. Par ailleurs, Oracle Corporation ne garantit pas qu'elles soient exemptes d'erreurs et vous invite, le cas échéant, à lui en faire part par écrit.

Si ce logiciel, ou la documentation qui l'accompagne, est livré sous licence au Gouvernement des Etats-Unis, ou à quiconque qui aurait souscrit la licence de ce logiciel pour le compte du Gouvernement des Etats-Unis, la notice suivante s'applique :

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

Ce logiciel ou matériel a été développé pour un usage général dans le cadre d'applications de gestion des informations. Ce logiciel ou matériel n'est pas conçu ni n'est destiné à être utilisé dans des applications à risque, notamment dans des applications pouvant causer un risque de dommages corporels. Si vous utilisez ce logiciel ou ce matériel dans le cadre d'applications dangereuses, il est de votre responsabilité de prendre toutes les mesures de secours, de sauvegarde, de redondance et autres mesures nécessaires à son utilisation dans des conditions optimales de sécurité. Oracle Corporation et ses affiliés déclinent toute responsabilité quant aux dommages causés par l'utilisation de ce logiciel ou matériel pour des applications dangereuses.

Oracle et Java sont des marques déposées d'Oracle Corporation et/ou de ses affiliés. Tout autre nom mentionné peut correspondre à des marques appartenant à d'autres propriétaires qu'Oracle.

Intel et Intel Xeon sont des marques ou des marques déposées d'Intel Corporation. Toutes les marques SPARC sont utilisées sous licence et sont des marques ou des marques déposées de SPARC International, Inc. AMD, Opteron, le logo AMD et le logo AMD Opteron sont des marques ou des marques déposées d'Advanced Micro Devices. UNIX est une marque déposée de The Open Group.

Ce logiciel ou matériel et la documentation qui l'accompagne peuvent fournir des informations ou des liens donnant accès à des contenus, des produits et des services émanant de tiers. Oracle Corporation et ses affiliés déclinent toute responsabilité ou garantie expresse quant aux contenus, produits ou services émanant de tiers, sauf mention contraire stipulée dans un contrat entre vous et Oracle. En aucun cas, Oracle Corporation et ses affiliés ne sauraient être tenus pour responsables des pertes subies, des coûts occasionnés ou des dommages causés par l'accès à des contenus, produits ou services tiers, ou à leur utilisation, sauf mention contraire stipulée dans un contrat entre vous et Oracle.

Accès aux services de support Oracle

Les clients Oracle qui ont souscrit un contrat de support ont accès au support électronique via My Oracle Support. Pour plus d'informations, visitez le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> ou le site <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> si vous êtes malentendant.

Contents

Using This Documentation	9
Product Documentation Library	9
Feedback	9
Getting Started with the Oracle ZFS Storage Appliance S3 Object API Service	11
Preparing S3 API Compatible Clients	11
s3cmd	12
Boto and Boto3	13
CloudBerry	15
Cyberduck	16
Jets3t Cockpit	18
Protocol Ports Requirements for Amazon S3	20
Interoperability With Other Data Access Protocols	20
S3 API Usage Guidelines	21
Authentication for S3 API	22
Supported Authorization Versions	22
Authenticating Requests	22
Supported and Unsupported S3 API Operations	23
Supported S3 Operations on Buckets	23
Supported S3 Operations on Objects	24
Unsupported S3 Operations on Buckets	25
Unsupported S3 Operations on Objects	26
Supported and Unsupported Header Requests	26
Supported Common Request Headers	27
Unsupported Request Headers	28
Supported and Unsupported Response Headers	28
Supported Common Response Headers	29
Unsupported Common Response Headers	30

Unsupported Configuration for ZFS Data Features	31
Working with the Oracle ZFS Storage Appliance S3 Object API Service	33
Key Concepts and Elements for Accessing Resources	33
Making Requests Using the S3 Object API	35
Controlling Access to Resources Using S3 ACLs	35
Specifying S3 ACL Permissions	36
Supported S3 ACL Permissions	38
Protecting Your Data with S3 Object Versioning	39
S3 Object API Operation Command Reference	41
Operations on Services	41
GET Service	41
Operations on Buckets	43
GET Bucket	43
GET Bucket ACL	45
GET Bucket Object Versioning	47
GET Bucket Tagging	50
GET Bucket Versioning	52
HEAD Bucket	54
PUT Bucket	56
PUT Bucket ACL	57
PUT Bucket Tagging	60
PUT Bucket Versioning	62
DELETE Bucket	64
DELETE Bucket Tagging	65
Operations on Objects	67
GET Object	67
GET Object ACL	70
GET Object Tagging	73
HEAD Object	75
OPTIONS Object	77
PUT Object	77
PUT Object Copy	80
PUT Object ACL	83
PUT Object Tagging	86
POST Object	88

DELETE Object	92
DELETE Object Tagging	95
S3 Client Error Handling Reference	97
Error Response Format	97
S3 Client Error Codes	97

Using This Documentation

- **Overview** – This guide describes the Oracle ZFS Storage Appliance Object API for Amazon S3 Service support.
- **Audience** – Web application architects, developers, or administrators who need to perform the tasks or learn about the concepts in this guide.
- **Required knowledge** – Experience with web development, specifically Amazon S3 API, HTTP, XML, JSON, and REST API programming skills.

Product Documentation Library

Documentation for this product and related products are available in the storage section of the Oracle Help Center at <https://docs.oracle.com/en/storage/>.

Feedback

Provide feedback about this documentation at <http://www.oracle.com/goto/docfeedback>.

Getting Started with the Oracle ZFS Storage Appliance S3 Object API Service

The Oracle ZFS Storage Appliance S3 Object API Service enables Amazon S3 clients and applications to store content on an Oracle ZFS Storage Appliance filesystem. The following sections provide information to help you get started with using the Oracle ZFS Storage Appliance S3 Object API Service.

- [“Preparing S3 API Compatible Clients” on page 11](#)
- [“Protocol Ports Requirements for Amazon S3” on page 20](#)
- [“Interoperability With Other Data Access Protocols” on page 20](#)
- [“S3 API Usage Guidelines” on page 21](#)
- [“Authentication for S3 API” on page 22](#)
- [“Supported and Unsupported S3 API Operations” on page 23](#)
- [“Supported and Unsupported Header Requests” on page 26](#)
- [“Unsupported Configuration for ZFS Data Features” on page 31](#)

Related Information

- [Oracle ZFS Storage Appliance RESTful API Guide, Release OS8.8.x](#)
- [Using Oracle Cloud Infrastructure Object Storage Classic](#)

Preparing S3 API Compatible Clients

The Oracle ZFS Storage Appliance S3 Object API Service supports the following Amazon S3 compatible clients.

Note - Other S3 compatible clients might work but have not been tested.

- [“s3cmd” on page 12](#)

- [“Boto and Boto3” on page 13](#)
- [“CloudBerry” on page 15](#)
- [“Cyberduck” on page 16](#)
- [“Jets3t Cockpit” on page 18](#)

s3cmd

The following sections provide example information to help you install, configure, and use the s3cmd command-line tool that is an independent development of the official s3cmd tool. For additional information about this tool, refer to the official product documentation listed under [“Related Information” on page 13](#).

Installation Example: s3cmd Client

Note - The Oracle ZFS Storage Appliance S3 Object API Service does *not* support the official S3cmd command-line client tool.

In this example, the GitHub address is used to clone and install the `eucalyptus/s3cmd` client.

```
git clone https://github.com/eucalyptus/s3cmd
cd s3cmd
python setup.py install
```

Configuration Example: s3cmd Client

Prior to accessing the Oracle ZFS Storage Appliance S3 Object API Service, the `eucalyptus/s3cmd` client must be configured. The following example creates a default configuration file manually, which eliminates a need to provide an `s3cfg` file for each client session.

```
cat s3cfg

[default]
access_key = your_access_key-ID
host_base = hostname.example.com
host_bucket = hostname.example.com
service_path = /s3/v1/export/S3_enabled_share
use_https = False
secret_key = your_secret key
```

Where:

- The `access_key` property points to the key required for S3 Authentication. For additional details, see [“Authentication for S3 API” on page 22](#).

- The `host_base` and `host_bucket` properties point to the network address of the Oracle ZFS Storage Appliance system.
- The `service_path` property points to the share object store that was enabled for S3 service.
- The `use_https` property enables you set the HTTPS service. For instance, set the `use_https` property to `False` to disable the HTTPS service or set it to `True` to enable the HTTPS service.
- The `secret_key` property points to the secret key generated for S3 authentication.

Usage Examples: s3cmd Client

The following examples show interaction with the Oracle ZFS Storage Appliance S3 Object API Service. Additional usage information about the `s3cmd` command-line tool is available at the `s3tools` web site (<http://s3tools.org/usage>).

```
s3cmd -c s3cfg ls # list all buckets
s3cmd -c s3cfg mb s3://new_bucket # make a new bucket called
new_bucket
s3cmd -c s3cfg put abc.txt s3://new_bucket # upload file abc.txt to
new_bucket
s3cmd -c s3cfg ls s3://new_bucket # list all objects in new_bucket
s3cmd -c s3cfg get s3://new_bucket/abc.txt abc.2.txt # download object abc.txt from
new_bucket to a new file abc.2.txt
s3cmd -c s3cfg del s3://new_bucket/abc.txt # delete object abc.txt from
new_bucket
s3cmd -c s3cfg rb s3://new_bucket # delete bucket new_bucket
s3cmd -c s3cfg <cmd> <parameters> --deug # enable debug mode
```

Related Information

- See GitHub web site (<https://github.com/eucalyptus/eucalyptus/wiki/HowTo-use-s3cmd-with-Eucalyptus>) for "How to Use S3cmd with Eucalyptus"
- See `s3tools` web site (<http://s3tools.org/usage>) for Amazon S3 tools and usage
- See Oracle's Cloud-based documentation for [Using S3 API-Compatible Clients to Access Storage Classic](#)

Boto and Boto3

The following sections provide example information to help you install, configure, and use the Boto and Boto3 command-line tools. For additional information about these tools, refer to the official product documentation listed under [“Related Information” on page 15](#).

Note - Boto and Boto3 are client functions in Amazon Web Services (AWS) Software Development Kit (SDK) for python. Boto3 is the next generation of Boto and is available for general use.

Installation Example: Boto and Boto3

Install the latest version of Boto or Boto3 using pip, for example:

```
pip install boto
```

```
pip install boto3
```

Configuration Example: Boto and Boto3

Prior to using Boto (or Boto3), you need to set up authentication credentials. Authentication credentials can be configured in multiple ways. For instance, you can pass authentication credentials as parameter methods, environmental variables, or within a file such as a shared credentials file or an AWS configuration file. The following example defines the authentication credentials in an AWS configuration file.

```
cat ~/.aws/credentials

[default]

aws_access_key_id = YOUR_ID

aws_secret_access_key = YOUR_SECRET
```

Usage Examples: Boto and Boto3

The following Boto and Boto3 examples show interaction with the Oracle ZFS Storage Appliance S3 Object API Service.

Boto:

```
#!/usr/bin/env python

import boto
import boto.s3.connection

access_key = 'coma-04042017'
secret_key = 'd0f8c646dc4303930c547b85ef549ce80aa0709f4720436ab8f24afeaebf80b9'

conn = boto.connect_s3(
    aws_access_key_id = coma-04042017,
    aws_secret_access_key =
    d0f8c646dc4303930c547b85ef549ce80aa0709f4720436ab8f24afeaebf80b9,
    host = "x4200-85.us.example.com", # Storage appliance host name.
```

```

port = 443,                # Set to HTTPS port number for HTTPS connection.
is_secure=True,           # Set to True for HTTPS connection.
debug = 2,
path = "/s3/v1/export/coma", # Configured share S3 filesystem.
calling_format=boto.s3.connection.OrdinaryCallingFormat()
    )

bucket = conn.create_bucket('new_bucket')

```

Boto3:

```

#!/usr/bin/env python

import boto3

access_key = 'open-sesame'
secret_key = 'ddc37fc17a3837dcb4757612cefa8f4cf0be9508b51b6d6a1087c24e3d7da95f'

b3_session = boto3.Session(aws_access_key_id=access_key,
                           aws_secret_access_key=secret_key,
                           region_name='lalaland')

b3_client = b3_session.client('s3', endpoint_url="https://192.0.2.0/s3/v1/export/
mystore")

bucket = b3_client.create_bucket(Bucket="newbucket")

```

Related Information

- See GitHub web site (<https://github.com/boto/boto>) for the latest versions of Boto or Boto3.
- See the Python Boto Organization web site (http://docs.pythonboto.org/en/latest/boto_config_tut.html) for Boto documentation.
- See Boto3 web site (<https://boto3.readthedocs.io/en/stable/index.html>) for the Boto3 documentation.

CloudBerry

The following sections provide example information to help you install, configure, and use CloudBerry backup tools. For additional information about this tool, refer to the official CloudBerry web site.

