



Technical Report

StorageGRID Object Storage Load Balancer Options

Brian Atkins, Florian Feldhaus, Steve Pruchniewski, Yahshanulla Syedshaw, Steve Waltner,
NetApp
September 2018 | TR-4626 v1.2

Abstract

This document helps you determine whether to use third-party load balancers or the provided API Gateway Node. It also provides configuration recommendations.

TABLE OF CONTENTS

1	Introduction	3
2	Load Balancing Concepts	3
3	HTTP Support in StorageGRID 11.1	3
4	Do You Need a Load Balancer?	4
5	StorageGRID API Gateway Nodes	4
5.1	API Gateway Node Best Practices.....	5
6	Third-Party Load Balancers	6
6.1	General Best Practices for Third-Party Load Balancers.....	6
6.2	F5 BIG-IP Local Traffic Manager Health Check Monitor	9
6.3	Brocade Virtual Traffic Manager Health Check Monitor	11
6.4	Citrix Netscaler.....	13
6.5	HAProxy.....	14
	Where to Find Additional Information	14
	Version History	14

LIST OF TABLES

Table 1)	Determining if a load balancer is required.	4
Table 2)	API gateway versus third-party load balancer.	4

LIST OF FIGURES

Figure 1)	Load balancing.....	3
-----------	---------------------	---

1 Introduction

A NetApp® StorageGRID® Object Storage deployment consists of multiple Storage Nodes and often is deployed across many sites. The Storage Nodes provide a service endpoint for applications, and they manage storage, replication, erasure-coding and metadata. A load balancer directs clients to an optimal Storage Node at an optimal site seamlessly, so that the failure of nodes or even an entire site is transparent.

StorageGRID includes a basic load balancer called the API Gateway Node at no extra cost. Some applications and use cases require features and customization beyond this load balancer's capabilities. In these cases, you can choose a third-party load balancer, either commercial or open source.

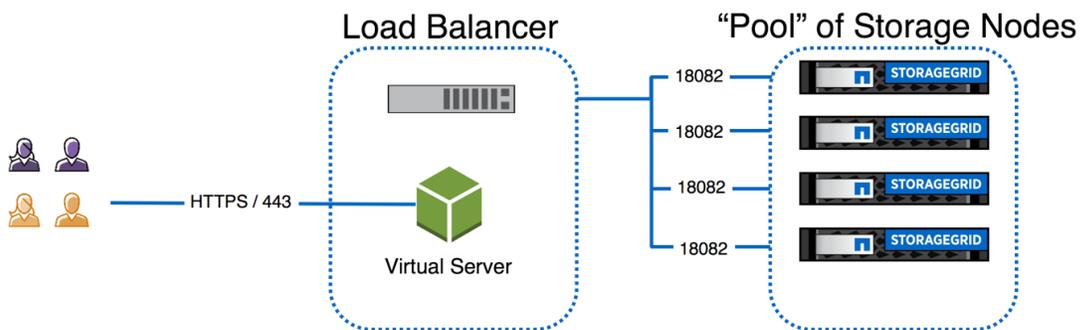
The goals of this document are to:

- Help you determine if a load balancer is required
- Help you decide if the API Gateway Node can meet requirements or if you need a third-party load balancer
- Make recommendations for load-balancer deployments and configuration

2 Load Balancing Concepts

While there are many load balancers to choose from they essentially all create pool of resources and expose them to end users via a virtual server as shown in this diagram. Here we see StorageGRID Storage Nodes serving the S3 API on port 18082. The load balancer then creates a Virtual Server and presents the S3 API on the standard HTTPS port 443.

Figure 1) Load balancing.



3 HTTP Support in StorageGRID 11.1

With the release of StorageGRID 11.1 HTTP non-encrypted client traffic is supported. This provides customers with the option to allow HTTP traffic if encryption is not required or to terminate SSL on a 3rd party load balancer and allow HTTP from the load balancer to the storage nodes which may provide a performance benefit.

All inter-grid and intra-site communication remains TLS encrypted.

4 Do You Need a Load Balancer?

Most StorageGRID deployments require a load balancer. There are, however, applications that can be configured to point at multiple Storage Nodes and archive workloads that can tolerate service disruptions. The following table helps you decide if a load balancer is required.

Table 1) Determining if a load balancer is required.

