September 2015

53-1003960-01

Brocade Virtual Traffic Manager and Oracle Application Server 10G

Deployment Guide

Contents

Preface

Welcome to the Brocade Virtual Traffic Manager (vTM) and Oracle Application Server 10G 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 Application Server 10G Deployment guide describes optimization of Oracle 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 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:

- Virtual Traffic Manager Overview
- Oracle 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 Application Server

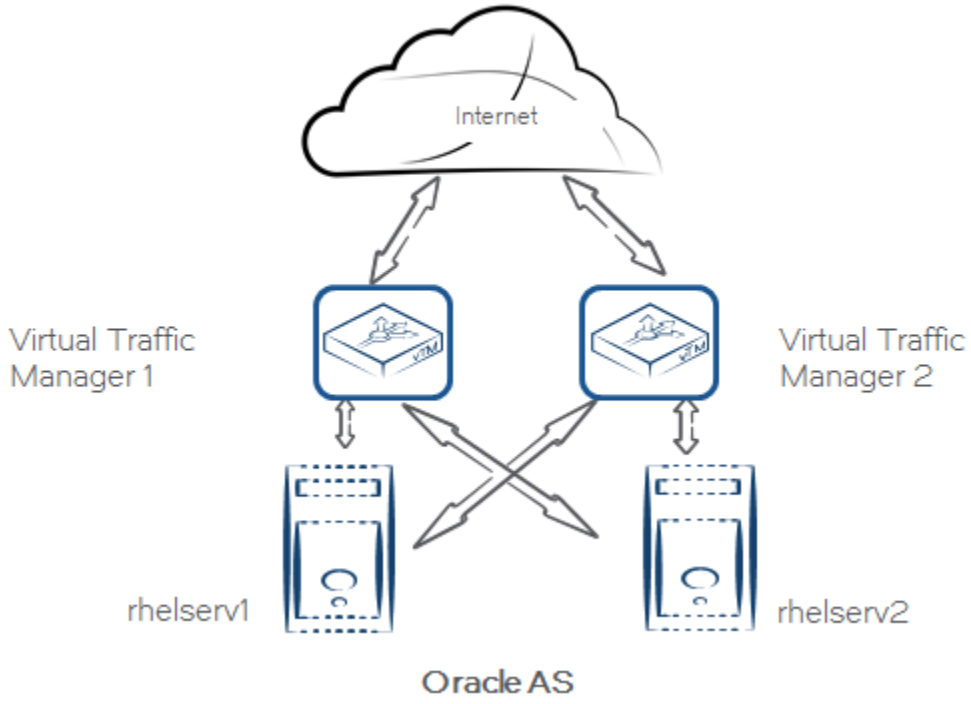
Oracle Application Server provides a single integrated packaged solution for middleware infrastructure including Oracle Containers for J2EE, Oracle Web Cache, Oracle HTTP Server, Oracle Forms, Oracle Reports, Oracle Portal and Oracle Discoverer. Also included in this infrastructure are integrated security, management and integration technologies. Oracle Application Server is a member of the Oracle Fusion Middleware family of products, which bring greater agility, better decision-making, and reduced cost and risk to diverse IT environments today.

Chapter 2: Oracle E-business Suite Architecture

Oracle clustering supports two high availability topologies. They are “Active – Active” and “Active – Passive”. This guide will address the more scalable configuration of the “Active – Active” topology.

In the “Active – Active” scenario all Oracle cluster members are load balanced by Virtual Traffic Manager and will therefore require session management with the vTM enabling Session persistence. The benefits of using the vTM this way can be in speed and efficiency, because the application server does not need to replicate any state information. However, a node failure will result in the loss of all sessions persisted to the node that failed.

In an “Active – Passive” scenario Stingray Traffic Manager will send all traffic to the active node and only fail over to the passive node when the active node fails. In this configuration, Oracle recommends to use some form of shared storage which is mounted on the active node with the need to remount the shared storage on to the other node when a failure occurs. Please read the Oracle Application Server High Availability guide for more information on “Active – Passive” topologies.



In the above topology, two vTMs are deployed in front of two Oracle AS servers installed on a supported Linux platform (Red Hat) with default install options and Oracle HTTP server listening on TCP port 7777.