Installation: CloudBerry

Download and install the appropriate version from the official CloudBerry web site.

Configuration Example: CloudBerry

The following CloudBerry Client configuration is based on Windows 10. To configure an S3 compatible account, perform the following:

1. In the Select Cloud Storage dialog box, choose S3 Compatible.
2. In the S3 Storage Account dialog box, do the following:
 - Enter the access key, secret key, and the service point.
Note that the service point is provided on the Oracle ZFS Storage Appliance share protocol. For example, `https://appliance_hostname.example.com/s3/v1/export/S3_enabled_sharename/`. Do not omit the trailing '/
 - In the Bucket name field, specify a new bucket name or an existing one.
 - Click Advanced Settings and clear the SSL check box.

Usage: CloudBerry

Start backing up files by (1) selecting the s3 compatible account that you just created and (2) following the CloudBerry backup plan and restore plan wizard.

Cyberduck

The following sections provide example information to help you install, configure, and use the Cyberduck S3 compatible browser. For additional information about this S3 compatible browser, refer to the official Cyberduck web site.

Note - Cyberduck is a libre multiple storage protocol browser for Apple Macintosh and Microsoft Windows.

Installation: Cyberduck

Download and install the appropriate version from the official Cyberduck web site.

Configuration Example: Cyberduck

The configuration is based on a Cyberduck client on Windows 10. The default Cyberduck S3 profile does *not* support the Oracle ZFS Storage Appliance S3 API. An Oracle ZFS Storage Appliance cyberduck profile template must be created manually. See the following examples:

Oracle ZFS Storage Appliance S3 (HTTPS) Cyberduck Profile:

The Oracle ZFS Storage Appliance S3 (HTTPS) Cyberduck profile uses S3 signature v4 (recommended version).

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "https://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Protocol</key>
    <string>s3</string>
    <key>Vendor</key>
    <string>s3-https</string>
    <key>Scheme</key>
    <string>https</string>
    <key>Description</key>
    <string>S3 (HTTPS)</string>
    <key>Default Port</key>
    <string>443</string>
    <key>Hostname Configurable</key>
    <true/>
    <key>Port Configurable</key>
    <true/>
    <key>Context</key>
    <string>/s3/v1/export/SHARE</string>
  </dict>
</plist>
```

Oracle ZFS Storage Appliance AWS2 Signature Version (HTTPS) Cyberduck Profile:

The Oracle ZFS Storage Appliance AWS2 Signature Version (HTTPS) Cyberduck profile uses S3 signature v2.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "https://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Protocol</key>
    <string>s3</string>
    <key>Vendor</key>
    <string>s3-aws2-https</string>
    <key>Scheme</key>
    <string>https</string>
    <key>Description</key>
    <string>S3 AWS2 Signature Version (HTTPS)</string>
    <key>Default Port</key>
    <string>443</string>
    <key>Hostname Configurable</key>
    <true/>
```

```
<key>Port Configurable</key>
<true/>
<key>Authorization</key>
<string>AWS2</string>
<key>Context</key>
<string>/s3/v1/export/SHARE</string>
</dict>
</plist>
```

In both templates, you need to change "SHARE" in the "Context" keyword to the share that has s3 service enabled. For example, you can change it to `/s3/v1/export/S3_enabled_share`.

```
<key>Context</key>
<string>/s3/v1/export/S3_enabled_share</string>
```

Cyberduck GUI-based Configuration:

1. Double-click the profile file you just created. An S3 configuration dialog box appears.
2. In the S3 configuration dialog box, specify the server address that points to the appliance. For example, `appliance_hostname.example.com`
3. In the S3 configuration dialog box, click Close. An S3 bookmark for the Oracle ZFS Storage Appliance system appears.
4. Double-click the bookmark. A Login dialog box appears.
5. In the Login dialog box, specify the required Access Key and Secret Keys, then click Login. You have successfully logged in to your Cyberduck S3 account.

Usage: Cyberduck

Cyberduck works as a file explorer, enabling you to browse Amazon S3 storage service such as a hard disk.

Jets3t Cockpit

The following sections provide example information to help you install, configure, and use the Jets3t Cockpit application. For additional information about this application, refer to the official product documentation listed under [“Related Information” on page 20](#).

Note - Jets3t is a free, open-source Java toolkit and application suite for Amazon S3, Amazon CloudFront, and Google Storage for Developers. Jets3t Cockpit is a graphical Java application for viewing and managing AWS S3 content.

Installation: Jets3t Cockpit

Download the zip package from the official Jets3t site (<http://www.jets3t.org/downloads.html>). Unzip the package. Download Java JRE and JDK to use appliance and library.

Configuration Example: Jets3t Cockpit

To configure Jets3t Cockpit on Windows 10, follow these steps:

- Under the jets3t package root, look for "RestS3Service, the edit configs/jets3t.properties as follows:
 - In the s3service.s3-endpoint= *property* , specify the network name for the appliance connection.
 - Specify the shared repository that is enabled for S3 by inserting the following line immediately after the line s3service.s3-endpoint=:


```
s3service.s3-endpoint-virtual-path=/s3/v1/export/S3_enabled_share
```

 Replace *S3_enabled_share* with the name of the S3 enabled share.
 - Change the property value for buckets= from false to true.

```
###
# RestS3Service
###

s3service.https-only=true      # if required, change from 'true' to 'false'

s3service.s3-endpoint=hostname.example.com # Storage appliance host name.
s3service.s3-endpoint-virtual-path=/s3/v1/export/S3_enabled_share # insert this line.
s3service.s3-endpoint-http-port=80
s3service.s3-endpoint-https-port=443
s3service.disable-dns-buckets=true # change from 'false' to 'true'.

s3service.default-bucket-location=US

s3service.enable-storage-classes=true
s3service.default-storage-class=STANDARD
```

- Browse to the bin directory and double-click cockpit.bat. (/cockpit.sh if in linux)
The Jets3t Cockpit Login dialog box appears.
- In the Jets3t Cockpit Login dialog box, click the Direct Login tab, and enter the Access and Secret Keys, then click Login.
You have successfully logged in to the Jets3t Cockpit account.

Usage: Jets3t Cockpit

Use the Jets3t Cockpit interface like a file explorer. If you want to use the jets3t library, refer to the Jets3t programmer guide and code samples for more information.

Related Information

- See Jets3t code samples on the official Jets3t site (<http://www.jets3t.org/toolkit/code-samples.html>)
- See the Jets3t programmer guide on the official Jets3t site (<http://www.jets3t.org/toolkit/guide.html>)

Protocol Ports Requirements for Amazon S3

The Oracle ZFS Storage Appliance S3 Object API operates on the following ports:

TABLE 1 S3 Protocol Ports

Port Number	Protocol
443	HTTPS
80	HTTP

Note - HTTP(S) requests sent to Oracle ZFS Storage Appliance that start with /s3/ are intended for the Oracle ZFS Storage Appliance S3 API.

Interoperability With Other Data Access Protocols

The following list defines interoperability limitations between the S3 protocol and other data access protocols:

Note - Other data access protocols include SWIFT API, NFS, SMB, WebDAV, and so on.

- **Container and Bucket** – A Swift container will appear as a bucket through S3 and a bucket as a container through Swift.
- **S3 Access** – Any change made by another protocol to the Oracle ZFS Storage Appliance Access Control Lists (ACLs) will have a direct effect on S3 access. The S3 Object API automatically ignores the Oracle ZFS Storage Appliance ACLs that cannot be mapped to S3

ACLs. Access to an object or bucket is dependent on the appliance ACL associations. Do *not* change the appliance ACLs on a filesystem or any of its contents via another protocol.

- **Read and Write Mode** – It is advisable to keep all other protocols in read-only mode when S3 is in read/write mode. If an object is edited by another protocol, users might see unexpected results the next time a request is made to retrieve the object. For instance, an edit from another protocol will be detected and checksums will be recalculated the next time the object is accessed using the S3 API.
- **DLO and Versioning Operations** – Complex operations like Dynamic Large Object (DLO) and versioning are *not* supported for interoperability between SWIFT and S3. These operations, which are very protocol specific, will not work across object protocols. DLO and versioned objects will appear like normal objects from the other protocol. It is advisable to maintain caution when dealing with such objects as accidental deletion or editing could break the functionality for the other protocol.
- **User-Defined Metadata** – User-defined metadata can be stored using either SWIFT or S3. User-defined metadata stored by one of the object protocols can then be retrieved by the other protocol. Note that S3 only allows setting of user-defined metadata for objects, whereas Swift allows it for containers and objects. The user metadata for containers will not be visible from S3.
- **Bucket Access** – Only the object and bucket creator (owner) is permitted access to the object and bucket from the other protocols.

S3 API Usage Guidelines

The following table describes usage guidelines for naming or mapping S3 API bucket directories and objects.

TABLE 2 Usage Guidelines: S3 API

No.	Guideline Description
1	The S3 API creates sub-directories in bucket for each "/" it encounters in the object's name. For example, when the user uploads an object called "accounting/billing.pdf", a directory called "accounting" is created in the bucket.
2	Objects that are mapped to the Oracle ZFS Storage Appliance directories, such as an object with names that end with "/", cannot store content for themselves. However, content can be created in them. For example: object "foo/" cannot have any data associated with it but object "foo/bar" can.
3	Any file name not permitted by the Oracle ZFS Storage Appliance filesystem is also not permitted while creating objects in the S3 API. For example, objects names with any of the following characters are not permitted: "/", "\", ".", ". Object names containing double "/" are also not permitted (foo//bar).
4	Object names that are relative paths are not permitted.

No.	Guideline Description
	For example, ". /foo"
5	Oracle ZFS Storage Appliance limits file and directory names to 255 characters. The same character limit (255) applies to the virtual directory names and object names for the Oracle ZFS Storage Appliance S3 API.
6	<p>When bucket versioning is either enabled or suspended, take the following naming conventions into consideration:</p> <ul style="list-style-type: none">■ When versioning is enabled for a bucket, previous versions are saved in the same directory as the current version of the object.■ The current version of an object is renamed when a new object or delete marker takes its place. The new name of the current version follows the pattern: <object name>-<versionId>. For example, <code>billing.pdf-0001</code>.

Authentication for S3 API

The following sections identify general aspects of the S3 authentication process as it relates to the Oracle ZFS Storage Appliance S3 Object API Service. For detailed S3 authentication information, refer to the official Amazon S3 documentation (<https://docs.aws.amazon.com/AmazonS3/latest/dev/RESTAuthentication.html>).

- “Supported Authorization Versions” on page 22
- “Authenticating Requests” on page 22

Supported Authorization Versions

The Oracle ZFS Storage Appliance S3 Object API Service supports both AWS S3 authentication v2 and v4. Both versions require you to create an access key and an associated secret key for proving your identity to the system.

Signature Version 2 Format:

Authorization: AWS <AWSAccessKeyId>:<Signature>

Signature Version 4 Format (recommended):

signature=Hex(HMAC-SHA256(SigningKey, StringtoSign))

Authenticating Requests

For all S3 bucket operations and object operations, Amazon uses an authorization header in all requests to provide the authentication information. When a request is made, the secret key

is used to generate a signature. This signature along with the access key are sent to the server as part of the HTTP/HTTPS request. The server will retrieve the secret key using the access key and generate its own signature. When the generated signature by the system matches the signature in the request, access is granted to the user who issued the keys.

Related Information:

- [Authenticating Requests \(AWS Signature Version 4\) \(https://docs.aws.amazon.com/AmazonS3/latest/API/sig-v4-authenticating-requests.html\)](https://docs.aws.amazon.com/AmazonS3/latest/API/sig-v4-authenticating-requests.html)
- [Authenticating Requests \(AWS Signature Version 2\) \(https://docs.aws.amazon.com/AmazonS3/latest/dev/auth-request-sig-v2.html\)](https://docs.aws.amazon.com/AmazonS3/latest/dev/auth-request-sig-v2.html)

Supported and Unsupported S3 API Operations

Refer to the following sections for supported and unsupported S3 API operations on buckets and objects:

- [“Supported S3 Operations on Buckets” on page 23](#)
- [“Supported S3 Operations on Objects” on page 24](#)
- [“Unsupported S3 Operations on Buckets” on page 25](#)
- [“Unsupported S3 Operations on Objects” on page 26](#)

Supported S3 Operations on Buckets

The following table identifies bucket operations that are supported by the Oracle ZFS Storage Appliance S3 Object API Service.