Application Requirement	Load Balancer Required?
A single service endpoint	Yes. You must abstract many nodes and even sites behind a single service endpoint.
Multiple service endpoints	No. Some applications can point to multiple service endpoints (Quantum StorNext, for example). These applications manage their own connections.
Multisite deployment	Yes. To provide a single namespace across multiple sites, a load balancer is required.
Active workload	Yes. Applications that demand constant connection to the service endpoint require a load balancer.
Archive workload	Optional. Some archive workloads can tolerate service disruptions. Depending on business requirements, you might still require a load balancer, especially if the throughput requirements are higher than the throughput of a single node.

5 StorageGRID API Gateway Nodes

StorageGRID includes an optional load balancer called the API Gateway Node. It can be deployed as a VMware VM or containerized node. It's included at no extra cost, and you can choose to deploy as many instances as needed. The API gateway understands the health of the grid and the utilization of each Storage Node by performing an ADC query —there is no need (or ability) to configure a health check. You simply deploy the API Gateway Node as part of a grid deployment.

Although the API Gateway Node is low cost and requires no configuration, it is not as robust as other load balancer options. The following table helps you determine when the API Gateway Node is appropriate for your deployment and when you need a third-party load balancer.

Table 2) API gateway versus third-party load balancer.

Requirement	API Gateway or Third-Party Load Balancer
High availability (HA)	Third-party or API gateway + DNS round-robin. The API Gateway Node is not HA by default. To provide an HA solution, choose a third-party load balancer, or pair the API gateway with DNS round-robin or similar technology.
Multisite with "fast" failover	Third-party. The API Gateway Node takes up to three minutes to fail over to a remote site. Third-party products might also allow you to set your own conditions for triggering failover. The API gateway fails over to a remote site only when all Storage Nodes in the local site are down.

Requirement	API Gateway or Third-Party Load Balancer
Site locality	Third-party. In a global multi-site setup it is expected that clients connect via a global endpoint, but traffic should go to the local site unless the local site is unavailable. This can be achieved with Third-party load balancers which support DNS and/or Anycast based load balancing.
Archive workload	API gateway. The API gateway is well suited for archive workloads such as NetApp AltaVault™.
Customization	Third-party. Third-party load balancers allow customization such as configurable health checks.
Untrusted networks	Third-party. Third-party load balancers, especially commercial ones, are built to be exposed to untrusted networks (internet) and handle security challenges such as denial-of-service attacks.
High performance	Third-party. Customers with demanding workloads typically choose commercial options.
Simplicity	API gateway. No configuration required; no need for additional applications.

5.1 API Gateway Node Best Practices

This section provides best practices based on customer experience. As noted earlier, the API gateway requires no configuration. The options are simply how many to deploy.

- Quantity of nodes required:
 - Minimum of one per site, two if HA required
 - General guidance is two per site and add as needed to support connections
- High availability:
 - Implement VMware HA for Gateway node
 - DNS round-robin, Anycast, or similar solution
- When to add more API Gateway nodes:
 - Bandwidth constrained: If you are not getting the expected bandwidth for the number of Storage Nodes expected, adding API Gateway Nodes should help.
- Based on number of connections (trend over time):
 - API Gateway Node > CLB > HTTP > Incoming Sessions - Established (CCES).
 - Each Gateway Node can serve 20K sessions.
- Open file descriptors (trend over time)
 - API Gateway Node > CLB > Resources > Open File Descriptors (FOPN).
 - Grid alarm threshold is 32K
- Requirement for network separation:
 - API Gateway Nodes allow bridging between client network and grid network. If networks need to be further separated, one or more API Gateway Nodes may be used per security zone.

Port Redirection

Customers may choose to use standard ports for S3 and Swift API vs the default StorageGRID ports. Consult StorageGRID documentation for instructions on how to configure the Gateway node to use standard ports such as 443 and 80.

6 Third-Party Load Balancers

This section covers configuration options for third-party load balancers commonly used by StorageGRID customers. It is not an exhaustive list of load balancers that will work with StorageGRID; any HTTPS load balancer should be compatible.

For Multi-Site setups consider using global DNS load balancing solutions such as F5 Big-IP DNS.

6.1 General Best Practices for Third-Party Load Balancers

The following configurations should apply to any third-party load balancer. Specific guidance is provided for commonly used commercial load balancers from F5, Brocade, Citrix and for the Opensource load balancer HAProxy..

As stated previously the high level configuration involves creating a pool of Storage Nodes and presenting them via Virtual Server.