Chapter 3: Deploying Traffic Manager for Oracle 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 Application Server

Requirements

- Brocade Virtual Traffic Manager (10.1 or later)
- Oracle Application Server (10G)

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 Application Server

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

Component	Procedure	Description
Virtual Traffic Manager (once)	Create Traffic IP Group for Oracle Application Server	A single Traffic IP Group must be created For details, see "Create Traffic IP Group"
	Create Pool for the business applications	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"
	Configure and associate Traffic script to pass client IP to Oracle servers	Configure and associate Traffic script for preserving client IP For details, see "Configure Traffic Script"

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 the application servers managed by the Traffic Manager. 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.
   - **Nodes:** hostname:7777 or ipaddress:7777
   - **Monitor:** Leave the default for now

3. In the next screen, click on **Load Balancing**.
4. Under **Algorithm**, select **Least Connections**.
5. Click on the **Update** button to apply changes.

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:** 443
   - **Default Traffic Pool:** Select the pool created in the previous step
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 Session Persistence

Oracle application server can manage session replication internally within the cluster. To make use of this, follow the J2EE specifications and refer to the instructions in the Oracle enterprise manager (ascontrol) when deploying an application. Alternatively, Virtual Traffic Manager can be used to ensure clients are always directed to the same server using Session persistence features.

The best method for doing this with Oracle AS is using a combination of the following persistence classes:

- Monitor Application Cookie
- URL rewriting

This combination is best suited for clients with and without cookies enabled.

Note: This will only work for browsers without cookies if you use are using the J2EE encodeURL() method from the HTTPServletResponse class to generate your URLs. This method will detect clients with cookies disabled and encode the session information inside the URL it generates. When you create a session the application server will set a JSESSIONID cookie which can be used by Stingray Traffic Manager to ensure that all requests with this session are sent back to the same node. If the client does not accept the cookie, encodeURL() will append the jsessionid to the URL and separate it from the real path by using a colon, e.g.
http://some.web.server/some/path;jsessionid=xxxxxxxxx

Monitoring Application Cookies

1. Go to **Services -> Pools -> your Oracle AS cluster** and click on **Session Persistence**.
2. Click on **Create New Session Persistence Class**, and create a class named **jsessionid_cookie**.
3. Set this class to **Monitor Application Cookies** and set the cookie name to **JSESSIONID**.
4. Leave the failure mode set to **choose a new node to use**. This will cause the Traffic Manager to send the request to a different node if the persistent node is not available.

URL Rewriting Persistence

Configuring URL Rewriting Persistence is a two stage process. First, a persistence class using "**Universal Session Persistence**" must be created, and then two TrafficScript rules must be return to detect a rewritten URL, extract the JSESSIONID from it and persist on this ID.

To create the session persistence class:

1. Go to **Catalogs -> Persistence** and create a new class called **url_rewriting**.
2. Set this class to use the **Universal Session Persistence** method and failure mode of **choose a new node to use**.
3. Click **Update** to finish. (Note that you should not associate the url_rewriting class with any particular pool - the TrafficScript rule below will associate it with a request as and when it is required.)
4. Go to **Services -> Virtual Servers -> your Oracle AS Cluster-> Rules**.
5. Click on **Manage Rules in Catalog** in the “Add New Request Rule” section.
6. Create a new TrafficScript rule called **url_rewriting_persistence**, and paste the following into the rule's text box, and click **Update**. Note that the argument to `connection.setPersistence` must match the name of the persistence class you created above.

```
$debug = 0; // Change value to 1 if debug needed
# Don't need to do this if we can persist on a cookie $cookie =
http.getCookie( "JSESSIONID" );
if( $cookie ) break;

$url = http.getpath();
if (string.regexmatch($url, ".*;JSESSIONID=([\\w.]*).*", "i")) {
    $sessionid = $1;
    connection.setPersistence( "url_rewriting" );
    connection.setPersistenceKey( $sessionid );
    if ($debug > 0) { log.info("Persistence values set");}
}
```