TABLE 3 Supported S3 Bucket Operations

Supported Operation	Amazon S3 API Documentation Reference
GET Service GET /	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTServiceGET.html
DELETE Bucket DELETE /	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketDELETE.html
DELETE Bucket tagging	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketDELETetagging.html

Supported Operation	Amazon S3 API Documentation Reference
DELETE /?tagging	
GET Bucket (List Objects) version 2 GET /?list-type=2	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGET.html
GET Bucket (List Objects) version 1	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGET.html
GET Bucket acl GET /?acl	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETacl.html
GET Bucket tagging GET /?tagging	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETtagging.html
GET Bucket Object versions GET /?versions	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETVersion.html
GET Bucket versioning GET /?versioning	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETVersioningStatus.html
HEAD Bucket HEAD /	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketHEAD.html
PUT Bucket PUT /	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUT.html
PUT Bucket acl PUT /?acl	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html
PUT Bucket tagging PUT /?tagging	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTtagging.html
PUT Bucket versioning PUT /?versioning	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTVersioningStatus.html

Supported S3 Operations on Objects

The following table identifies object operations that are supported by the Oracle ZFS Storage Appliance S3 Object API Service.

TABLE 4 Supported S3 Object Operations

Supported Operation	Amazon S3 API Documentation Reference
DELETE Object:	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectDELETE.html

Supported Operation	Amazon S3 API Documentation Reference
DELETE /ObjectName	
Delete Multiple Objects POST /?delete	https://docs.aws.amazon.com/AmazonS3/latest/API/multiobjectdeleteapi.html
GET Object GET /ObjectName	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectGET.html
GET Object ACL GET /ObjectName?acl	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectGETacl.html
HEAD Object HEAD /ObjectName	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectHEAD.html
OPTIONS Object OPTIONS /ObjectName	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTOPTIONSubject.html If CORS is not enabled on the bucket, Amazon S3 returns a 403 Forbidden response.
POST Object POST /	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPOST.html POST object is done through HTML forms.
PUT Object PUT /ObjectName	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUT.html
PUT Object ACL PUT /ObjectName?acl	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUTacl.html
PUT Object - Copy PUT /destinationObject	https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectCOPY.html

Unsupported S3 Operations on Buckets

The following table identifies bucket operations that are not supported by the Oracle ZFS Storage Appliance S3 Object API Service, including cross-origin resource sharing (CORS).

TABLE 5 Unsupported S3 Bucket Operations

Unsupported Bucket Operation	Unsupported Bucket Operation
GET Bucket location	GET Bucket accelerate PUT Bucket accelerate
GET Bucket cors	GET Bucket website

Unsupported Bucket Operation	Unsupported Bucket Operation
PUT Bucket cors	PUT Bucket website
PUT Bucket cors	DELETE Bucket website
DELETE Bucket cors	
PUT Bucket lifecycle	GET Bucket notification
DELETE Bucket lifecycle	PUT Bucket notification
PUT Bucket policy	GET Bucket requestPayment
DELETE Bucket policy	PUT Bucket requestPayment
GET Bucket replication	GET Bucket logging
PUT Bucket replication	PUT Bucket logging
DELETE Bucket replication	
List Multipart Uploads	

Unsupported S3 Operations on Objects

The following table identifies object operations that are not supported by the Oracle ZFS Storage Appliance S3 Object API Service.

TABLE 6 Unsupported S3 Object Operations

Unsupported Object Operation	Unsupported Object Operation
GET Object torrent	POST Object restore
Initiate Multipart Upload	
Upload Part	
Complete Multipart Upload	
Abort Multipart Upload	
List Parts	

Supported and Unsupported Header Requests

The following topics identify header request behavior for the Oracle ZFS Storage Appliance S3 Object API Service.

- [“Supported Common Request Headers” on page 27](#)

- “Unsupported Request Headers” on page 28

Supported Common Request Headers

The following table identifies the common request headers supported by the Oracle ZFS Storage Appliance S3 Object API Service.

TABLE 7 Common Supported Request Headers

Supported Request Header	Description	Required
Authorization	The Authorization header field identifies the information required for request signature authentication. For more information, see "The Authentication Header" in the <i>Amazon Simple Storage Service Developer Guide</i> (https://docs.aws.amazon.com/AmazonS3/latest/dev/s3-dg.pdf).	Yes Note - For anonymous requests, this header is not required
Content-Length	The Content-Length header field represents the length of the message (<i>without the headers</i>) according to RFC 2616.	No Note - This header is required for PUTs and load XML operations
Content-MD5	The Content-MD5 header field represents the base64 encoded 128-bit MD5 digest of the message (<i>without the headers</i>) according to RFC 1864. This header is used as a message integrity check to verify that the data is the same data that was originally sent.	No
Date -or- x-amz-date	These header fields represents the current date and time according to the requester. When you specify the Authorization header, you must specify either the x-amz-date or the Date header. If you specify both, the value specified for the x-amz-date header takes precedence	Yes.
Expect	The Expect header field is used only when sending a request body with the property values: 100-continue. Note - When your application uses 100-continue as a property value, the request body is not sent until after it receives an acknowledgment. If the message is rejected based on the headers, the body of the message is not sent.	No
Host	The Oracle ZFS Storage Appliance Host FQDNT header field is required for HTTP 1.1 (<i>most toolkits add this header automatically</i>); optional for HTTP/1.0 requests.	Yes
x-amz-content-sha256	When using signature version 4 to authenticate a request, the x-amz-content-sha256 header field provides a hash of the request payload. For more information, see Amazon's official documentation for Signature Calculations for the	Yes

Supported Request Header	Description	Required
	<p>Authorization Header: Transferring Payload in a Single Chunk (AWS Signature Version 4).</p> <p>When uploading an object in chunks, you can set the value to STREAMING-AWS4-HMAC-SHA256-PAYLOAD to indicate that the signature covers only headers and that there is no payload. For more information, see Amazon's official documentation for Signature Calculations for the Authorization Header: Transferring Payload in Multiple Chunks (Chunked Upload) (AWS Signature Version 4).</p>	

Unsupported Request Headers

The following table lists request headers that are not supported by the Oracle ZFS Storage Appliance S3 Object API Service.

TABLE 8 Unsupported Request Headers

Unsupported Request Header	Notes
x-amz-security-token	The Amazon S3 header for generating a temporary security credential is not supported by the Oracle ZFS Storage Appliance S3 API.
x-amz-server-side-encryption-customer-algorithm	These Amazon S3 headers for custom server-side encryption are not supported. Oracle ZFS Storage Appliance provides its own data encryption capability that is managed through its management interfaces (BUI, CLI, and REST). For more details, see “NFS Authentication and Encryption Options” in Oracle ZFS Storage Appliance Security Guide, Release OS8.8.x
x-amz-server-side-encryption-customer-key	
x-amz-server-side-encryption-customer-key-MD5	

Supported and Unsupported Response Headers

The following topics identify response header behavior for the Oracle ZFS Storage Appliance S3 Object API Service.

- [“Supported Common Response Headers” on page 29](#)
- [“Unsupported Common Response Headers” on page 30](#)

Supported Common Response Headers

The following table identifies the response headers that are common to most Amazon S3 responses.

TABLE 9 Supported Response Headers

Supported Response Header	Description
Content-Length	The Content-Length header field identifies the response body length in bytes. Type: String Default: None
Content-Type	The Content-Type header field identifies the MIME type of the content. For example, Content-Type: text/html; charset=utf-8 Type: String Default: None
Date	The Date header field identifies the data and time of the S3 response. For example, Wed, 01 Mar 2018 12:00:00 GMT. Type: String Default: None
ETag	The ETag header field identifies a specific version of a resource (a hash of the object). The ETag reflects changes only to the contents of an object, not its metadata. The ETag is an MD5 digest of the object data. Type: String
Server	The Server header field identifies the name of the server that created the response. Type: String Valid Value: Apache
x-amz-delete-marker	The x-amz-delete-marker identifies whether the object returned was a delete marker (true) or not (false). Type: Boolean Valid Values: true false Default: false
x-amz-request-id	The x-amz-request-id is a value that is created by Amazon S3 to uniquely identify a request for troubleshooting purposes. Type: String

Supported Response Header	Description
	Default: None
x-amz-version-id	<p>When versioning is enabled, the <code>x-amz-version-id</code> identifies the version of the object. When versioning is suspended, the version ID is always null.</p> <p>Type: String</p> <p>Valid Values: null string</p> <p>Default: null</p>
x-amz-tagging-count	<p>When the count is greater than zero, the <code>x-amz-tagging-count</code> returns the count of the tags associated with the object.</p> <p>Type: String</p> <p>Default: None</p>

Unsupported Common Response Headers

The following table lists response headers that are not supported by the Oracle ZFS Storage Appliance S3 Object API Service.

TABLE 10 Unsupported Response Headers

Unsupported Response Header	Notes
x-amz-restore	The <code>x-amz-restore</code> header is not supported by the Oracle ZFS Storage Appliance S3 API Service.
x-amz-replication-status	The <code>x-amz-replication-status</code> header is not supported by the S3 API. Oracle ZFS Storage Appliance provides its own remote replication capability, that is managed through its management interfaces (BUI, CLI, REST). For more details, see “Remote Replication” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x
x-amz-server-side-encryption	These Amazon S3 headers for custom server-side encryption are not supported. Oracle ZFS Storage Appliance provides its own data encryption capability that is managed through its management interfaces (BUI, CLI, and REST). For more details, see “NFS Authentication and Encryption Options” in Oracle ZFS Storage Appliance Security Guide, Release OS8.8.x .
x-amz-server-side-encryption-aws-kms-key-id	
x-amz-server-side-encryption-customer-algorithm	
x-amz-server-side-encryption-customer-key-MD5	
x-amz-id-2	The <code>x-amz-id-2</code> token is not supported by the Oracle ZFS Storage Appliance S3 API Service.

Unsupported Response Header	Notes
	Use the <code>x-amz-request-id</code> response header value as an alternative solution.

Unsupported Configuration for ZFS Data Features

The following table identifies Oracle ZFS Storage Appliance data features that are not supported by the Oracle ZFS Storage Appliance S3 Object API Service.

TABLE 11 Unsupported Oracle ZFS Storage Appliance S3 Object API Data Features

Unsupported Feature	Notes
Encryption	The configuration of the Oracle ZFS Storage Appliance encryption data feature is not supported by the Oracle ZFS Storage Appliance S3 Object API Service. However, the ZFS encryption data feature is configurable from the appliance management interfaces (BUI, CLI and REST). For details, see “NFS Authentication and Encryption Options” in <i>Oracle ZFS Storage Appliance Security Guide, Release OS8.8.x</i> .
Replication	The configuration of the Oracle ZFS Storage Appliance replication data feature is not supported by the Oracle ZFS Storage Appliance S3 Object API Service. However, the ZFS replication data feature is configurable from the appliance management interfaces (BUI, CLI and REST). For details, see “Remote Replication” in <i>Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x</i> .
Snapshot	The configuration of the Oracle ZFS Storage Appliance snapshot data feature is not supported by the Oracle ZFS Storage Appliance S3 Object API Service. However, the ZFS snapshot data feature is configurable from the appliance management interfaces (BUI, CLI and REST). For details, see “Snapshot Space Management” in <i>Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x</i> .

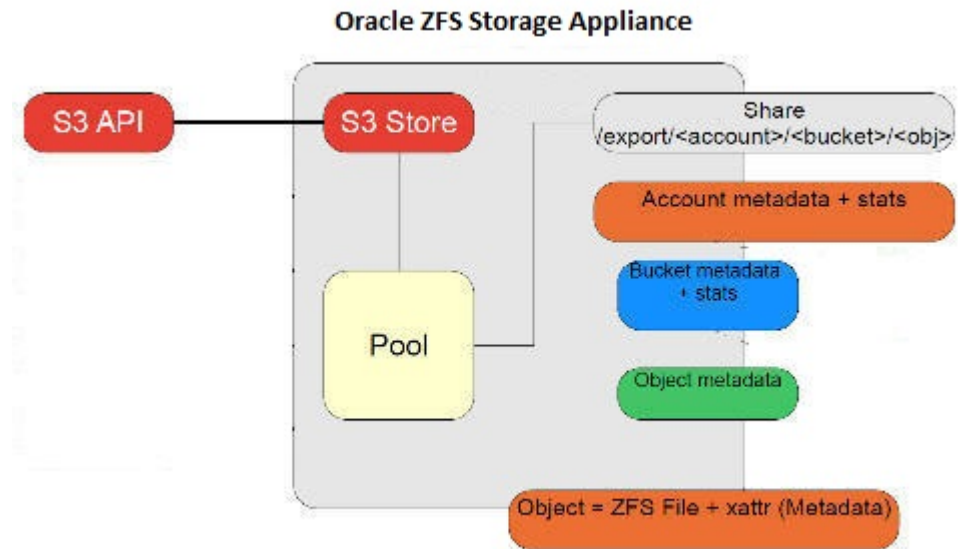
Working with the Oracle ZFS Storage Appliance S3 Object API Service

The following sections describe information about accessing resources, controlling access to resources, as well as supported versioning options to protect your data.