Ports

StorageGRID Storage Nodes present the S3 and Swift APIs on the following ports:

- S3 HTTPS: 18082
- S3 HTTP: 18084
- SWIFT HTTPS: 18083
- SWIFT HTTP: 18085

Most customer will choose to present the APIs on the Virtual Server via the standard HTTPS and HTTP ports (443 and 80).

Health Checks

Third-party load balancers require a method to determine the health of each node and its eligibility to receive traffic. NetApp recommends the HTTP `OPTIONS` method to perform the health check. The load balancer issues HTTP `OPTIONS` requests to each individual Storage Node and expects a 200 status response.

If any Storage Node does not provide a 200 response, the node is not able to service storage requests. Your application and business requirements should determine the timeout for these checks and the action your load balancer takes.

For example, if three of four Storage Nodes in data center 1 are down, you might direct all traffic to data center 2.

S3 Health Check Example

In this example, we are sending `OPTIONS` and checking for `200 OK`. We need to use `OPTIONS` because Amazon Simple Storage Service (S3) does not support unauthorized requests.

```
curl -X OPTIONS https://10.63.174.75:18082 --verbose --insecure
* Rebuilt URL to: https://10.63.174.75:18082/
* Trying 10.63.174.75...
* TCP_NODELAY set
* Connected to 10.63.174.75 (10.63.174.75) port 18082 (#0)
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
* Server certificate: webscale.stl.netapp.com
* Server certificate: NetApp Corp Issuing CA 1
* Server certificate: NetApp Corp Root CA
> OPTIONS / HTTP/1.1
> Host: 10.63.174.75:18082
> User-Agent: curl/7.51.0
> Accept: */*
```

```
>  
< HTTP/1.1 200 OK  
< Date: Mon, 22 May 2017 15:17:30 GMT  
< Connection: KEEP-ALIVE  
< Server: StorageGRID/10.4.0  
< x-amz-request-id: 3023514741
```

Swift Health Check Example

In this example, we are sending GET and checking for 200 OK.

```
curl https://10.63.174.75:18083/info --verbose --insecure  
* Trying 10.63.174.75...  
* TCP_NODELAY set  
* Connected to 10.63.174.75 (10.63.174.75) port 18083 (#0)  
* TLS 1.2 connection using TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384  
* Server certificate: 12665090  
* Server certificate: GPT  
> GET /info HTTP/1.1  
> Host: 10.63.174.75:18083  
> User-Agent: curl/7.51.0  
> Accept: */*  
>  
< HTTP/1.1 200 OK  
< Date: Tue, 20 Jun 2017 15:00:36 GMT  
< Connection: KEEP-ALIVE  
< X-Trans-Id: 528732654  
< Content-Length: 317  
< Content-Type: application/json; charset=UTF-8  
<  
* Curl_http_done: called premature == 0  
* Connection #0 to host 10.63.174.75 left intact  
{  
  "swift": {  
    "account_listing_limit": 1000, "container_listing_limit": 1000, "max_account_name_length": 256, "max_container_name_length": 255, "max_file_size": 5368709122, "max_header_size": 8192, "max_meta_count": 90, "max_meta_name_length": 128, "max_meta_overall_size": 4096, "max_meta_value_length": 256, "max_object_name_length": 255  
  }  
}
```

File- or Content-Based Health Checks

In general, file-based health checks are not recommended. Typically, a small file—`healthcheck.htm`, for example—is created in a bucket with a read-only policy; this file is then fetched and evaluated by the load balancer. This approach has several disadvantages:

- **Dependent on a single account.** If the account owning the file is disabled, the health check fails, and no storage requests are processed.
- **Data protection rules.** The default data protection scheme is a two-copy approach. In this scenario, if the two Storage Nodes hosting the health check file are unavailable, the health check fails, and storage requests are not sent to healthy Storage Nodes rendering the grid offline.
- **Audit log bloat.** As the load balancer fetches the file from every Storage Node every X minutes, many audit log entries are created.
- **Resource intensive.** Fetching the health check file from every node every few seconds consumes grid and network resources.

If a Content-Based Health Check is required, then a dedicated tenant should be used. Inside the tenant, a dedicated S3 bucket or Swift Container should be created. Then a S3 bucket or container ACL should be created which only allows PUT request and optionally restricts them to the IP of the Load balancer. Then the Health Check can be implemented by uploading an object via a PUT request.