7. Finally, create a new response rule. Go to **Services -> Virtual Servers -> your Oracle AS Cluster -> Rules**.
8. Click on the **Manage Rules in Catalog** link in the "Add New Response Rule" section.
9. Create a new TrafficScript rule called **url_rewriting_response**, cut and paste the following into the rule's text box, and click **Update**.

```
# We're only interested in intercepting html responses
$debug = 0; // Change value to 1 if debug needed
$contenttype = http.getResponseHeader( "Content-Type" );
if( ! string.startsWith( $contenttype, "text/html" ) ) break;

# Don't need to do this if we can persist on a cookie $cookie =
http.getCookie( "JSESSIONID" );
if( $cookie ) break;

$body = http.getresponsebody();
if (string.regexmatch($body, ".*;JSESSIONID=([\\w.]*).*", "i")) {
    $sessionid = $1;
    connection.setPersistence( "url_rewriting" );
    connection.setPersistenceKey( $sessionid );
    if ($debug > 0) { log.info("Persistence values set");}
}
```

Configure Traffic Script

In order to have the Oracle application server log the real IP of the client and make that IP available to standard J2EE methods such as getRemoteAddress you will need to configure the OHS to retrieve the client IP address from a CLIENTIP host header. To do this you need to set the following directive in your httpd.conf:

```
UseWebCacheIP On
```

Once that is set you can use the following TrafficScript rule to add this header to all incoming connections.

```
#// TS Rule for setting CLIENTIP in the header
$debug = 0; // Change value to 1 if debug needed
# Set the remote address in the CLIENTIP header.
http.setHeader("CLIENTIP", request.getRemoteIP());
if ($debug > 0) { log.info("set CLIENTIP Header");}
```

Associate the TrafficScript to 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**.

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.

Chapter 4: Using and protecting Enterprise Manager

The main server in the Oracle HTTP Server runs the Oracle Enterprise Manager (EM) and any security conscious administrator will want to restrict who can access that service. Virtual Traffic Manager allows access to the enterprise manager while protecting from unauthorized users or, if preferred, deny access completely.

Denying access is simple; however, if restricted access to the console is needed, additional configuration is needed. The Enterprise Manager only runs on one of the cluster members so our service must ensure connections to that node always. The Enterprise Manager will also send redirects if the HTTP host header does not match the server name and hence, the correct host name must be used to connect to it.

Deny All Access to the EM console

Access to the ascontrol through the Virtual Traffic Manager can be denied by the following TrafficScript Request rule:

```
#// TS Rule for denying access to EM Console
$debug = 0; // Change value to 1 if debug needed
$path = http.getPath();
if ( string.startsWith($path,”/em/”) ) {
    connection.close("401 Denied\r\n");
    if ($debug > 0) { log.info("Access Denied");}
}
```

Allow restricted access to EM console

For restricted access to the EM console with a protection class configuration in the vTM, the following steps are needed:

1. Create a new service on port 7777 with one node (cluster manager) and with protocol http.
2. Create a new protection class called “Oracle Admin” and in Access restrictions add 0.0.0.0/0 to the banned list. Then add IP addresses you want to allow into the allowed list.
3. Redirect all requests to the path /em to the new virtual server running on port 7777 using the following Request Rule on the Virtual Server created in the earlier section.

```
#// TS Rule for redirecting requests to port 7777
$debug = 0; // Change value to 1 if debug needed
$path = http.getPath();
if ( string.startswith($path, "/em/" ) )
{
  $hostheader = http.getHostHeader();
  http.redirect("http://".$hostheader.":7777/em/");
  if ($debug > 0) { log.info("Redirected to port 7777");}
}
```

4. Rewrite incoming host header that matches the server name of the Oracle server using the following TrafficScript request rule.

```
#// TS Rule for adding additional headers
$debug = 0; // Change value to 1 if debug needed
# Set the host header to the name of the Oracle cluster
controller.
http.setHeader("Host","rhelserv1.techserv.cam.brocade.com");
if ($debug > 0) { log.info("Header rewritten");}
```

Further protection options

The Protection classes available in the Traffic Manager can use more than just IP addresses to make access decisions. TrafficScript rules can be used to decide access rights. For example, a TrafficScript can be written to only allow access if the Host header matches a certain string. Then, on the client, create a hosts file entry based on what is allowed in the Traffic Manager and ensure that the entry resolves to the virtual server IP address.

Chapter 5: Conclusion

This document briefly discusses how to configure Traffic Manager to load balance traffic to a farm of Oracle 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.