



September 2015

53-1003965-01

Brocade Virtual Traffic Manager and Oracle GlassFish Server

Deployment Guide

© 2015 Brocade Communications Systems, Inc. All Rights Reserved.

ADX, Brocade, Brocade Assurance, the B-wing symbol, DCX, Fabric OS, HyperEdge, ICX, MLX, MyBrocade, OpenScript, The Effortless Network, VCS, VDX, Vplane, and Vyatta are registered trademarks, and Fabric Vision and vADX, vTM, vWAF, and SD are trademarks of Brocade Communications Systems, Inc., in the United States and/or in other countries. Other brands, products, or service names mentioned may be trademarks of others.

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

The product described by this document may contain open source software covered by the GNU General Public License or other open source license agreements. To find out which open source software is included in Brocade products, view the licensing terms applicable to the open source software, and obtain a copy of the programming source code, please visit <http://www.brocade.com/support/oscd>.

Contents

Preface.....	4
About This Guide.....	4
Audience	4
Contacting Brocade	4
Internet.....	4
Technical Support	4
Professional Services.....	4
Chapter 1: Solution Overview	5
Virtual Traffic Manager Overview	5
Performance.....	5
Reliability and scalability	5
Advanced scripting and application intelligence.....	5
Application acceleration	6
Application-layer security	6
Oracle GlassFish Application Server	6
Chapter 2: Oracle GlassFish Server Architecture	6
Chapter 3: Deploying Traffic Manager for Oracle GlassFish Application Server.....	7
Requirements	7
Configure vTM for Oracle GlassFish Server	7
Create Traffic IP Group.....	8
Create Pool	8
Create Virtual Server.....	8
SSL Decryption	9
Configure Traffic Script	9
Configure Session Persistence	10
Configure X-Forwarded-For	10
Configuration Summary	10
Chapter 4: Using Java Extensions	11
Chapter 5: Conclusion.....	11

Preface

Welcome to the Brocade Virtual Traffic Manager (vTM) and Oracle GlassFish Server Deployment Guide. Read this preface for an overview of the information provided in this guide and contact information. This preface includes the following sections:

- About This Guide
- Contacting Brocade

About This Guide

The Brocade Virtual Traffic Manager and Oracle GlassFish Server Deployment guide describes optimization of Oracle GlassFish Application Server farms.

Audience

This guide is written for network operations professionals, server administrators and DevOps professionals familiar with administering and managing Application Delivery Controllers (ADCs), Servers and Applications.

You must also be familiar with:

- Oracle GlassFish Application Server
- Brocade Virtual Traffic Manager

For more details on the Brocade vADC product family, see:

<http://www.brocade.com/vADC>

Contacting Brocade

This section describes how to contact departments within Brocade.

Internet

You can learn about Brocade products through the company Web site: <http://www.brocade.com>.

Technical Support

If you have problems installing, using, or replacing Brocade products, contact Brocade Support or your channel partner who provides support. To contact Brocade Support, see <http://www.brocade.com/en/support.html>.

Professional Services

Brocade Global Services has the expertise to help organizations build scalable, and efficient cloud infrastructures. Leveraging 15 years of expertise in storage, networking, and virtualization, Brocade Global Services delivers world-class professional services, technical support, and education services, enabling organizations to maximize their Brocade investments, accelerate new technology deployments, and optimize the performance of networking infrastructures.

Chapter 1: Solution Overview

This chapter includes the following sections:

- Oracle GlassFish Application Server
- Virtual Traffic Manager Overview
- Oracle GlassFish Application Server

Virtual Traffic Manager Overview

Brocade Virtual Traffic Manager (vTM) is a software-based application delivery controller (ADC) designed to deliver faster and more reliable access to public web sites and private applications. vTM frees applications from the constraints of legacy, proprietary, hardware-based load balancers, which enables them to run on any physical, virtual, or cloud environment. With vADC products from Brocade, organizations can:

- Make applications more reliable with local and global load balancing
- Scale application servers by up to 3x by offloading TCP and SSL connection overhead
- Accelerate applications by up to 4x by using web content optimization (WCO)
- Secure applications from the latest application attacks, including SQL injection, XSS, CSRF, and more
- Control applications effectively with built-in application intelligence and full-featured scripting engine

Virtual Traffic Manager offers much more than basic load balancing. It controls and optimizes end-user services by inspecting, transforming, prioritizing, and routing application traffic. The powerful TrafficScript® engine facilitates the implementation of traffic management policies that are unique to an application by allowing organizations to build custom functionality or to leverage existing features in Virtual Traffic Manager in a specialized way. With vTM, organizations can deliver:

Performance

Improve application performance for users by offloading encryption and compression from the web server by dynamic caching and reducing the number of TCP sessions on the application.