Session Persistence

Session persistence, or stickiness, refers to the time a given HTTP session is allowed to persist. By default, sessions are dropped by Storage Nodes after 10 minutes. Longer persistence can lead to better

performance, because applications don't have to reestablish their sessions for every action; however, holding these sessions open consumes resources. You can reduce the session persistence on a third-party load balancer if you determine your workload would benefit.

For more information, refer to “Benefits of active, idle, and concurrent HTTP connections” on page 52 of [S3 \(Simple Storage Service\) Implementation Guide](#).

SSL Termination

There are security benefits to Secure Sockets Layer (SSL) termination on third-party load balancers. If the load balancer is compromised, the grid is compartmentalized. This keeps the attack surface off StorageGRID and is a significant benefit over the API Gateway Node.

The StorageGRID default configuration is HTTPS only, however in 11.1 you may also enable HTTP. If you choose to terminate SSL on the load balancer, the connection from the load balancer to the Storage Nodes is still encrypted unless you choose to enable HTTP on the grid. SSL termination has security benefits for deployments on untrusted networks and enabling HTTP with SSL termination on the load balancer may provide the performance benefit of SSL offload. At this time NetApp engineering has not conducted performance tests for this configuration.

There are three supported configurations:

- **SSL pass-through.** The SSL certificate is installed on StorageGRID as a custom server certificate.
- **SSL termination and re-encryption.** This might be beneficial if you are already doing SSL certificate management on the load balancer rather than installing the SSL certificate on StorageGRID. This configuration provides the additional security benefit of limiting the attack surface to the load balancer.
- **SSL termination with HTTP.** In this configuration SSL is terminated on the 3rd party load balancer and communication from the load balancer to StorageGRID is non-encrypted to take advantage of SSL off load.

Most S3 and Swift clients do not support Loadbalancing using redirects as the destination hostname is part of the authentication and a client would have to generate new authentication information for the redirected hostname.

Source/Client IP Visibility

If the client source IP address is required for audit logging, configure your load balancer such that it passes the requests through with the original requesting IP address.

Enable the Load Balancer to insert X-Forwarded-For for each request and then configure Audit to log the X-Forwarded-For header (Configuration > Audit > Header Name 1 = “X-Forwarded-For”)

See “audit protocol header” in Administration Guide.

Remote Site Failover

Most load balancers have the concept of pools—groups of Storage Nodes that should Configure HAProxy to use `OPTIONS` and check for a 200 status response for the health check in `haproxy.cfg`. `equate` to StorageGRID sites. Configure your load balancer to fail over to a remote site when fewer than two Storage Nodes are responding to the health check for an individual site or pool. StorageGRID needs at least two healthy nodes per site.

Note: A minimum StorageGRID deployment is three Storage Nodes. In a three-node grid, you can lose a single node and function with two healthy nodes.

Load Balancing Strategies

Most load balancing solutions offer multiple load balancing strategies. The following are common strategies

- **Round Robin.** A universal fit but suffers with few nodes and large transfers clogging single nodes.
- **Least Connection.** A good fit for small and mixed object workloads resulting in an equal distribution of the connections to all nodes.

The choice of algorithm becomes less important with an increasing number of storage nodes to choose from.

Verifying Distribution of Connections

To verify your method is evenly distributing load across storage nodes check the established sessions on each node in a given site:

- **UI Method:** Storage Node > LDR > HTTP > Currently Established Incoming Sessions
- **Metrics API:** storagegrid_http_sessions_incoming_currently_established

6.2 F5 BIG-IP Local Traffic Manager Health Check Monitor

- S3 Example
- Type: https
- Configure interval and timeout as desired
- Send string: OPTIONS / HTTP/1.1\r\n\r\n
- \r\n are carriage returns; different versions of BIG-IP software require zero, one, or two sets of \r\n sequences.

Note: For details, see <https://support.f5.com/csp/article/K10655>.

- Receive String: HTTP/1.1 200 OK

Local Traffic » Monitors » **New Monitor...**

General Properties

Name	https_storagegrid
Description	
Type	HTTPS
Parent Monitor	https

Configuration: Basic

Interval	5 seconds
Timeout	16 seconds
Send String	OPTIONS / HTTP/1.1\r\n\r\n
Receive String	HTTP/1.1 200 OK
Receive Disable String	
Cipher List	DEFAULT:+SHA:+3DES:+kEDH
User Name	
Password	
Reverse	<input type="radio"/> Yes <input checked="" type="radio"/> No
Transparent	<input type="radio"/> Yes <input checked="" type="radio"/> No
Alias Address	* All Addresses
Alias Service Port	* All Ports
Adaptive	<input type="checkbox"/> Enabled

- Create Pool: create one pool for each port required.
 - Assign the health monitor from the previous step.
 - Select a load-balancing method.
 - Select service port: 18082 (for S3) or 18083 (Swift).
 - Add nodes.