- [“Key Concepts and Elements for Accessing Resources” on page 33](#)
- [“Making Requests Using the S3 Object API” on page 35](#)
- [“Controlling Access to Resources Using S3 ACLs” on page 35](#)
- [“Protecting Your Data with S3 Object Versioning” on page 39](#)

Key Concepts and Elements for Accessing Resources

The Oracle ZFS Object Storage Model is a share filesystem, where each share is mapped to a project or tenant. Each share is an account. Each account-share holds buckets and each bucket holds data objects. The S3 API uses these key elements to access the appliance resources. An illustration and description of these key elements follow.



- **Share** – A share is a collection of S3 buckets. In the Oracle ZFS Storage Appliance object store implementation, the exported share name represents the account name, for example: `/export/<account_name_share_mount_point>`.

Note - A share is typically associated with a departmental group-type entity. Access to resources within a share are managed by user roles. For more information about appliance user roles, see “[Configuring Users](#)” in *Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x*.

- **Bucket** – A bucket is a user-definable element in the object data path. It is a container for storing S3 data objects. In the AWS S3 architecture, buckets and objects are known as resources. For additional information about the use and configuration of buckets, refer to the official AWS S3 web site (<https://docs.aws.amazon.com/AmazonS3/latest/dev/Introduction.html>).
- **Object** – Objects represent the entities stored in a bucket. Each object within a bucket is uniquely identified by a key name and version ID. On Oracle ZFS Storage Appliance, each object consist of a file and a set of metadata that describes the object.

Making Requests Using the S3 Object API

To send a request message to Oracle ZFS Storage Appliance using the S3 Object API, the request message should include the following entities:

- **Request Type** – The request type states the action to be performed on a resource that is identified in the URI, for example:

```
GET https://<appliance>:443/s3/v1/export/<sharename>/<bucketname>/<objectname>
```

Where:

- *GET* is the action to be performed on the resource identified in the URI. For a list of supported actions on buckets and objects, see [“Supported and Unsupported S3 API Operations” on page 23](#).
- `https://<appliance>:443/s3/v1/export/<share_mountpoint_path>/<bucketname>/<objectname>` is the request URI. The request URI identifies the resources upon which to perform actions.

Where:

- *appliance* represents the network address or DNS name of the Oracle ZFS Storage Appliance system.
- *sharename* represents the S3 account name of the share mount point upon which the action is to be performed.
- *bucketname* represents the name of the bucket upon which the action is to be performed.
- *objectname* represents the name of the object upon which the action is to be performed.
- **Request-Header Fields** – The request-header fields act as request modifiers that pass additional information to the S3 appliance. The request headers appear after the request line in the message body. For a list of supported request headers, see [“Supported and Unsupported Header Requests” on page 26](#).

Controlling Access to Resources Using S3 ACLs

Access to AWS S3 resources are, by default, private. Only the owner of a resource has access. Optionally, resource owners can grant resource permission to other users by specifying resource-based policy options, such as Access Control Lists (ACLs).

Note - Other AWS S3 resource-based policy options such as Bucket Policies and User Policies are not supported by the Oracle ZFS Storage Appliance S3 API Service. These AWS S3 policies are similar to the appliance roles that are granted to users. For more information about the Oracle ZFS Storage Appliance roles, see [“Configuring Users” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x.](#)

Note - To support a unified view of the Oracle ZFS Storage Appliance filesystem from other appliance-supported protocols, S3 ACLs are automatically mapped to the equivalent appliance filesystem ACLs. For additional information about Oracle ZFS Storage Appliance ACLs, see [“Access Control Lists for Filesystems” in Oracle ZFS Storage Appliance Administration Guide, Release OS8.8.x.](#)

To better understand how to manage appliance resource permissions using AWS S3 ACLs, see the following topics.

- [“Supported and Unsupported Header Requests” on page 26](#)
- [“Setting ACL Policy Permissions in a Request” on page 37](#)
- [“Supported Amazon S3 Predefined User Groups” on page 37](#)
- [“Supported S3 ACL Permissions” on page 38](#)

For further details about managing access permissions with AWS S3 ACLs, refer to the official product documentation on Amazon's web site (<https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html#specifying-grantee>).

Specifying S3 ACL Permissions

S3 ACLs enable you to manage access to buckets and objects. Each bucket and object has an ACL attached to it as a subresource. It defines which user or user groups are granted access, as well as the type of access granted. For instance, when a request is received against a resource, the S3 API checks the corresponding ACL to verify that the requester has the necessary access permissions. Each time you create a bucket or object, the S3 API creates a default ACL Policy that grants the resource owner full access control over the newly created resource as shown in the following example.

Example: Default ACL Policy for new bucket or object.

```
<?xml version="1.0" encoding="UTF-8"?>
<AccessControlPolicy xmlns="https://s3.amazonaws.com/doc/2018-05-23/">
  <Owner>
    <ID>***Owner-Canonical-User-ID***</ID>
    <DisplayName>Bucket Owner Display Name</DisplayName>
  </Owner>
```

```

<AccessControlList>
  <Grant>
    <Grantee xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
      xsi:type="Canonical User">
      <ID>***Canonical-User ID or Group URI ***</ID>
      <DisplayName>Appliance Grantee's username</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
</AccessControlPolicy>

```

Where:

- <Owner> reflects the appliance canonical user ID for the owner that created the bucket.
- <Grant> reflects the name of the grantee, and the permission granted.

Note that the default ACL Policy includes only one <Grant> element for the bucket owner. To grant bucket permission to others, you need to add a <Grant> element for each additional user or predefined group. Each <Grant> element must always identify the name of the grantee, as well as the permissions granted.

- <Grantee> reflects the name of an individual or predefined group receiving access permission.

The grantee can either be an authorized appliance user or an S3 predefined group. When granting access to individual appliance users, you need to specify the canonical user ID associated with the appliance user account. When granting permission to an S3 predefined group, you need to specify the predefined group URI. For a list of supported predefined groups, see [“Supported Amazon S3 Predefined User Groups” on page 37](#).
- <Permission> reflects the type of access permission that is being granted to the grantee. For a list of supported ACL permissions, see [“Supported S3 ACL Permissions” on page 38](#).

Setting ACL Policy Permissions in a Request

Use one of the following request methods when using S3 APIs (such as PUT/GET/DELETE) to set access policy permissions:

- When creating resources – Set the ACL permissions in the request's HTTP header.
- When editing ACLs associated with existing resources – Set the ACL permissions either in the request's HTTP header or in the request's body.

Supported Amazon S3 Predefined User Groups

The following table describes the supported Amazon S3 predefined user groups.

TABLE 12 Amazon S3 Predefined User Groups

Predefined User Group	Description
All Users Group	<p>The All Users Group is represented by the following URI: <code>http://acs.amazonaws.com/groups/global/AllUsers</code></p> <p>Access permission to this group enables anyone to access the resource. The requests can either be signed (authenticated) or unsigned (anonymous).</p> <p>Note - Unsigned requests omit the Authentication header in the request. Anonymous users will be mapped to the user <code>nobody</code> in Oracle ZFS Storage Appliance.</p> <p>For security reasons, a resource owner should never grant the All Users Group any of the following permissions: <code>WRITE</code>, <code>WRITE_ACP</code>, or <code>FULL_CONTROL</code>.</p>
Authenticated Users Group	<p>The Authenticated Users Group is represented by the following URI: <code>http://acs.amazonaws.com/groups/global/AuthenticatedUsers</code></p> <p>This group represents all Oracle ZFS Storage Appliance authenticated user accounts. Access permission to this group enables any authenticated user access to the resource. Therefore, when using this group, all requests must be signed (authenticated).</p>

Supported S3 ACL Permissions

The following tables describe the supported permissions for primary and canned ACLs:

- [Table 13, “Primary ACL: Grantee Supported Permissions,” on page 38](#)
- [Table 14, “Canned ACL: Supported Group Permissions,” on page 39](#)

Note - You can specify only one canned ACL in a request.

TABLE 13 Primary ACL: Grantee Supported Permissions

Permission	When Granted on Bucket	When Granted on Object
READ	Enables grantee to list the objects in the bucket.	Enables grantee to read the object data and its metadata.
WRITE	Enables grantee to create, overwrite, and delete any object in the bucket.	Not applicable.
READ_ACP	Enables grantee to read the bucket ACL.	Enables grantee to read the object ACL.
WRITE_ACP	Enables grantee to write the ACL for the applicable bucket.	Enables grantee to write the ACL for the applicable object.

Permission	When Granted on Bucket	When Granted on Object
FULL_CONTROL	Allows grantee the READ, WRITE, READ_ACP, and WRITE_ACP permissions on the bucket.	Enables grantee the READ, READ_ACP, and WRITE_ACP permissions on the object.

TABLE 14 Canned ACL: Supported Group Permissions

Canned ACL	Applies To	Permissions Added To ACL
private	Bucket and object	Owner gets FULL_CONTROL. No one else has access rights (default).
public-read	Bucket and object	Owner gets FULL_CONTROL. The All Users Group gets READ access.
public-read-write	Bucket and object	Owner gets FULL_CONTROL. The All Users Group gets READ and WRITE access. For security reasons, granting this canned ACL on a bucket is generally not recommended.
authenticated-read	Bucket and object	Owner gets FULL_CONTROL. The Authenticated Users Group gets READ access.
bucket-owner-read	Object	Object owner gets FULL_CONTROL. Bucket owner gets READ access. If you specify this canned ACL when creating a bucket, the appliance S3 API ignores it.
bucket-owner-full-control	Object	Both the object owner and the bucket owner get FULL_CONTROL over the object. If you specify this canned ACL when creating a bucket, the appliance S3 API ignores it.

Protecting Your Data with S3 Object Versioning

To preserve, retrieve, and restore every version of every object stored in your Amazon versioning-enabled S3 bucket, the Oracle ZFS Storage Appliance S3 Object API Service supports the use of the following AWS S3 versioning operations.

- **Versioning-Enabled Bucket Object Operations** – The Oracle ZFS Storage Appliance S3 Object API Service supports the following versioning-enabled bucket operations:
 - Adding Objects to Versioning-Enabled Buckets
 - Listing Objects in a Versioning-Enabled Bucket
 - Retrieving Object Versions
 - Deleting Object Versions
- **Versioning-Suspended Bucket Object Operations** – The Oracle ZFS Storage Appliance S3 Object API Service supports the following versioning-suspended bucket operations:
 - Adding Objects to Versioning-Suspended Buckets

- Retrieving Object Versions from Versioning-Suspended Buckets
- Deleting Object Versions from Versioning-Suspended Buckets

For further details about AWS S3 versioning features, refer to the official product documentation on Amazon's web site (<https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html>).

S3 Object API Operation Command Reference

This reference identifies the operation commands supported by the Oracle ZFS Storage Appliance S3 Object API Service.

- [“Operations on Services” on page 41](#)
- [“Operations on Buckets” on page 43](#)
- [“Operations on Objects” on page 67](#)

Operations on Services

The Oracle ZFS Storage Appliance S3 Object API supports the following operation on services.

- [“GET Service” on page 41](#)

GET Service

Returns a list of all buckets owned by the authenticated sender of the request. Anonymous requests cannot list buckets, and you cannot list buckets that you did not create. For more details, see the following:

- [“Syntax Example” on page 41](#)
- [“Request Parameters” on page 42](#)
- [“Request Headers” on page 42](#)
- [“Response Elements” on page 42](#)
- [“Normal Response Code” on page 42](#)

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/
```

Request Parameters

The GET Service operation does not support the use of request parameters.

Request Headers

The GET Service operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Response Elements

For a list of supported elements in the XML response for the GET Service operation, see Amazon's official S3 GET Service API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTServiceGET.html>).

Normal Response Code

200

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<ListAllMyBucketsResult>
  <Owner>
    <ID>s3_user</ID>
    <DisplayName>s3_user</DisplayName>
  </Owner>
  <Buckets>
    <Bucket>
      <Name>quotes</Name>
      <CreationDate>2006-02-03T16:45:09.000Z</CreationDate>
    </Bucket>
    <Bucket>
      <Name>samples</Name>
      <CreationDate>2006-02-03T16:41:58.000Z</CreationDate>
    </Bucket>
  </Buckets>
</ListAllMyBucketsResult>
```

Operations on Buckets

The Oracle ZFS Storage Appliance S3 Object API supports the following operations on buckets:

- [“GET Bucket” on page 43](#)
- [“GET Bucket ACL” on page 45](#)
- [“GET Bucket Object Versioning” on page 47](#)
- [“GET Bucket Tagging” on page 50](#)
- [“GET Bucket Versioning” on page 52](#)
- [“HEAD Bucket” on page 54](#)
- [“PUT Bucket” on page 56](#)
- [“PUT Bucket ACL” on page 57](#)
- [“PUT Bucket Tagging” on page 60](#)
- [“PUT Bucket Versioning” on page 62](#)
- [“DELETE Bucket” on page 64](#)
- [“DELETE Bucket Tagging” on page 65](#)

GET Bucket

The GET Bucket operation returns some or all (up to 1,000) of the objects in a bucket. You can use the request parameters as selection criteria to return a subset of the objects in a bucket. For more details about this operation, see the following:

- [“Syntax Example” on page 43](#)
- [“Request Parameters” on page 44](#)
- [“Request Headers” on page 44](#)
- [“Request Elements” on page 44](#)
- [“Response Headers” on page 44](#)
- [“Normal Response Code” on page 44](#)
- [“Example Response” on page 45](#)

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucket_name>?  
list-type=2
```

Request Parameters

For a list of request parameters supported in a request query, see Amazon's official S3 GET Bucket API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/v2-RESTBucketGET.html>).

Request Headers

The GET Bucket operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The GET Bucket operation does not support the use of request elements.

Response Headers

The GET Bucket operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

For a list of supported elements in the XML response for the GET Bucket operation, see Amazon's official S3 GET Bucket API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/v2-RESTBucketGET.html>).

Normal Response Code

200

Error Response Code

The GET Bucket operation returns special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example Response

For a list of response examples, see Amazon's official S3 GET Bucket API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/v2-RESTBucketGET.html>).

GET Bucket ACL

The GET Bucket ACL operation returns the Access Control List (ACL) of a bucket. To use GET to return the ACL of the bucket, you must have READ_ACP access to the bucket. For more details about this operation, see the following:

- “Syntax Example” on page 45
- “Request Parameters” on page 45
- “Request Headers” on page 45
- “Request Elements” on page 45
- “Response Headers” on page 46
- “Response Elements” on page 46
- “Normal Response Code” on page 46
- “Error Response Code” on page 46

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucket_name>?  
acl
```

Request Parameters

The GET Bucket ACL operation does not support the use of request parameters.

