Technical Report

StorageGRID Load Balancer Options

Brian Atkins, Florian Feldhaus, Steve Pruchniewski, Yahshanulla Syedshaw, Steve Waltner, NetApp
March 2019 | TR-4626

Abstract

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

NetApp®

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

A NetApp® StorageGRID® deployment consists of multiple storage nodes and is often 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 seamlessly directs clients to an optimal storage ode at an optimal site, 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.

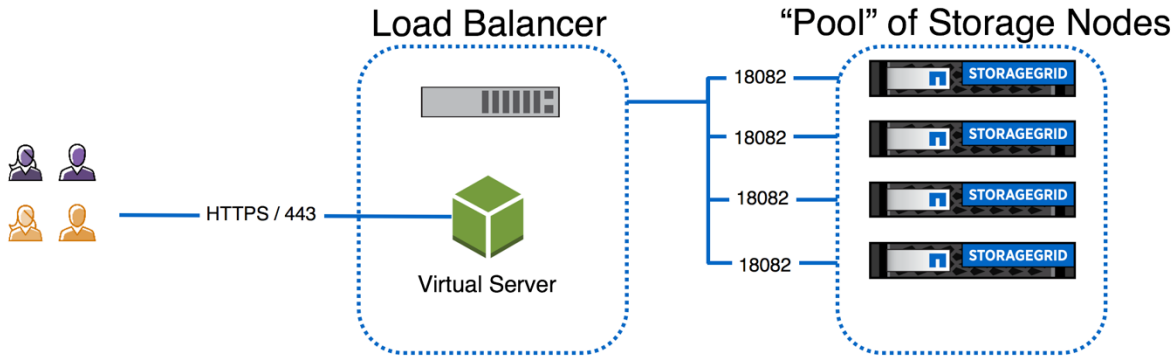
The goals of this document are to:

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

2 Load-Balancing Concepts

Although there are many load balancers to choose from, essentially, they all create a pool of resources and expose them to end users via a virtual server. Figure 1 shows 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 balancer concept diagram.



3 SSL Certificates

NetApp highly recommends configuring your StorageGRID instance with an SSL certificate from a trusted certificate authority (CA). A SSL certificate configuration is outside the scope of this document; however, this is an important subject for application integration. Many application issues are the result of using a nontrusted or misconfigured SSL certificate, which can lead to a poor customer experience.

If you cannot use an SSL certificate from a trusted CA, install a self-signed certificate. Self-signed certificates can be generated in several ways:

- Configuring StorageGRID Certificates for ONTAP Clients Using FabricPool

- NetApp StorageGRID SSL Certificate Configuration

4 HTTP Support in StorageGRID

With the release of StorageGRID 11.1, HTTP nonencrypted client traffic is supported. This support gives customers the option to allow HTTP traffic if encryption is not required, or to terminate SSL on a third-party load balancer and allow HTTP from the load balancer to the storage nodes, which may provide a performance benefit.

All intergrid and intrasite communication remain TLS encrypted.

5 Do You Need a Load Balancer?

Most StorageGRID deployments require a load balancer. However, some applications can be configured to point to multiple storage nodes and archive workloads that can tolerate service disruptions. Table 1 can help you decide whether a load balancer is required.

Table 1) Determining whether 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.

6 StorageGRID API Gateway Node

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 deploy as many instances as needed. The API Gateway Node understands the health of the grid and the use 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. Table 2 can help 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 3 minutes to failover to a remote site. Third-party products might 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.
Site locality	Third party. In a global multisite 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 that support DNS and/or Anycast based load balancing.
Archive workload	API gateway. The API gateway is well suited for archive or cold data workloads
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 to 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.

6.1 API Gateway Node Best Practices

This section describes best practices based on customer experience. As noted earlier, the API Gateway requires no configuration. The only option is simply how many to deploy.

- Number of nodes required:
  - Minimum of one per site, two if HA is 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 nodes:
  - Bandwidth constrained: If you aren’t 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 nodes can be used per security zone.

Port Redirection

Customers can choose to use standard ports for S3 and Swift API rather than the default StorageGRID ports. Consult the StorageGRID documentation for instructions on how to configure the API Gateway Node to use standard ports such as 443 and 80.

The procedure to enable port redirection for the Gateway node for VMware and bare metal installs is provided in StorageGRID documentation. These steps should be performed at installation of the node.