Swift Example

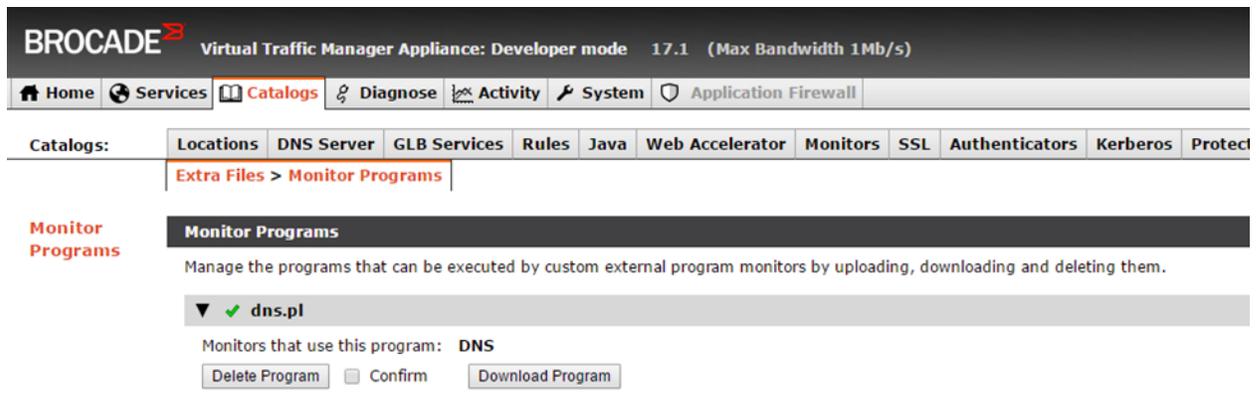
- Send String: `OPTIONS /info HTTP/1.1\r\nHost: \r\nConnection: Close\r\n`
- Receive String: `HTTP/1.1 204 No Content`

Send String	<code>OPTIONS /info HTTP/1.1\r\nHost: \r\nConnection: Close\r\n</code>
Receive String	<code>HTTP/1.1 204 No Content</code>

6.3 Brocade Virtual Traffic Manager Health Check Monitor

Brocade provides a Perl library and example scripts for creating custom monitors. NetApp recommends that you use one of the example scripts as a starting point for creating the StorageGRID custom monitor. The following example shows one way to configure a custom monitor using version 17.1 of Virtual Traffic Manager (vTM). Refer to current Brocade vTM documentation for details about the procedure or about later versions of the product.

Download the example script from Catalogs > Extra Files > Monitor Programs and modify.



Remove the statements in the script that are specific to DNS and replace them with the `OPTIONS` request and response check. For example, the following Perl statements use `curl` to issue the request and confirm that the response contains the string `200 OK`:

```
BEGIN { unshift @INC, "$ENV{ZEUSHOME}/zxtm/lib/perl",
        "$ENV{ZEUSHOME}/zxtmadmin/lib/perl"; }

use Zeus::ZXTM::Monitor qw( ParseArguments MonitorWorked MonitorFailed Log );

# Process the arguments
my %args = ParseArguments();

my $cmd = "curl --insecure -I -X OPTIONS https://$args{ipaddr}:$args{port}/";
Log( "Running $cmd" );
```

```

my $curl_out = qx($cmd);
Log( "Output:\n$curl_out" );

if (index($curl_out, '200 OK') == -1) {
    MonitorFailed( $curl_out );
}

Log( $curl_out );
MonitorWorked();

```

After you create the custom script, upload it using Catalogs > Extra Files > Monitor Programs > Upload Monitor Program. In this example, the new script is named `sn_options_monitor.pl`.

The screenshot shows the Brocade Virtual Traffic Manager Appliance interface in Developer mode, version 17.1. The navigation menu includes Home, Services, Catalogs, Diagnose, Activity, System, and Application Firewall. The Catalogs section is active, and the breadcrumb path is Extra Files > Monitor Programs. A green notification bar indicates that the configuration has been updated. The Monitor Programs section lists two programs: `dns.pl` (active) and `dns_port.pl` (inactive). Below this is the Upload Monitor Program section, which includes a file name input field containing `sn_options_monitor.pl` and an Upload Program button.