Reliability and scalability

Increase application reliability by load balancing traffic across web and application servers, balancing load across multiple data centers (private or public clouds), monitoring the response time of servers in real-time to decide the fastest way to deliver a service, protecting against traffic surges, and by managing the bandwidth and rate of requests used by different classes of traffic.

Advanced scripting and application intelligence

Manage application delivery more easily with fine-grained control of users and services using TrafficScript, an easy-to-use scripting language that can parse any user transaction, and take specific, real-time action based on user, application, request, or more. Development teams use TrafficScript to enable a point of control in distributed applications, while operations teams use it to quickly respond to changing business requirements or problems within an application before developers can fix it.

Application acceleration

Dramatically accelerate web-based applications and websites in real-time with optional web content optimization (WCO) functionality. It dynamically groups activities for fewer long distance round trips, resamples and sprites images to reduce bandwidth, and minifies JavaScript and combines style sheets to give the best possible response time for loading a web page on any browser or device.

Application-layer security

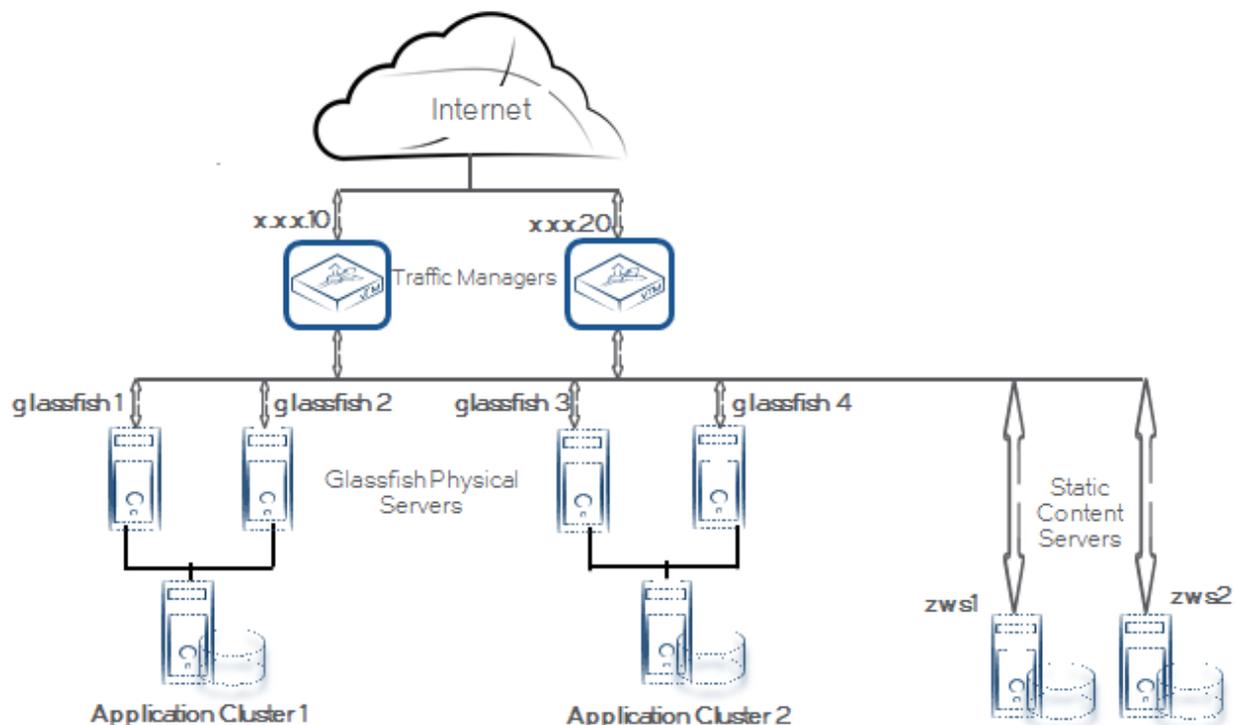
Enhance application security by filtering out errors in web requests, and protecting against external threats, with the option of a comprehensive Layer 7 firewall to defend against deliberate attacks.

Oracle GlassFish Application Server

GlassFish application server is a free, open-source, production-ready, supported application server based on Java EE 5. The server runs on many platforms including Windows, Linux, Solaris, AIX, and Mac OS.