7 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 multisite setups, consider using a global DNS load-balancing solution such as F5 Big-IP DNS.

7.1 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, and Citrix and for the open source 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 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, that 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":2
56,"max_container_name_length":255,"max_file_size":5368709122,"max_header_size":8192,"max_meta_co
unt":90,"max_meta_name_length":128,"max_meta_overall_size":4096,"max_meta_value_length":256,"max_
object_name_length
```

File- or Content-Based Health Checks

In general, NetApp does not recommend file-based health checks. 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 that owns 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.** The load balancer fetches the file from every storage node every X minutes, creating many audit log entries.
- **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, a dedicated tenant should be used. Inside the tenant, a dedicated S3 bucket or Swift container should be created. Then an S3 bucket or container ACL should be created that only allows PUT requests 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. If you determine that your workload would benefit, you can reduce the session persistence on a third-party load balancer.

For more information, refer to "Benefits of active, idle, and concurrent HTTP connections" on page 52 of the 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, which is a significant benefit over the API Gateway Node.

The StorageGRID default configuration is HTTPS only; however, in 11.1 and later you can 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. As of this writing, 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 reencryption.** 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 third-party load balancer and communication from the load balancer to StorageGRID is nonencrypted to take advantage of SSL off-load.

Most S3 and Swift clients do not support load balancing by using redirects because 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 so 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 the StorageGRID Administration Guide.

Remote Site Failover

Most load balancers employ the concept of pools—groups of storage nodes that equate to StorageGRID sites. Configure HAProxy to use `OPTIONS` and check for a 200-status response for the health check in `haproxy.cfg`. 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 strategies for load balancing. 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 that your method is distributing load evenly 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`

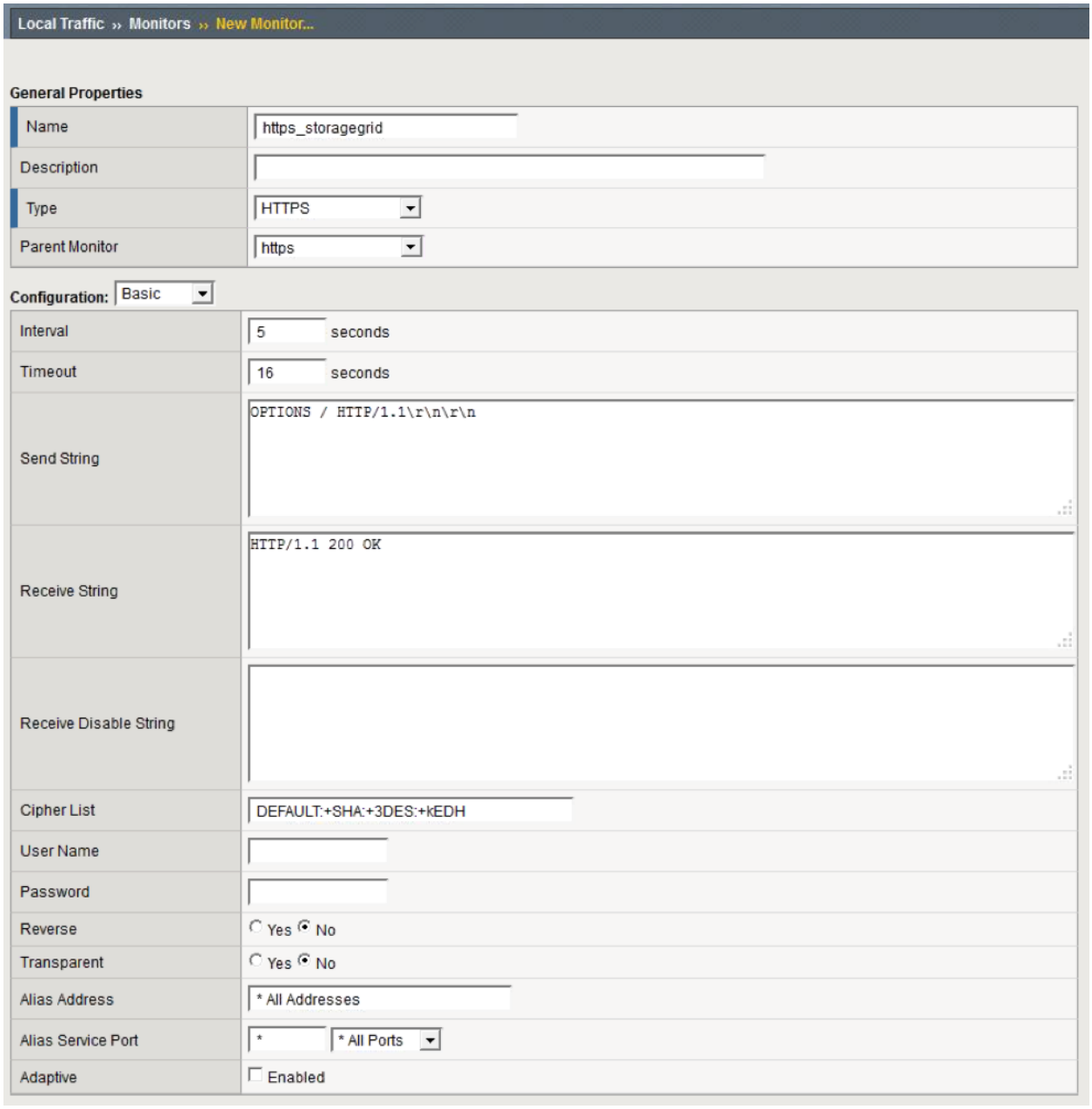
7.2 F5 BIG-IP Local Traffic Manager Health Check Monitor

The following examples illustrate configurations for F5 for S3 and Swift APIs.