Next, navigate to Catalog > Monitors, scroll down, and access Create New Monitor. Provide a monitor name, choose External Program Monitor, and select the uploaded script from the section list. Under Scope, make sure Node: Monitor Each Node in the Pool Separately is selected, and then click Create Monitor. On the configuration page that appears, it is not necessary to change monitor parameters.

Create new monitor

Name:

The internal monitor implementation of this monitor.

type:

- Ping monitor
- TCP Connect monitor
- HTTP monitor
- TCP transaction monitor
- External program monitor ...
 -
- SIP monitor
- RTSP monitor

A monitor can either monitor each node in the pool separately and disable an individual node if it fails, or it can monitor a specific machine and disable the entire pool if that machine fails. GLB location monitors must monitor a specific machine.

scope:

- Node: Monitor each node in the pool separately
- Pool/GLB: Monitor a specified machine ...

You can select the new monitor when creating a pool or apply it to an existing pool.

BROCADE Virtual Traffic Manager Appliance: Developer mode 17.1 (Max Bandwidth 1Mb/s)

Home Services Catalogs Diagnose Activity System Application Firewall

Configuring: Traffic IP Groups Virtual Servers Pools > StorageGRID > Monitors Config Summary

Edit Monitors

✓ Your configuration has been updated.

Pool: StorageGRID (not used, 2 nodes)

Monitors

Monitors watch the nodes in a pool, and inform the traffic manager if the nodes are functioning correctly.

No monitors have been configured for this pool

Add monitor:

6.4 Citrix Netscaler

- Terminology:
 - Clients connect to Virtual Servers
 - Virtual Servers are serviced by application servers grouped together as Services
 - Monitors check health
- Notes:
 - Persistence can use SSL session IDs, HTTP cookies, source IP, and so on.
 - Http-ecv monitor might provide verb setting, otherwise script based custom monitor
 - SSL offload can reencrypt by using end-to-end encryption.
 - HTTP > HTTPS:
<http://docs.citrix.com/en-us/netscaler/11/traffic-management/ssl/frontend-http-backend-ssl.html>
 - HTTP-ECV configuration:
<https://support.citrix.com/article/CTX205406>
 - HTTP-ECV takes send string, expects receive string

6.5 HAProxy

Configure HAProxy to use `OPTIONS` and check for a 200 status response for the health check in `haproxy.cfg`. The bind port in the frontend could be changed to a different port such as 443.

The following is an example for SSL termination on the HAProxy:

```
frontend s3
    bind *:8082 crt /etc/ssl/server.pem ssl
    default_backend s3-servers
backend s3-servers
    balance leastconn
    option httpchk
    http-check expect status 200
    server dc1-s1 10.63.174.71:18082 ssl verify none check inter 3000
    server dc1-s2 10.63.174.72:18082 ssl verify none check inter 3000
    server dc1-s3 10.63.174.73:18082 ssl verify none check inter 3000
```

Example for SSL pass-through:

```
frontend s3
    mode tcp
    bind *:8082
    default_backend s3-servers
backend s3-servers
    balance leastconn
    option httpchk
    http-check expect status 200
    server dc1-s1 10.63.174.71:18082 check-ssl verify none inter 3000
    server dc1-s2 10.63.174.72:18082 check-ssl verify none inter 3000
    server dc1-s3 10.63.174.73:18082 check-ssl verify none inter 3000
```

Note: Full examples configuration for StorageGRID are documented on [GitHub](#).

Where to Find Additional Information

To learn more about the information that is described in this document, review the following documents and/or websites::

- F5 documentation
<https://support.f5.com/csp/article/K10655>
- StorageGRID Webscale 11.1 Documentation Center
<http://docs.netapp.com/sgws-111/index.jsp>

Version History

Version	Date	Document Version History
Version 1.0	August 2017	Initial release
Version 1.1	July 2018	Updated for StorageGRID 11.1
Version 1.2	September 2018	Added high-level diagram and port info

Refer to the [Interoperability Matrix Tool \(IMT\)](#) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

Copyright Information

Copyright © 2018 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.

Data contained herein pertains to a commercial item (as defined in FAR 2.101) and is proprietary to NetApp, Inc. The U.S. Government has a non-exclusive, non-transferrable, non-sublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b).

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.

TR-4626-1018