Chapter 2: Oracle GlassFish Server Architecture

These deployment instructions will consider deploying the Traffic Manager and GlassFish application server in the following architecture:



GlassFish application server already supports load-balancing by utilizing a web-server and a plug-in. However, we will be using the Virtual Traffic Manager to take over this function, with advantages such as intelligent traffic manipulation and routing, high-speed SSL off-loading, content caching and all others associated with its use.

The Traffic Manager servers are shown operating in active-active mode, using a pair of Traffic IP addresses. The DNS for the sites hosted by this architecture would list two A records for each website domain name. Were Traffic Manager configured to operate in active-passive mode, only a single IP address would be required for each domain name. Of course, multiple domain names can share the same IP address.

For a test deployment, it would be sufficient to deploy a single Traffic Manager, reducing effort and requirements. Although two application server clusters have been shown here in order to demonstrate request routing, Virtual Traffic Manager can be deployed in front of a single cluster. Similarly, a separate web-server cluster for static content is not essential, although it is recommended in many deployments since it reduces the load on the application servers.

Chapter 3: Deploying Traffic Manager for Oracle GlassFish Application Server

This chapter describes the process for deploying Virtual Traffic Manager to optimize the Oracle Application Server installation. It includes the following sections:

- Requirements
- Configure vTM for Oracle GlassFish Server

Requirements

- Brocade Virtual Traffic Manager (10.1 or later)
- Oracle GlassFish Server

Note: This deployment guide was certified while the product was with Riverbed and for 9.x or earlier versions of the Traffic Manager.

Configure vTM for Oracle GlassFish Server

This section contains step by step instructions on configuring Traffic Manager for Oracle Glassfish Server suite:

Component	Procedure	Description
Virtual Traffic Manager (once)	Create Traffic IP Group for Oracle GlassFish Server	A single Traffic IP Group must be created For details, see "Create Traffic IP Group"
	Create Pool for the GlassFish server farms (once for each server farm)	A Pool needs to have a set of servers to load-balance. Enter the hostname or IP address of the node along with the TCP/UDP port For details, see "Create Pool"
	Create Virtual Server for the application servers	Create and associate the Virtual Server to the server pool. For details, see "Create Virtual Server"
	SSL decryption	Configure SSL Decryption to enable SSL offloads. For details, see "SSL Decryption"
	Configure Session Persistence	Configure SSL Decryption to enable SSL offloads. For details, see "Configure Session Persistence"

Component	Procedure	Description
	Configure and associate Traffic routing	Configure and associate Traffic script for routing traffic For details, see "Configure Traffic Script"
	Pass the Client IP address to the application servers	Configure and associate Traffic script for routing traffic For details, see "Configure X-Forwarded-For"

Create Traffic IP Group

A Traffic IP Group (also known as a Virtual IP) will need to be created on which the Virtual server will be listening on. To create a new Traffic IP Group:

1. Navigate to **Services->Traffic IP Groups** and scroll down to **Create a new Traffic IP Group**.
2. Fill in the fields as follows:
 - **Name:** A descriptive name for the application server
 - **IP Addresses:** An IP Address that is mapped to FQDN of the application.
3. Click **Create Traffic Group**.

Create Pool

A Pool has to be created for each application server farm as shown in the topology diagram. To create a new Pool:

1. Navigate to **Services->Pools** and scroll down to **Create a new Pool**.
2. Fill in the fields as follows:
 - **Pool Name:** A descriptive name for the pool ex. Glassfish1-servers
 - **Nodes:** hostname:38080 or ipaddress:38080
 - **Monitor:** Select "Full HTTP"

Repeat the above steps for another set of application servers on port 38080 (ex. Glassfish2-servers) and a static server farm (ex static-servers) on port 80.

Create Virtual Server

Create a Virtual server that will handle all the application Traffic. To create a new Virtual Server:

1. Navigate to **Services->Virtual Servers** and scroll down to **Create a new Virtual Server**.
2. Enter the following:
 - **Virtual Server Name:** A descriptive name for the Virtual Server
 - **Protocol:** HTTP
 - **Port:** 80
 - **Default Traffic Pool:** Select the static-servers pool created in the step above.
3. Click on **Create Virtual Server**.
4. In the next screen, under **Listening on**, select **Traffic IP Groups** and check the appropriate Traffic IP Group that was created earlier.

5. Set **Enabled**: to **Yes**.
6. Click on the **Update** button to apply changes.

SSL Decryption

In order to perform SSL decryption, the certificate and the private key used for the Virtual Server created in the previous step must be imported into the Traffic Manager.

1. Navigate to the **Catalogs->SSL->SSL Certificates** catalog.
2. Click on **Import Certificate** to import the appropriate certificate.