S3 Example

1. In the Type field, enter HTTPS.
2. Configure the interval and timeout as desired.
3. Send String: `OPTIONS / HTTP/1.1\r\n\r\n`.
4. `\r\n` are carriage returns; different versions of BIG-IP software require zero, one, or two sets of `\r\n` sequences.

   Note: For more information, see CR/LF Characters Appended to the HTTP Monitor Send String.
5. Receive string: `HTTP/1.1 200 OK`.



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

Swift Example

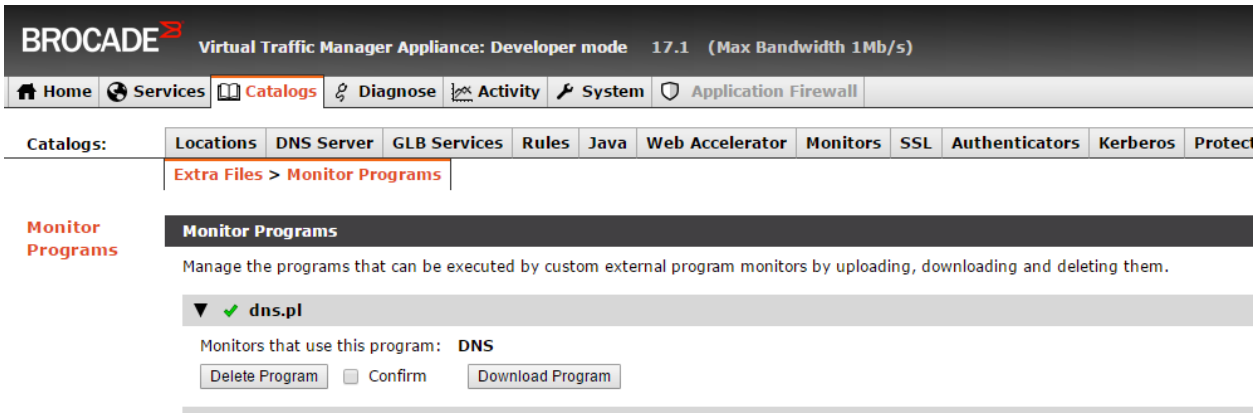
- Send String: `OPTIONS /info HTTP/1.1\r\nHost: \r\nConnection: Close\r\n`
- Receive String: `HTTP/1.1 204 No Content`

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

7.3 Brocade Virtual Traffic Manager Health Check Monitor

Brocade provides a Perl library and example scripts for creating custom monitors. NetApp recommends using 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 the 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 it as necessary.

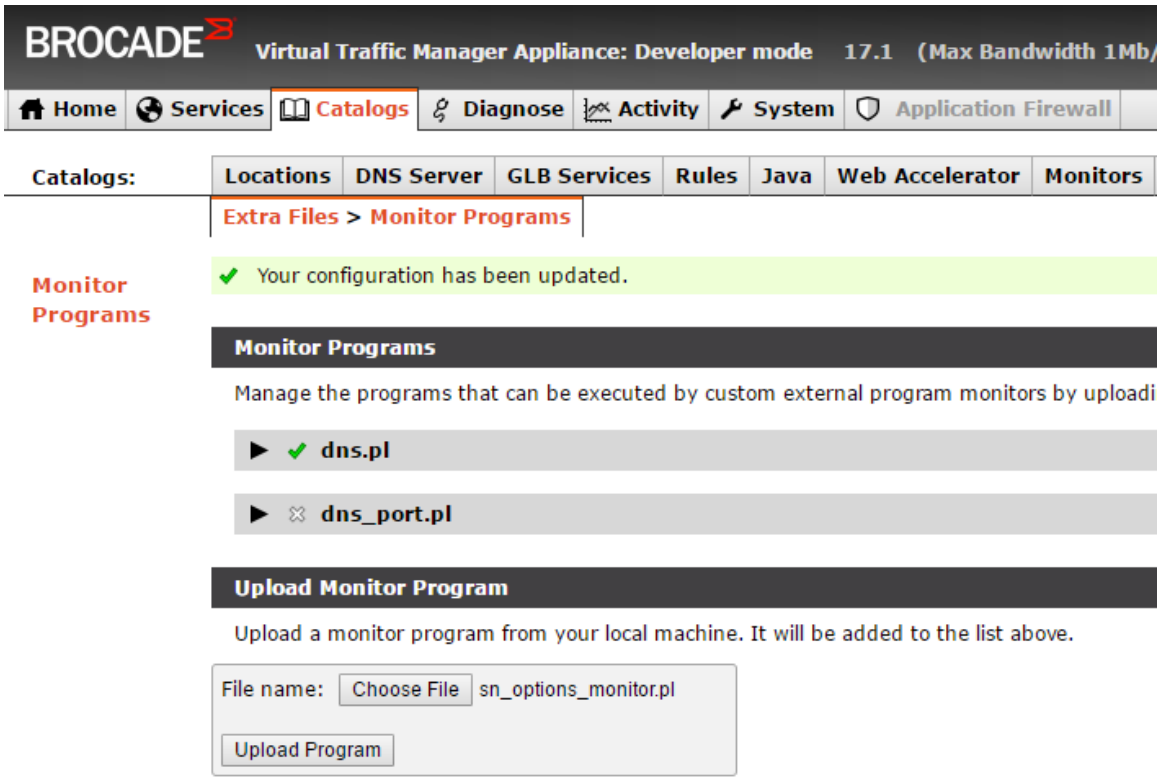


Remove the statements in the script that are specific to DNS and replace them with the `OPTIONS` request and response check. For example, the following Perl statements use `curl` to issue the request and confirm that the response contains the string `200 OK`:

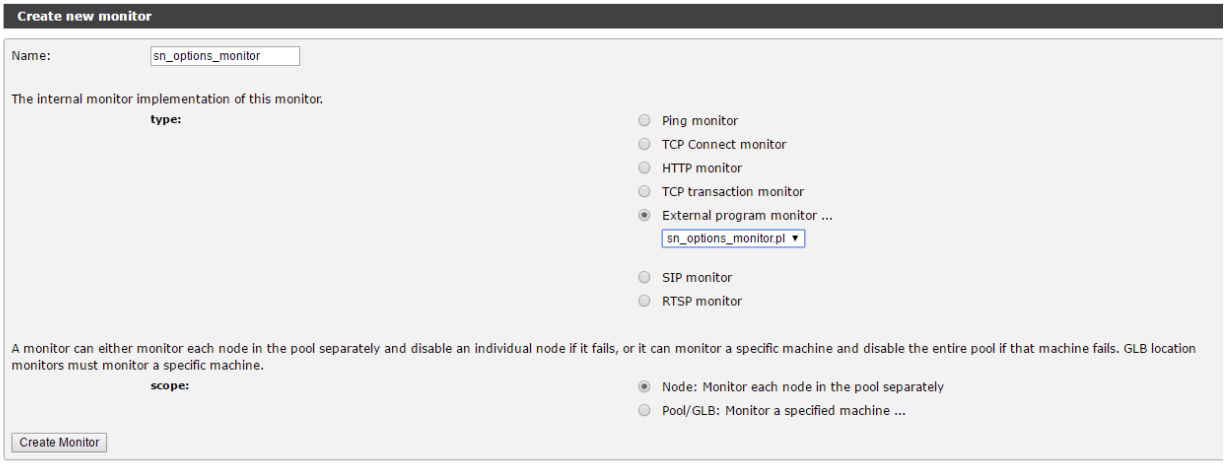
```
BEGIN { unshift @INC, "$ENV{ZEUSHOME}/zxtm/lib/perl",
                      "$ENV{ZEUSHOME}/zxtmadmin/lib/perl"; }