Request Headers

The GET Bucket ACL operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The GET Bucket ACL operation does not support the use of request elements.

Response Headers

The GET Bucket ACL operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

For a list of supported elements in the XML response for the GET Bucket ACL operation, see Amazon's official S3 GET Bucket ACL API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETacl.html>).

Normal Response Code

200

Error Response Code

The GET Bucket API return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
GET ?acl
HTTP/1.1 200 OK
x-amz-request-id: tx318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2018 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Server: Apache
```

```
<AccessControlPolicy>
  <Owner>
    <ID>john</ID>
    <DisplayName>mary</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>jane</ID>
        <DisplayName>Bob</DisplayName>
```

```
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
</AccessControlPolicy>
```

GET Bucket Object Versioning

The GET Bucket Object Versioning operation lists metadata about all of the object versions in a bucket. Additionally, you can use request parameters as selection criteria to return metadata about a subset of all the object versions. For more details about this operation, see the following:

- “Syntax Example” on page 47
- “Request Parameters” on page 47
- “Request Headers” on page 45
- “Request Elements” on page 48
- “Response Headers” on page 48
- “Response Elements” on page 48
- “Normal Response Code” on page 48
- “Error Response Code” on page 48

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>?
versions
```

Request Parameters

For a list of request parameters supported in a request query, see Amazon's official S3 GET Bucket Object Version API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETVersion.html>).

Request Headers

The GET Bucket Object Versioning operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The GET Bucket Object Versioning operation does not support the use of request elements.

Response Headers

The GET Bucket Object Versioning operation uses only response headers that are common to most responses. For more information, [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

For a list of supported elements in the XML response for the GET Bucket Object Versioning operation, see Amazon's official S3 API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETVersion.html>).

Normal Response Code

200

Error Response Code

The GET Bucket Object Versioning API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

Note - For additional request and response examples, see Amazon's official S3 GET Bucket Object Versioning API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETVersion.html>).

```
GET /?versions
<?xml version="1.0" encoding="UTF-8"?>

<ListVersionsResult>
  <Name>bucket</Name>
  <Prefix>my</Prefix>
  <KeyMarker/>
```



```
<VersionIdMarker/>
<MaxKeys>5</MaxKeys>
<IsTruncated>>false</IsTruncated>
<Version>
  <Key>my-image.jpg</Key>
  <VersionId>003</VersionId>
  <IsLatest>>true</IsLatest>
  <LastModified>2018-10-12T17:50:30.000Z</LastModified>
  <ETag>&quot;fba9dede5f27731c9771645a39863328&quot;</ETag>
  <Size>434234</Size>
  <StorageClass>STANDARD</StorageClass>
  <Owner>
    <ID>mary</ID>
    <DisplayName>mary</DisplayName>
  </Owner>
</Version>
<DeleteMarker>
  <Key>my-second-image.jpg</Key>
  <VersionId>001</VersionId>
  <IsLatest>>true</IsLatest>
  <LastModified>2009-11-12T17:50:30.000Z</LastModified>
  <Owner>
    <ID>jill</ID>
    <DisplayName>jill</DisplayName>
  </Owner>
</DeleteMarker>
<Version>
  <Key>my-second-image.jpg</Key>
  <VersionId>002</VersionId>
  <IsLatest>>false</IsLatest>
  <LastModified>2009-10-10T17:50:30.000Z</LastModified>
  <ETag>&quot;9b2cf535f27731c974343645a3985328&quot;</ETag>
  <Size>166434</Size>
  <StorageClass>STANDARD</StorageClass>
  <Owner>
    <ID>jill</ID>
    <DisplayName>jill</DisplayName>
  </Owner>
</Version>
<DeleteMarker>
  <Key>my-third-image.jpg</Key>
  <VersionId>002</VersionId>
  <IsLatest>>true</IsLatest>
  <LastModified>2009-10-15T17:50:30.000Z</LastModified>
  <Owner>
    <ID>moe</ID>
    <DisplayName>moe</DisplayName>
  </Owner>
```

```
</DeleteMarker>
<Version>
  <Key>my-third-image.jpg</Key>
  <VersionId>001</VersionId>
  <IsLatest>false</IsLatest>
  <LastModified>2009-10-11T12:50:30.000Z</LastModified>
  <ETag>&quot;772cf535f27731c974343645a3985328&quot;</ETag>
  <Size>64</Size>
  <StorageClass>STANDARD</StorageClass>
  <Owner>
    <ID>moe</ID>
    <DisplayName>moe</DisplayName>
  </Owner>
</Version>
</ListVersionsResult>
```

GET Bucket Tagging

The GET Bucket Tagging operation returns the tag set associated with the bucket. For more details about this operation, see the following:

- [“Syntax Example” on page 50](#)
- [“Request Parameters” on page 50](#)
- [“Request Headers” on page 51](#)
- [“Request Elements” on page 51](#)
- [“Response Headers” on page 51](#)
- [“Response Elements” on page 51](#)
- [“Normal Response Code” on page 51](#)
- [“Error Response Code” on page 51](#)

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>?
tagging
```

Request Parameters

The GET Bucket Tagging operation does not support the use of request parameters.

Request Headers

The GET Bucket Tagging operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,”](#) on page 27.

Request Elements

The GET Bucket Tagging operation does not support the use of request elements.

Response Headers

The GET Bucket Tagging operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,”](#) on page 29.

Response Elements

For a list of supported elements in the XML response for the GET Bucket Tagging operation, see Amazon's official S3 GET Bucket API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETtagging.html>).

Normal Response Code

200

Error Response Code

The GET Bucket Tagging operation does not return errors associated with a bucket. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference”](#) on page 97.

Example

```
GET /?tagging
HTTP/1.1 200 OK
Date: Wed, 25 Nov 2018 12:00:00 GMT
Connection: close
```

```
Server: Apache

<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>msmith</Value>
    </Tag>
  </TagSet>
</Tagging>
```

GET Bucket Versioning

The GET Bucket Versioning operation returns the versioning state of a bucket. To retrieve the versioning state of a bucket, you must be the bucket owner. The following versioning states apply to this operation:

- When versioning is enabled on a bucket, the response is as follows:

```
<VersioningConfiguration> <Status>Enabled</Status> </VersioningConfiguration>
```
- When versioning is suspended on a bucket, the response is as follows:

```
<VersioningConfiguration> <Status>Suspended</Status> </VersioningConfiguration>
```
- When versioning is not enabled or suspended on a bucket, the response is as follows:

```
<VersioningConfiguration/>
```

For further details about the using the GET Bucket Versioning operation, see the following:

- [“Syntax Example” on page 53](#)
- [“Request Parameters” on page 53](#)
- [“Request Headers” on page 53](#)
- [“Request Elements” on page 53](#)
- [“Response Headers” on page 53](#)
- [“Response Elements” on page 53](#)
- [“Normal Response Code” on page 53](#)
- [“Error Response Code” on page 54](#)
- [“Example” on page 54](#)

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>?
versioning
```

Request Parameters

The GET Bucket Versioning operation does not support the use of request parameters.

Request Headers

The GET Bucket Versioning operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The GET Bucket Versioning operation does not support the use of request elements.

Response Headers

The GET Bucket Versioning operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

The following XML response elements are supported:

- Status
- VersioningConfiguration

For a detailed description of these response elements, see Amazon's official S3 GET Bucket Versioning API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketGETversioningStatus.html>).

Normal Response Code

200

Error Response Code

The GET Bucket Versioning operations does not support errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
GET /?versioning
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2018-03-01/">
  <Status>Enabled</Status>
</VersioningConfiguration>
```

HEAD Bucket

The HEAD Bucket operation determines whether a bucket exists and if you have permission to access it. To use this operation, you must have permissions to list content in the share target. For more details about this operation, see the following:

- [“Syntax Example” on page 54](#)
- [“Request Parameters” on page 54](#)
- [“Request Headers” on page 55](#)
- [“Request Elements” on page 55](#)
- [“Response Headers” on page 55](#)
- [“Response Elements” on page 55](#)
- [“Normal Response Code” on page 55](#)
- [“Error Response Codes” on page 55](#)
- [“Example” on page 55](#)

Syntax Example

```
HEAD https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>
```

Request Parameters

The HEAD Bucket operation does not support the use of request parameters.

Request Headers

The HEAD Bucket operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The HEAD Bucket operation does not support the use of request elements.

Response Headers

The HEAD Bucket operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

The HEAD Bucket operation does not support the use of response elements.

Normal Response Code

200 OK

Note - The HEAD Bucket operation returns the 200 OK response code when both the bucket and access permissions to the bucket exist.

Error Response Codes

- 404 Not Found — This response code is returned when the bucket does not exist.
- 403 Forbidden — This response code is returned when permissions to access the bucket do not exist.

For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

HEAD /

```
HTTP/1.1 200 OK
x-amz-request-id: tx32FE2CEB32F5EE25
Date: Fri, 10 2018 21:34:56 GMT
Server: Apache
```

PUT Bucket

The PUT Bucket operation creates a new bucket. To create a bucket with this operation, you must have an Oracle ZFS Storage Appliance user account and a valid Access Key assigned to your user account. Anonymous requests are never allowed to create buckets. When you create a bucket, you automatically become the bucket owner. Optionally, a bucket owner can grant permissions to other appliance users or predefined groups. For more details about using the PUT Bucket operation, see the following:

- “Syntax Example” on page 56
- “Request Parameters” on page 56
- “Request Headers” on page 56
- “Request Elements” on page 57
- “Response Headers” on page 57
- “Response Elements” on page 57
- “Error Response Code” on page 57

Syntax Example

```
PUT https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>
```

Request Parameters

The PUT Bucket operation does not support the use of request parameters.

Request Headers

In addition to supporting common request headers, the PUT Bucket operation also supports headers for specifying canned ACLs and specific ACL access permissions for appliance users and predefined groups. For further details about these request headers, see Amazon's official S3 PUT Bucket API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUT.html>).

Request Elements

This implementation of the PUT Bucket operation does not support the use of request elements.

Response Headers

The PUT Bucket operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

This implementation of the PUT Bucket operation does not return response elements.

Error Response Code

The PUT Bucket API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

Note - For additional request and response examples, see Amazon's official S3 PUT Bucket API documentation (<https://docs.aws.amazon.com/AmazonS3/Latest/API/RESTBucketPUT.html>).

