Overview

StorageGRID

NetApp
July 18, 2022

This PDF was generated from https://docs.netapp.com/us-en/storagegrid-116/swift/history-of-swift-api-support-in-storagegrid.html on July 18, 2022. Always check docs.netapp.com for the latest.

Table of Contents

Use Swift: Overview 1
History of Swift API support in StorageGRID 1
How StorageGRID implements Swift REST API 2
Recommendations for implementing Swift REST API 3

Use Swift: Overview

Client applications can use the OpenStack Swift API to interface with the StorageGRID system.

StorageGRID supports the following specific versions of Swift and HTTP.

Item	Version
Swift specification	OpenStack Swift Object Storage API v1 as of November 2015
HTTP	1.1 For more information about HTTP, see HTTP/1.1 (RFCs 7230-35). **Note**: StorageGRID does not support HTTP/1.1 pipelining.

Related information

OpenStack: Object Storage API

History of Swift API support in StorageGRID

You should be aware of changes to the StorageGRID system's support for the Swift REST API.

Release	Comments
11.6	Minor editorial changes.
11.5	Removed Weak consistency control. The Available consistency level will be used instead.
11.4	Added support for TLS 1.3 and updated list of supported TLS cipher suites. CLB is deprecated. Added description of interrelationship between ILM and consistency setting.
11.3	Updated PUT Object operations to describe the impact of ILM rules that use synchronous placement at ingest (the Balanced and Strict options for Ingest Behavior). Added description of client connections that use load balancer endpoints or high availability groups. Updated list of supported TLS cipher suites. TLS 1.1 ciphers are no longer supported.
11.2	Minor editorial changes to document.

Release	Comments
11.1	Added support for using HTTP for Swift client connections to grid nodes. Updated the definitions of consistency controls.
11.0	Added support for 1,000 containers for each tenant account.
10.3	Administrative updates and corrections to the document. Removed sections for configuring custom server certificates.
10.2	Initial support of the Swift API by the StorageGRID system. The currently supported version is OpenStack Swift Object Storage API v1.

How StorageGRID implements Swift REST API

A client application can use Swift REST API calls to connect to Storage Nodes and Gateway Nodes to create containers and to store and retrieve objects. This enables service-oriented applications developed for OpenStack Swift to connect with on-premise object storage provided by the StorageGRID system.

Swift object management

After Swift objects have been ingested in the StorageGRID system, they are managed by the information lifecycle management (ILM) rules in the system's active ILM policy. The ILM rules and policy determine how StorageGRID creates and distributes copies of object data and how it manages those copies over time. For example, an ILM rule might apply to objects in specific Swift containers and might specify that multiple object copies be saved to several data centers for a certain number of years.

Contact your StorageGRID administrator if you need to understand how the grid's ILM rules and policies will affect the objects in your Swift tenant account.

Conflicting client requests

Conflicting client requests, such as two clients writing to the same key, are resolved on a "latest-wins" basis. The timing for the "latest-wins" evaluation is based on when the StorageGRID system completes a given request, and not on when Swift clients begin an operation.

Consistency guarantees and controls

By default, StorageGRID provides read-after-write consistency for newly created objects and eventual consistency for object updates and HEAD operations. Any GET following a successfully completed PUT will be able to read the newly written data. Overwrites of existing objects, metadata updates, and deletes are eventually consistent. Overwrites generally take seconds or minutes to propagate, but can take up to 15 days.

StorageGRID also allows you to control consistency on a per container basis. You can change the consistency control to make a trade-off between the availability of the objects and the consistency of those objects across

different Storage Nodes and sites, as required by your application.

Related information

Manage objects with ILM

GET container consistency request

PUT container consistency request

Recommendations for implementing Swift REST API

You should follow these recommendations when implementing the Swift REST API for use with StorageGRID.

Recommendations for HEADs to non-existent objects

If your application routinely checks to see if an object exists at a path where you do not expect the object to actually exist, you should use the “Available” consistency control. For example, you should use the “Available” consistency control if your application performs a HEAD operation to a location before performing a PUT operation to that location.

Otherwise, if the HEAD operation does not find the object, you might receive a high number of 500 Internal Server errors if one or more Storage Nodes are unavailable.

You can set the “Available” consistency control for each container using the PUT container consistency request.

Recommendations for object names

For containers that are created in StorageGRID 11.4 or later, restricting object names to meet performance best practices is no longer required. For example, you can now use random values for the first four characters of object names.

For containers that were created in releases earlier than StorageGRID 11.4, continue to follow these recommendations for object names:

- You should not use random values as the first four characters of object names. This is in contrast to the former AWS recommendation for name prefixes. Instead, you should use non-random, non-unique prefixes, such as `image`.
- If you do follow the former AWS recommendation to use random and unique characters in name prefixes, you should prefix the object names with a directory name. That is, use this format:

  ```
  mycontainer/mydir/f8e3-image3132.jpg
  ```

  Instead of this format:

  ```
  mycontainer/f8e3-image3132.jpg
  ```

Recommendations for “range reads”

If the **Compress Stored Objects** option is selected (**CONFIGURATION** > **System** > **Grid options**), Swift client applications should avoid performing GET object operations that specify a range of bytes be returned. These “range read” operations are inefficient because StorageGRID must effectively uncompress the objects to access the requested bytes. GET Object operations that request a small range of bytes from a very large object are especially inefficient; for example, it is very inefficient to read a 10 MB range from a 50 GB compressed object.

If ranges are read from compressed objects, client requests can time out.

> If you need to compress objects and your client application must use range reads, increase the read timeout for the application.

Related information

GET container consistency request

PUT container consistency request

Administer StorageGRID

Copyright Information

Copyright © 2022 NetApp, Inc. All rights reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means-graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system- without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

Trademark Information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.