After importing the certificate, enable SSL decryption on the Virtual Server created:

1. Navigate to **Services->Virtual Servers** and select the virtual server that will be performing SSL decryption.
2. Scroll down and click on **SSL Decryption**.
3. Set **ssl_decrypt** to **Yes**.
4. Select the certificate imported in the previous step.
5. Scroll down to the bottom of the page and click **Update**.

Configure Traffic Script

In order to have the Oracle GlassFish Server route traffic to the appropriate server farm (based on path in the URL), configure the following TrafficScript rule.

```
#!/ TS Rule for redirecting HTTP requests based on URL
$debug = 0; // Change value to 1 if debug needed
$path = http.getPath();
if(string.startsWith($path, "/clusterjsp"))
{
    pool.use("GlassFish1-servers");
    if ($debug > 0) { log.info(" GlassFish Server Farm 1 Selected ");}
} else if(string.startsWith($path, "/.jsp"))
{
    pool.use("GlassFish2-servers");
    if ($debug > 0) { log.info(" GlassFish Server Farm 2 Selected ");}
}
```

Associate the TrafficScript to the virtual server:

1. Navigate to **Services --> Virtual Server**.
2. Click the virtual server that was created above.
3. Click on **Rules**.
4. Under **Request Rules**, select the rule that was created in the above step from the dropdown.
5. Click **Add Rule**.

Configure Session Persistence

GlassFish application server does support the sharing of session state between server instances, but this comes at a price in terms of performance. To ensure maximize performance, disable the session state sharing and rely on the Traffic Manager to ensure all requests in a session go to the same server instance. However, in this document, configure session persistence but leave GlassFish application server's state sharing in place, to demonstrate the advantages it offers with regard to fault tolerance.

1. Go to **Catalogs** -> **Persistence** and create a new class called "cluster-jsp-persistence".
2. Set this class to use the **J2EE JSESSIONID** method and the failure mode of **choose a new node to use**.
3. Click **Update** to finish.
4. Go to **Services** -> **Pools** -> **GlassFish1-servers-> Session Persistence**.
5. Select the class from the list and click **Update**.

Configure X-Forwarded-For

Since the Traffic Manager is a proxy server, GlassFish application server applications will typically see connections originating from the Traffic Manager's IP address, rather than the client. By default, Traffic Manager puts the true client's IP address in a header named "X-Cluster-Client-Ip" which is added to each request, so the load-balanced application could use this instead of the source IP address. It is also possible to use TrafficScript to put the client's IP address in a header with a different name, if necessary for compatibility reasons.

For example:

```
http.setHeader("X-Forwarded-For", request.getRemoteIp());
```

Servlet code can be modified to query the correct header, or the Servlet `req.getRemoteAddr()` function can be overridden by subclassing the `HttpServletRequest` object to an appropriate wrapper.

Configuration Summary

By accessing the **Services** → **Config Summary** on the webGUI a complete snapshot of all the configured services is provided. This is very useful table to glance through to get a good understanding of how the services are configured.

Configuration Summary			
Virtual Servers ▾	Rules	Pools	Nodes
▼ http *:80	glassfish-routing	glassfish-pool-1	glassfish1:38080 glassfish2:38080
		glassfish-pool-2	glassfish3:38080 glassfish4:38080
	Use default pool	static	zws1:80 zws2:80
▼ Unused			

Chapter 4: Using Java Extensions

Java Extensions are small snippets of Java code, written to an API derived from the Servlet API (although the Traffic Manager does not run Servlets directly, and GlassFish application server cannot run Traffic Manager Java Extensions). These Java Extensions can inspect request or response data in any protocol, process the traffic (rewriting it if necessary) and can control how traffic is routed and managed within the Traffic Manager. Just like TrafficScript, Java Extensions can control all of the Traffic Manager capabilities.

In a large cluster of machines with different capabilities (GlassFish application servers, static content servers, etc), it may not be possible to apply common functionality such as authentication or watermarking to all servers. In this case, the functionality can be implemented on the Traffic Manager application delivery platform, which acts as a single point of entry and exit for all traffic.

Chapter 5: Conclusion

This document briefly discusses how to configure Traffic Manager to load balance traffic to a farm of Oracle GlassFish Application servers. Traffic Manager is able to manage traffic in a wide variety of ways, to improve the performance, security, reliability and integrity. Please refer to the product documentation on the Brocade Community Forums (<http://community.brocade.com>) for examples of how Brocade Virtual Traffic Manager can be deployed to meet a range of service hosting problems.