```
PUT HTTP/1.1
x-amz-date: Sat, 07 Apr 2018 00:54:40 GMT
Authorization: authorization string
x-amz-grant-write: id="scobby", id="shaggy"
```

```
HTTP/1.1 200
```

PUT Bucket ACL

The PUT Bucket ACL operation sets permissions on an existing bucket using the Access Control List (ACL). For more details about this operation, see the following:

Note - For further details about managing access permissions using ACLs, see [“Controlling Access to Resources Using S3 ACLs”](#) on page 35.

- [“Syntax Example”](#) on page 58
- [“Request Parameters”](#) on page 58
- [“Request Headers”](#) on page 58
- [“Request Elements”](#) on page 58
- [“Response Headers”](#) on page 59
- [“Response Elements”](#) on page 59
- [“Error Response Code”](#) on page 59

Syntax Example

```
PUT https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>?acl
```

Request Parameters

This implementation of the PUT Bucket ACL operation does not support the use of request parameters.

Request Headers

The PUT Bucket ACL operation supports the following type of request headers. For more information, see [Table 7, “Common Supported Request Headers,”](#) on page 27.

- Common request headers – For more details, see [Table 7, “Common Supported Request Headers,”](#) on page 27.
- Canned ACL and Grantee Permission request headers – For more details, see Amazon's official PUT Bucket ACL documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html>).

Request Elements

The PUT Bucket ACL operation only supports the use of request elements when using a request body to specify an ACL. For a description of supported request elements, see Amazon's official PUT Bucket ACL documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTacl.html>).

Response Headers

The PUT Bucket ACL operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

This implementation of the PUT Bucket ACL operation does not return response elements.

Error Response Code

The PUT Bucket ACL operation returns special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
PUT ?acl HTTP/1.1
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2018 20:04:21 GMT
Authorization: authorization string

<AccessControlPolicy>
  <Owner>
    <ID>bob</ID>
    <DisplayName>bob</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>bill</ID>
        <DisplayName>bill</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
    <Grant>
      <Grantee xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="Group">
        <URI xmlns="">http://acs.amazonaws.com/groups/global/AllUsers</URI>
      </Grantee>
      <Permission xmlns="">READ</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

PUT Bucket Tagging

The PUT Bucket Tagging operation enables you to add a set of tags to an existing bucket. The bucket owner has this permission by default and can grant this permission to others. For more details about this operation, see the following:

- “Syntax Example” on page 60
- “Request Parameters” on page 60
- “Request Headers” on page 60
- “Request Elements” on page 60
- “Response Headers” on page 61
- “Response Elements” on page 61
- “Expected HTTP Response Code” on page 61
- “Error Response Code” on page 61

Syntax Example

PUT `https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>?tagging`

Request Parameters

This implementation of the PUT Bucket Tagging operation does not support the use of request parameters.

Request Headers

The PUT Bucket Tagging operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The PUT Bucket Tagging operation supports the use of request elements. For description of these request elements, see Amazon's official PUT Bucket Tagging documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTtagging.html>).

Response Headers

The PUT Bucket Tagging operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

This implementation of the PUT Bucket Tagging operation does not return response elements.

Expected HTTP Response Code

204 No Content

Error Response Code

This implementation of the PUT Bucket Tagging operation supports the use of the following response errors

- Special Error – MalformedXMLError where the XML provided does not match the schema.
- Common Errors – For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
PUT ?tagging HTTP/1.1
Content-Length: 1660
x-amz-date: Thu, 12 Apr 2018 20:04:21 GMT
Authorization: authorization string
```

```
<Tagging>
  <TagSet>
    <Tag>
      <Key>Project</Key>
      <Value>Project One</Value>
    </Tag>
    <Tag>
      <Key>User</Key>
      <Value>jsmith</Value>
    </Tag>
  </TagSet>
```

```
</Tagging>  
HTTP/1.1 204 No Content
```

PUT Bucket Versioning

The PUT Bucket Versioning operation enables the bucket owner to set the versioning state of an existing bucket. Supported versioning state values are as follows:

- **Enabled** – Enables versioning for the objects in the bucket. All objects added to the bucket receive a unique version ID.
- **Disabled** – Disables versioning for the objects in the bucket. All objects added to the bucket receive the version ID null.

Note - If the versioning state has never been set on a bucket, it has no versioning state; a GET versioning request does not return a versioning state value.

For more details about this operation, see the following:

- [“Syntax Example” on page 62](#)
- [“Request Parameters” on page 62](#)
- [“Request Headers” on page 63](#)
- [“Request Elements” on page 63](#)
- [“Response Headers” on page 61](#)
- [“Response Elements” on page 63](#)
- [“Expected HTTP Response Code” on page 63](#)
- [“Error Response Code” on page 63](#)
- [“Example” on page 63](#)

Syntax Example

```
PUT https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>?  
versioning
```

Request Parameters

This implementation of the PUT Bucket Versioning operation does not support the use of request parameters.

Request Headers

The PUT Bucket Versioning operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The PUT Bucket Versioning operation supports the use of the Status and VersioningConfiguration request elements. For more information about these request elements, see Amazon's official PUT Bucket Versioning documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTBucketPUTVersioningStatus.html>).

Response Headers

The PUT Bucket Versioning operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

This implementation of the PUT Bucket Versioning operation does not return response elements.

Expected HTTP Response Code

200 OK

Error Response Code

The PUT Bucket Versioning operation does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
PUT /?versioning
<VersioningConfiguration>
  <Status>Enabled</Status>
```

```
</VersioningConfiguration>  
HTTP/1.1 200 OK
```

DELETE Bucket

The DELETE Bucket operation deletes the bucket named in the URI. Note that all objects (including all object versions and delete markers) in the bucket must be deleted before the bucket itself can be deleted. For more details about this operation, see the following:

- [“Syntax Example” on page 64](#)
- [“Request Parameters” on page 64](#)
- [“Request Headers” on page 64](#)
- [“Request Elements” on page 64](#)
- [“Response Headers” on page 65](#)
- [“Response Elements” on page 65](#)
- [“Expected HTTP Response Code” on page 65](#)
- [“Error Response Code” on page 65](#)
- [“Example” on page 65](#)

Syntax Example

```
DELETE https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>
```

Request Parameters

This implementation of the DELETE Bucket operation does not support the use of request parameters.

Request Headers

The DELETE Bucket operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

This implementation of the DELETE Bucket operation does not support the use of request elements.

Response Headers

The DELETE Bucket operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

This implementation of the DELETE Bucket does not return response elements.

Expected HTTP Response Code

204 No Content

Error Response Code

The DELETE Bucket API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

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

DELETE Bucket Tagging

The DELETE Bucket Tagging operation removes a tag set from the specified bucket. For more details, see the following:

- [“Syntax Example” on page 66](#)
- [“Request Parameters” on page 66](#)
- [“Request Headers” on page 66](#)
- [“Request Elements” on page 66](#)
- [“Response Headers” on page 66](#)
- [“Response Elements” on page 66](#)
- [“Expected HTTP Response Code” on page 65](#)

- [“Error Response Code” on page 67](#)

Syntax Example

```
DELETE https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>?tagging
```

Request Parameters

This implementation of the DELETE Bucket Tagging operation does not support the use of request parameters.

Request Headers

The DELETE Bucket Tagging operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

This implementation of the DELETE Bucket Tagging operation does not support the use of request elements.

Response Headers

The DELETE Bucket Tagging operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

This implementation of the DELETE Bucket Tagging operation does not return response elements.

Expected HTTP Response Code

204 No Content

Error Response Code

The DELETE Bucket Tagging API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
DELETE /?tagging
HTTP/1.1 204 No Content
```

Operations on Objects

The Oracle ZFS Storage Appliance S3 Object API supports the following operations on objects.

- [“GET Object” on page 67](#)
- [“GET Object ACL” on page 70](#)
- [“GET Object Tagging” on page 73](#)
- [“HEAD Object” on page 75](#)
- [“OPTIONS Object” on page 77](#)
- [“PUT Object” on page 77](#)
- [“PUT Object Copy” on page 80](#)
- [“PUT Object ACL” on page 83](#)
- [“PUT Object Tagging” on page 86](#)
- [“POST Object” on page 88](#)
- [“DELETE Object” on page 92](#)
- [“DELETE Object Tagging” on page 95](#)

GET Object

The GET Object retrieves S3 objects. To use this operation, you must have READ access to the object. If READ access is granted to an anonymous user, the object is returned without an authorization header. Note that the GET Object operation, by default, returns the current version of an object. To return a different version, use the `versionId` subresource. In cases where the current version of the object is a delete marker, S3 behaves as if the object was deleted and includes `x-amz-delete-marker: true` in the response. For more details about this operation, see the following:

- “Syntax Example” on page 68
- “Request Parameters” on page 68
- “Request Headers” on page 68
- “Request Elements” on page 68
- “Response Headers” on page 68
- “Response Elements” on page 69
- “Expected HTTP Response Code” on page 69
- “Error Response Code” on page 69

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>/<objectname>
```

Request Parameters

This implementation of the GET Object operation does not support the use of request parameters.

Request Headers

The GET Object operation supports the use of the following request headers:

- Request headers that are common to all operations – For more information, see [Table 7, “Common Supported Request Headers,”](#) on page 27.
- Request headers for retrieving objects – For a description of these request headers, see Amazon's official GET Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectGET.html>).

Request Elements

The GET Object operation does not support the use of request elements.

Response Headers

The GET Object operation supports the use of response headers. For a description of the response headers supported for the GET Object operation, see Amazon's official GET Object

API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectGET.html>).

Response Elements

This implementation of the GET Object operation does not return response elements.

Expected HTTP Response Code

200 OK

Error Response Code

This implementation of the GET Object operation does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
GET /my-image.jpg
HTTP/1.1 200 OK
x-amz-request-id: tx318BC8BC148832E5
Date: Mon, 3 Oct 2016 22:32:00 GMT
Last-Modified: Wed, 12 Oct 2009 17:50:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
```

[434234 bytes of object data]

Example Response when the Latest Object is a Delete Marker

```
HTTP/1.1 404 Not Found
x-amz-request-id: 318BC8BC148832E5
x-amz-version-id: 003
x-amz-delete-marker: true
Date: Wed, 28 Oct 2018 22:32:00 GMT
Content-Type: text/plain
Connection: close
```

Example Request Getting a Specified Version of an Object

```
GET /myObject?versionId=002 HTTP/1.1
Date: Wed, 28 Oct 2018 22:32:00 GMT
Authorization: authorization string
HTTP/1.1 200 OK
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2009 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2018 12:00:00 GMT
x-amz-version-id: 002
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
[434234 bytes of object data]
```

GET Object ACL

The GET Object ACL operation returns the Access Control List (ACL) for the specified object. To use this operation, you must have READ_ACP access to the object. For more details about this operation, see the following:

- [“Syntax Example” on page 70](#)
- [“Request Parameters” on page 70](#)
- [“Request Headers” on page 71](#)
- [“Request Elements” on page 71](#)
- [“Response Headers” on page 71](#)
- [“Response Elements” on page 71](#)
- [“Normal Response Code” on page 71](#)
- [“Error Response Code” on page 71](#)

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>/<objectname>?acl
```

Request Parameters

This implementation of the GET Object ACL does not support the use of request parameters.

Request Headers

The GET Object ACL operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The GET Object ACL operation does not support the use of request elements.

Response Headers

The GET Object ACL operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

For a list of supported elements in the XML response for the GET Object ACL operation, see Amazon's official S3 GET Object ACL API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectGETacl.html>).

Normal Response Code

200 OK

Error Response Code

The GET Object API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

Sample Response

```
GET /my-image.jpg?acl
HTTP/1.1 200 OK
x-amz-request-id: tx318BC8BC148832E5
```

```
x-amz-version-id: 009
Date: Wed, 28 Oct 2018 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2018 12:00:00 GMT
Content-Length: 124
Content-Type: text/plain
Connection: close
```

```
<AccessControlPolicy>
  <Owner>
    <ID>micky</ID>
    <DisplayName>micky</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee type="CanonicalUser">
        <ID>minny</ID>
        <DisplayName>minny</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
</AccessControlPolicy>
```

Sample Request Getting the ACL of the Specific Version of an Object

```
GET /my-image.jpg?versionId=0003
HTTP/1.1 200 OK
x-amz-request-id: 318BC8BC148832E5
Date: Wed, 28 Oct 2018 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2018 12:00:00 GMT
x-amz-version-id: 0004
Content-Length: 124
Content-Type: text/plain
Connection: close
```

```
<AccessControlPolicy>
  <Owner>
    <ID>micky</ID>
    <DisplayName>micky</DisplayName>
  </Owner>
  <AccessControlList>
    <Grant>
      <Grantee xsi:type="CanonicalUser">
        <ID>minny</ID>
        <DisplayName>minny</DisplayName>
      </Grantee>
      <Permission>FULL_CONTROL</Permission>
    </Grant>
  </AccessControlList>
```



```
</AccessControlPolicy>
```

GET Object Tagging

The GET Object Tagging operation returns the tags associated with an object by sending the GET request against the tagging subresource associated with the object. To use this operation, you must have READ permissions on the object. For more details about this operation, see the following:

- [“Syntax Example” on page 73](#)
- [“Request Parameters” on page 73](#)
- [“Request Headers” on page 73](#)
- [“Request Elements” on page 73](#)
- [“Response Headers” on page 74](#)
- [“Response Elements” on page 74](#)
- [“Expected HTTP Response Code” on page 74](#)
- [“Error Response Code” on page 74](#)

Syntax Example

```
GET https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>/<objectname>?tagging
```

Request Parameters

This implementation of GET Object Tagging API does not support the use of request parameters.

Request Headers

The GET Object Tagging operation uses only request headers that are common to all operations. For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).

Request Elements

The GET Object Tagging operation does not support the use of request elements.

Response Headers

The GET Object Tagging operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

For a list of supported elements in the XML response for the GET Object Tagging operation, see Amazon's official S3 GET Object Tagging API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectGETtagging.html>).

Expected HTTP Response Code

200 OK

Error Response Code

The GET Object Tagging operation does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
GET /example-object?tagging
HTTP/1.1 200 OK
Date: Thu, 22 Sep 2018 21:33:08 GMT
Connection: close
Server: Apache
<?xml version="1.0" encoding="UTF-8"?>
<Tagging xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <TagSet>
    <Tag>
      <Key>tag1</Key>
      <Value>val1</Value>
    </Tag>
    <Tag>
      <Key>tag2</Key>
      <Value>val2</Value>
    </Tag>
  </TagSet>
```

</Tagging>

HEAD Object

The HEAD Object operation retrieves metadata from an object without returning the object itself. This operation is useful if you are interested only in an object's metadata. To use this operation, you must have READ access to the object. A HEAD request has the same options as a GET operation on an object. The response is identical to the GET response except that there is no response body. For more details about the Head Object operation, see the following:

- “Syntax Example” on page 75
- “Request Parameters” on page 75
- “Request Headers” on page 75
- “Request Elements” on page 76
- “Response Headers” on page 76
- “Response Elements” on page 76
- “Expected HTTP Response Code” on page 76
- “Error Response Code” on page 76

Syntax Example

```
HEAD https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>/<objectname>
```

Request Parameters

This implementation of the HEAD Object operation does not support the use of request parameters.