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();
```

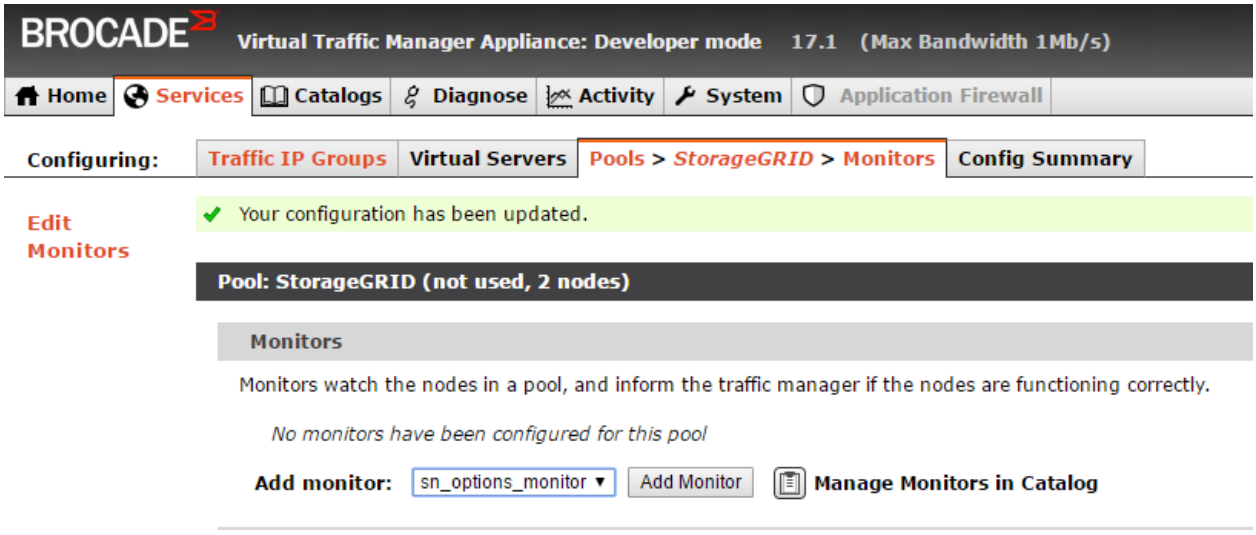
After you create the custom script, upload it using Catalogs > Extra Files > Monitor Programs > Upload Monitor Program. In this example, the new script is named `sn_options_monitor.pl`.



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



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



7.4 Citrix NetScaler

Several customers are currently running Citrix NetScaler with StorageGRID. At this time, we do not have detailed implementation steps, but plan to add them in the future.

Citrix NetScaler creates a virtual server for the storage endpoint. It refers to StorageGRID storage nodes as application servers, which are then grouped into services.

Use the HTTPS-ECV health check monitor to create a custom monitor to perform the recommended health check using OPTIONS and receiving 200. HTTP-ECV is configured with a send string and validates a receive string

For more information, see How to Use HTTPS–ECV Health Check Monitor on NetScaler (in the Citrix Knowledge Center).

7.5 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 front end could be changed to a different port, such as 443.

The following is an example for SSL termination on 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
```

For full examples of StorageGRID configurations, see Examples for HAProxy Configuration 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:

- NetApp StorageGRID Webscale 11.1 Documentation Center
  http://docs.netapp.com/sgws-111/index.jsp
- F5 AskF5 Documentation
  https://support.f5.com/csp/article/K10655

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	October 2018	Added high-level diagram and port information.
Version 1.3	March 2019	Updated for 11.2 release.

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.