Request Headers

The HEAD Object operation supports the use of the following type of request headers:

- Request headers common to all operations – For more information, see [Table 7, “Common Supported Request Headers,”](#) on page 27
- Request headers for Head Object operations – For more information, see Amazon's official HEAD S3 Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectHEAD.html>).

Request Elements

The HEAD Object operation does not support the use of request elements.

Response Headers

This implementation of the HEAD Object operation supports the use of the `x-amz-meta-*` and the `x-amz-version-id` response headers. For more details about these response headers, see Amazon's official S3 HEAD Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectHEAD.html>).

Response Elements

This implementation of the HEAD Object operation does not return response elements.

Expected HTTP Response Code

200 OK

Error Response Code

The HEAD Object API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
HEAD /my-image.jpg
HTTP/1.1 200 OK
x-amz-request-id: 318BC8BC143432E5
x-amz-version-id: 0007
x=custom-made: custom-value
Date: Wed, 28 Oct 2018 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2018 12:00:00 GMT
ETag: "fba9dede5f27731c9771645a39863328"
Content-Length: 434234
Content-Type: text/plain
Connection: close
```

Server: Apache

OPTIONS Object

The 403 Forbidden response code is always returned for the OPTIONS Object API operation. The Oracle ZFS Storage Appliance S3 API does not support the use of a cross-origin resource sharing CORS configuration on a bucket.

Expected HTTP Response Code

403 Forbidden

PUT Object

The PUT Object operation adds an object to a bucket. To add an object to a bucket, you must have WRITE permissions on the bucket. To ensure data is not corrupted when using the PUT Object operation, you should use the Content-MD5 header. To configure your application to send the Request Headers prior to sending the request body, use the 100-continue HTTP status code. For more details about using this operation, see the following:

- [“Storage Class Options” on page 77](#)
- [“Access Permissions” on page 78](#)
- [“Syntax Example” on page 78](#)
- [“Object Versioning” on page 78](#)
- [“Request Parameters” on page 78](#)
- [“Request Headers” on page 78](#)
- [“Request Elements” on page 79](#)
- [“Response Headers” on page 79](#)
- [“Response Elements” on page 79](#)
- [“Expected HTTP Response Code” on page 79](#)
- [“Error Response Code” on page 79](#)

Storage Class Options

Oracle ZFS Storage Appliance only supports the STANDARD storage class option.

Access Permissions

To grant specific permission on an object using a request header, you can either:

- Specify a canned (predefined) ACL using the `x-amz-acl` request header – For more information, see [“Controlling Access to Resources Using S3 ACLs” on page 35](#).
- Specify access permissions explicitly using the `x-amz-grant-read`, `x-amz-grant-read-acp`, and `x-amz-grant-write-acp`, `x-amz-grant-full-control` headers – These headers map to the set of permissions S3 supports in an ACL. For more information, see [“Controlling Access to Resources Using S3 ACLs” on page 35](#).

Syntax Example

```
PUT https://<appliance>:443/s3/v1/export/<share_mount_point_path>/<bucketname>/<objectname>
```

Object Versioning

If you enable versioning for a bucket, S3 automatically generates a unique version ID for the object being stored. S3 returns this ID in the response using the `x-amz-version-id` response header. If versioning is suspended, S3 always uses `null` as the version ID for the object stored. If you enable versioning for a bucket, when S3 receives multiple write requests for the same object simultaneously, it stores all of the objects as separate versions.

Request Parameters

This implementation of the PUT Object operation does not support the use of request parameters.

Request Headers

The PUT Object operation supports the use of following request headers:

- Request headers common to all operations – For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).
- Request headers for PUT Object operations, which include `Content-Disposition`, `Content-Encoding`, `Content-Length`, `Content-MD5`, `Content-Type`, `Expect`, `x-amz-meta-`, `x-amz-tagging` – For a description of these request headers, see Amazon's official PUT

Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUT.html>).

Request Elements

The PUT Object operation does not support the use of request elements.

Response Headers

The PUT Object operation supports the use of the following response headers:

- Response headers common to all operations – For more information, see [Table 9, “Supported Response Headers,”](#) on page 29.
- The `x-amz-version-id` header – This header describes the object version.

Response Elements

This implementation of the PUT Object operation does not return response elements.

Expected HTTP Response Code

200 OK

Error Response Code

The PUT Object API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference”](#) on page 97.

Example

```
PUT /my-image.jpg
Date: Wed, 12 Oct 2018 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 11434
x-amz-grant-full-control: id="michael"
x-amz-meta-author: Janet
Expect: 100-continue
```

```
HTTP/1.1 100 Continue
```

```
HTTP/1.1 200 OK
x-amz-request-id: tx0A49CE4060975EAC
Date: Wed, 12 Oct 2018 17:50:00 GMT
ETag: "1b2cf535f27731c974343645a3985328"
Content-Length: 0
Connection: close
Server: Apache
```

When versioning is enabled on the bucket, the response includes the `x-amz-version-id` header:

```
HTTP/1.1 100 Continue
```

```
HTTP/1.1 200 OK
x-amz-request-id: tx0A49CE4060975EAC
x-amz-version-id: 0075
Date: Wed, 12 Oct 2018 17:50:00 GMT
ETag: "fbacf535f27731c9771645a39863328"
Content-Length: 0
Connection: close
```

PUT Object Copy

The PUT Object Copy operation creates a copy of a stored S3 object. A PUT Object Copy operation is the same as performing a GET and then a PUT. Adding the request header, `x-amz-copy-source`, makes the PUT operation copy the source object into the destination bucket. When copying an object, you can preserve most of the metadata (default) or specify new metadata. However, the ACL is not preserved and is set to private for the user making the request. All copy requests must be authenticated and cannot contain a message body. Additionally, you must have READ access to the source object and WRITE access to the destination bucket. To copy an object only under certain conditions, such as whether the ETag matches or whether the object was modified before or after a specified date, use the request headers `x-amz-copy-source-if-match`, `x-amz-copy-source-if-none-match`, `x-amz-copy-source-if-unmodified-since`, or `x-amz-copy-source-if-modified-since`. For more details about using the PUT Object Copy operation, see the following:

- [“Syntax Example” on page 81](#)
- [“Versioning” on page 81](#)
- [“Access Permissions” on page 81](#)
- [“Request Parameters” on page 81](#)
- [“Request Headers” on page 82](#)
- [“Request Elements” on page 82](#)

- “Response Headers” on page 82
- “Response” on page 82
- “Expected HTTP Response Code” on page 83
- “Error Response Code” on page 83

Syntax Example

For a syntax example, see Amazon's official S3 PUT Object Copy API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectCOPY.html>).

Versioning

By default, `x-amz-copy-source` identifies the current version of an object to copy. However, if the current version is a delete marker, S3 behaves as if the object were deleted.

To copy a different version, use the `versionId` subresource. If you enable versioning on the target bucket, S3 generates a unique version ID for the object being copied. This version ID is different from the version ID of the source object. S3 returns the version ID of the copied object in the `x-amz-version-id` response header in the response. Note that if you do not enable versioning or suspend versioning on the target bucket, the version ID S3 generates a null.

Access Permissions

To grant specific permission on an object using a request header, you can either:

- Specify a canned (predefined) ACL using the `x-amz-acl` request header. – For more information, see “Controlling Access to Resources Using S3 ACLs” on page 35.
- Specify access permissions explicitly using the `x-amz-grant-read`, `x-amz-grant-read-acp`, and `x-amz-grant-write-acp`, `x-amz-grant-full-control` headers. These headers map to the set of permissions S3 supports in an ACL. – For more information, see “Controlling Access to Resources Using S3 ACLs” on page 35.

Note - You can use either a canned ACL or specify access permissions explicitly. You cannot do both.

Request Parameters

This implementation of PUT Object Copy operation does not support the use of request parameters.

Request Headers

The PUT Object Copy operation supports the use of following request headers:

- Request headers common to all operations. – For more information, see [Table 7, “Common Supported Request Headers,” on page 27](#).
- Request headers for PUT Object operations, which include `x-amz-copy-source`, `x-amz-metadata-directive`, `x-amz-copy-source-if-match`, `x-amz-copy-source-if-none-match`, `x-amz-copy-source-if-unmodified-since`, `x-amz-copy-source-if-modified-since`, `x-amz-tagging-directive`. – For a description of these request headers, see Amazon's official PUT Object Copy API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectCOPY.html>).

Request Elements

The PUT Object Copy operation supports the following requests elements:

- `CopyObjectResult`
- `ETag`
- `LastModified`

For a description of the supported request elements, see Amazon's official PUT Object Copy API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectCOPY.html>).

Response Headers

The PUT Object Copy operation supports the use of the following response headers:

- Response headers common to all operations. – For more information, see [Table 9, “Supported Response Headers,” on page 29](#).
- Response headers for PUT Object operation, which include `x-amz-version-id` and `x-amz-copy-source-version-id`. – For a description of these response headers, see Amazon's official PUT Object Copy API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectCOPY.html>).

Response

This implementation of the PUT Object Copy operation does not return response elements.

Expected HTTP Response Code

200 OK

Error Response Code

The PUT Object Copy API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
PUT /my-second-image.jpg HTTP/1.1
Host: bucket.s3.amazonaws.com
Date: Wed, 28 Oct 2018 22:32:00 GMT
x-amz-copy-source: /bucket/my-image.jpg
HTTP/1.1 200 OK
x-amz-request-id: tx318BC8BC148832E5
x-amz-copy-source-version-id: 0009
x-amz-version-id: 0099
Date: Wed, 28 Oct 2018 22:32:00 GMT
Connection: close
Server: Apache

<CopyObjectResult>
  <LastModified>2009-10-28T22:32:00</LastModified>
  <ETag>"9b2cf535f27731c974343645a3985328"</ETag>
</CopyObjectResult>
```

Where `x-amz-version-id` returns the version ID of the object in the destination bucket, and `x-amz-copy-source-version-id` returns the version ID of the source object.

PUT Object ACL

The PUT Object ACL sets the Access Control List (ACL) permissions on an existing bucket object. To set ACL permissions on an existing bucket object, you must have `WRITE_ACP` permissions. You can choose to use request headers to specify the permissions, or specify the ACL in the request body. For more details about using the PUT Object ACL, see the following:

- [“Versioning” on page 84](#)
- [“Syntax Example” on page 84](#)
- [“Request Parameters” on page 84](#)

- “Request Headers” on page 84
- “Request Elements” on page 85
- “Response Headers” on page 85
- “Response Elements” on page 85
- “Expected HTTP Response Code” on page 85
- “Error Response Codes” on page 85

Versioning

The ACL for an object is set at the object version level. By default, a PUT request sets the ACL for the current version of the object. To set the ACL for a different version, use the `versionId` subresource.

Syntax Example

For syntax examples of the PUT Object ACL operation, see Amazon's official S3 API PUT Object ACL documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUTacl.html>).

Request Parameters

This implementation of the PUT Object ACL operation does not support the use of request parameters.

Request Headers

The PUT Object ACL operation supports the use of the following access-control headers to set permissions:

- `x-amz-acl` header – Use this header to specify canned ACL permissions.
- `x-amz-grant-permission` header – Use this header to individually specify the permissions for a grantee.

For more information about how to specify ACL permissions, see:

- “Controlling Access to Resources Using S3 ACLs” on page 35.
- Amazon's official S3 API documentation for PUT Object ACL (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUTacl.html>).

Request Elements

The PUT Object ACL operation supports the use of request elements when not using a request body. Note that if you use a request body, you cannot use the request headers to set an ACL. For a list of supported request elements, see Amazon's official S3 API PUT Object ACL documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUTacl.html>).

Response Headers

The PUT Object ACL operation supports the use of the following response headers:

- Response headers common to all operations. – For more information, see [Table 9, “Supported Response Headers,”](#) on page 29.
- Response headers for PUT Object operation, which include `x-amz-version-id`. – For further details about this response header, see Amazon's official S3 API PUT Object ACL documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUTacl.html>).

Response Elements

This implementation of the PUT Object ACL does not support the use of response elements.

Expected HTTP Response Code

200 OK

Error Response Codes

The PUT Object ACL operation does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference”](#) on page 97.

Example

```
PUT /my-image.jpg?acl
<AccessControlPolicy>
```

```
<Owner>
  <ID>joe</ID>
  <DisplayName>joe/DisplayName>
</Owner>
<AccessControlList>
  <Grant>
    <Grantee type="CanonicalUser">
      <ID>jack</ID>
      <DisplayName>joe</DisplayName>
    </Grantee>
    <Permission>FULL_CONTROL</Permission>
  </Grant>
</AccessControlList>
</AccessControlPolicy>
```

```
HTTP/1.1 200 OK
x-amz-request-id: tx318BC8BC148832E5
x-amz-version-id: 0055
Date: Wed, 28 Oct 2018 22:32:00 GMT
Last-Modified: Sun, 1 Jan 2018 12:00:00 GMT
Content-Length: 0
Connection: close
Server: Apache
```

Alternatively, a request can also be made to a specific version of an object, for instance:

```
PUT /my-image.jpg?acl&versionId=0099
```

PUT Object Tagging

The PUT Object Tagging operation adds a set of tags to an existing object. A tag is a key-value pair. You can associate tags with an object by sending a PUT request against the tagging subresource associated with the object. You can retrieve tags by sending a GET request. For more details, see the following:

- [“Syntax Example” on page 87](#)
- [“Request Parameters” on page 87](#)
- [“Request Headers” on page 87](#)
- [“Request Elements” on page 87](#)
- [“Response Headers” on page 87](#)
- [“Response Elements” on page 87](#)
- [“Expected HTTP Response Code” on page 87](#)

- [“Error Response Code” on page 88](#)

Syntax Example

For syntax examples of the PUT Object Tagging operation, see Amazon's official S3 API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUTtagging.html>).

Request Parameters

This implementation of the PUT Object Tagging operation does not support the use of request parameters.

Request Headers

This implementation of the PUT Object Tagging operation does not support the use of request headers.

Request Elements

For a list of supported request elements, see Amazon's official S3 API PUT Object Tagging documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPUTtagging.html>).

Response Headers

The PUT Object Tagging operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

This implementation of the PUT Object Tagging operation does not return response elements.

Expected HTTP Response Code

200 OK

Error Response Code

MalformedXMLError – The XML provided does not match the schema..

Example

```
PUT object-key?tagging HTTP/1.1
Content-Length: length
Content-MD5: pUNXr/BjKK5G2UKEexample==
x-amz-date: 20180923T001956Z
Authorization: authorization string
<Tagging>
  <TagSet>
    <Tag>
      <Key>tag1</Key>
      <Value>val1</Value>
    </Tag>
    <Tag>
      <Key>tag2</Key>
      <Value>val2</Value>
    </Tag>
  </TagSet>
</Tagging>
HTTP/1.1 200 OK
x-amz-request-id: tx236A8905248E5A01
Date: Fri, 23 Sep 2018 00:20:19 GMT
```

POST Object

The POST Object operation adds an object to a specified bucket using HTML forms.

Note - POST is an alternate form of PUT that enables browser-based uploads as a way of putting objects in buckets. Parameters that are passed to PUT in HTTP Headers are instead passed as form fields to POST in the multipart-form-data encoded message body. WRITE access is required to add an object to a bucket. To ensure that data is not corrupted traversing the network, use the Content-MD5 form field. When you use this form field, S3 checks the object against the provided MD5 value. If they do not match, S3 returns an error. Additionally, you can calculate the MD5 value while posting an object to S3 and compare the returned ETag to the calculated MD5 value. The ETag only reflects changes to the contents of an object, not its metadata.

For more details about using the POST Object operation, see the following:

- “Syntax Example” on page 89
- “Request Parameters” on page 89
- “Request Headers” on page 89
- “Request Elements” on page 87
- “Form Field Names Supported In Request” on page 89
- “Response Headers” on page 90
- “Response Elements” on page 90
- “Expected HTTP Error Response Codes” on page 90
- “Error Response Code” on page 91

Syntax Example

For request syntax examples, see Amazon's official S3 POST Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPOST.html>).

Request Parameters

This implementation of the POST Object operation does not support the use of request parameters.

Request Headers

This implementation of the POST Object operation does not support the use of request headers.

Request Elements

The request is made through an HTTP form.

Form Field Names Supported In Request

The POST Object operation supports the use of following form fields in a request.

Note - Server side encryption form fields are not supported.

Field Name	Field Name
AWSAccessKeyId	policy
acl	success_action_redirect
Cache-Control, Content-Type, Content- Disposition, Content- Encoding, Expires	success_action_status
file	tagging
key	x-amz-storage-class
x-amz-meta-*	

For a description of these supported form fields, see Amazon's official S3 POST Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPOST.html>).

Response Headers

In addition to the response headers common to all responses, this implementation of the POST Object operation can include the following response headers:

- success_action_redirect
- x-amz-version-id

For a more information about these response headers, see Amazon's official S3 POST Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPOST.html>). For a description of common response headers, see [Table 9, “Supported Response Headers,”](#) on page 29.

Response Elements

For a list of supported elements in the XML response for the POST Object operation, see Amazon's official S3 POST Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectPOST.html>).

Expected HTTP Error Response Codes

200 or 201 or 204

Error Response Code

The POST Object API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```

POST / HTTP/1.1
User-Agent: browser_data
Accept: file_types
Accept-Language: Regions
Accept-Encoding: encoding
Accept-Charset: character_set
Keep-Alive: 300
Connection: keep-alive
Content-Type: multipart/form-data; boundary=9431149156168
Content-Length: length

--9431149156168
Content-Disposition: form-data; name="key"

acl
--9431149156168
Content-Disposition: form-data; name="tagging"

<Tagging><TagSet><Tag><Key>Tag Name</Key><Value>Tag Value</Value></Tag></TagSet></
Tagging>
--9431149156168
Content-Disposition: form-data; name="success_action_redirect"

success_redirect
--9431149156168
Content-Disposition: form-data; name="Content-Type"

content_type
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-uuid"

uuid
--9431149156168
Content-Disposition: form-data; name="x-amz-meta-tag"

metadata
--9431149156168
Content-Disposition: form-data; name="AWSAccessKeyId"

```

```
access-key-id
--9431149156168
Content-Disposition: form-data; name="Policy"

encoded_policy
--9431149156168
Content-Disposition: form-data; name="Signature"

signature=
--9431149156168
Content-Disposition: form-data; name="file"; filename="MyFilename.jpg"
Content-Type: image/jpeg

file_content
--9431149156168
Content-Disposition: form-data; name="submit"

Upload to S3
--9431149156168--

response:
HTTP/1.1 100 Continue
HTTP/1.1 200 OK
x-amz-request-id: tx0A49CE4060975EAC
x-amz-version-id: null
Date: Wed, 01 Mar 2018 12:00:00 GMT
ETag: "828ef3fdfa96f00ad9f27c383fc9ac7f"
Content-Length: 0
Connection: close
Server: Apache
```

DELETE Object

If a null version of an object exists, the DELETE operation removes the null version of the object and inserts a delete marker. For more details, see the following:

- [“Syntax Example” on page 93](#)
- [“Versioning” on page 93](#)
- [“Request Parameters” on page 93](#)
- [“Request Parameters” on page 93](#)
- [“Request Elements” on page 93](#)
- [“Response Headers” on page 93](#)
- [“Response Elements” on page 94](#)

- “Expected HTTP Error Response Code” on page 94
- “Error Response Code” on page 94

Syntax Example

For a syntax example, see Amazon's official S3 DELETE Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectDELETE.html>).

Versioning

To remove a specific version, you must be the bucket owner and you must use the `versionId` subresource. Using this subresource permanently deletes the version. If the object deleted is a delete marker, S3 sets the response header, `x-amz-delete-marker`, to `true`.

Request Parameters

This implementation of the DELETE Object operation does not support the use of request parameters.

Request Headers

This implementation of the DELETE Object operation does not support the use of request headers.

Request Elements

This implementation of the DELETE Object operation does not support the use of request elements.

Response Headers

This implementation of the DELETE Object operation supports the use of the `x-amz-version-id` response header. For more details about this response header, see Amazon's official S3 DELETE Object API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectDELETE.html>).

Response Elements

This implementation of the DELETE Object operation does not return response elements.

Expected HTTP Error Response Code

204 No Content

Error Response Code

The DELETE Object operation does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

The following request deletes my-desk-image.jpg.

```
DELETE /my-desk-image.jpg
HTTP/1.1 204 NoContent
x-amz-request-id: tx0A49CE4060975EAC
Date: Wed, 12 Oct 2018 17:50:00 GMT
Content-Length: 0
Connection: close
Server: Apache
```

The following request deletes the specified version of the object, my-desk-image.jpg.

```
DELETE /my-third-image.jpg?versionId=00012
HTTP/1.1 204 NoContent
x-amz-request-id: tx0A49CE4060975EAC
x-amz-version-id: 00012
Date: Wed, 12 Oct 2018 17:50:00 GMT
Content-Length: 0
Connection: close
Server: Apache
```

If the object deleted is a delete marker, the following response example appears.

```
HTTP/1.1 204 NoContent
x-amz-request-id: tx0A49CE4060975EAC
x-amz-version-id: 0011
x-amz-delete-marker: true
```

Date: Wed, 12 Oct 2018 17:50:00 GMT
Content-Length: 0
Connection: close
Server: Apache

DELETE Object Tagging

The DELETE Object Tagging operation removes the entire tag set from the specified object. For more details, see the following:

- “Syntax Example” on page 95
- “Versioning” on page 95
- “Request Parameters” on page 95
- “Request Headers” on page 95
- “Request Elements” on page 96
- “Response Headers” on page 96
- “Response Elements” on page 96
- “Expected HTTP Response Code” on page 96
- “Error Response Code” on page 96

Syntax Example

For a syntax example, see Amazon's official S3 DELETE Object Tagging API documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/RESTObjectDELETetagging.html>).

Versioning

To delete tags of a specific object version, add the `versionId` query parameter in the request.

Request Parameters

The DELETE Object Tagging operation does not support the use of request parameters.

Request Headers

The DELETE Object Tagging operation does not support the use of request headers.

Request Elements

The DELETE Object Tagging operation does not support the use of request elements.

Response Headers

The DELETE Object Tagging operation uses only response headers that are common to most responses. For more information, see [Table 9, “Supported Response Headers,” on page 29](#).

Response Elements

The DELETE Object Tagging operation does not return response elements.

Expected HTTP Response Code

204 No Content

Error Response Code

The DELETE Object API does not return special errors. For general information about S3 errors and a list of error codes, see [“S3 Client Error Handling Reference” on page 97](#).

Example

```
DELETE exampleobject/?tagging
HTTP/1.1 204 No Content
Date: Wed, 25 Nov 2018 12:00:00 GMT
Connection: close
Server: Apache
```


S3 Client Error Handling Reference

This reference identifies errors supported by the Oracle ZFS Storage Appliance S3 Object API.

- [“Error Response Format” on page 97](#)
- [“S3 Client Error Codes” on page 97](#)

Error Response Format

When an error occurs, the header contains the following information:

- Content-Type: application/xml
- HTTP status code (4xx or 5xx)

Information about an error is also available in the body or response. The following example shows the structure of response elements that are common to all REST error responses.

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <Code>NoSuchKey</Code>
  <Message>The resource you requested does not exist</Message>
  <Resource>/mybucket/myfoto.jpg</Resource>
  <RequestId>4442587FB7D0A2F9</RequestId>
</Error>
```

For a further information about REST error response elements, see Amazon's official S3 REST API Error Response documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/ErrorResponses.html>).

S3 Client Error Codes

For a list of supported error codes, see Amazon's official S3 REST API Error Code documentation (<https://docs.aws.amazon.com/AmazonS3/latest/API/ErrorResponses.html>).

