Oracle® Traffic Director

Administrator's Guide

11g Release 1 (11.1.1.9)

E21036-06

December 2016

ORACLE®

Oracle Traffic Director Administrator's Guide, 11g Release 1 (11.1.1.9)

E21036-06

Primary Author: Thrupthi N. T.

Contributors: Isvaran Krishnamurthy, Amit Gupta, Basant Kukreja, Julien Pierre, Hideaki Hayashi, Murthy Chintalapati, Savija Vijayaraghavan, Sandhya Prasad, Zhong Xu, Sriram Natarajan, Meena Vyas, Prem Kumar Venkatasalapathy, Prashant Sharma, Raghunandan Seshadri, Sarath Babu

Contents

8 Managing Virtual Servers

9 Managing TCP Proxies

10 Managing Listeners

11 Managing Security

12 Managing Logs

13 Monitoring Oracle Traffic Director Instances

14 Configuring Oracle Traffic Director for High Availability

15 Tuning Oracle Traffic Director for Performance

16 Diagnosing and Troubleshooting Problems

A Metrics Tracked by Oracle Traffic Director

B Web Application Firewall Examples and Use Cases

C Securing Oracle Traffic Director Deployment

List of Tables

List of Figures

Preface

This guide provides an overview of Oracle Traffic Director, and describes how to create, administer, monitor, and troubleshoot Oracle Traffic Director instances.

Audience

This guide is intended for users who are responsible for installing, configuring, administering, monitoring, and troubleshooting web-tier components such as web servers, reverse proxy servers, and load balancers.

It is assumed that readers of this guide are familiar with the following:

- Using web browsers
- Working in a terminal window
- Executing operating system commands on UNIX-like platforms

In addition, a basic understanding HTTP and SSL/TSL protocols is desirable, though not mandatory.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents, which are available on the Oracle Technology Network:

- *Oracle Traffic Director Release Notes*
- *Oracle Traffic Director Installation Guide*
- *Oracle Traffic Director Command-Line Reference*
- *Oracle Traffic Director Configuration Files Reference*

- *Oracle Virtual Assembly Builder User's Guide*
- *Oracle Exalogic Elastic Cloud Documentation*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Getting Started

Part I contains the following chapters:

- Chapter 1, "Getting Started with Oracle Traffic Director" provides an overview of Oracle Traffic Director and its features, explains the related terminology, describes the administrative framework of the product.
- Chapter 2, "Managing the Administration Server" describes how to create, start, access, and manage the Oracle Traffic Director administration server.
- Chapter 3, "Managing Administration Nodes" describes how to create, start, and manage administration nodes on which you can deploy Oracle Traffic Director instances.

1

Getting Started with Oracle Traffic Director

Oracle Traffic Director is a fast, reliable, and scalable layer-7 software load balancer. You can set up Oracle Traffic Director to serve as *the* reliable entry point for all HTTP, HTTPS and TCP traffic to application servers and web servers in the back end. Oracle Traffic Director distributes the requests that it receives from clients to servers in the back end based on the specified load-balancing method, routes the requests based on specified rules, caches frequently accessed data, prioritizes traffic, and controls the quality of service.

The architecture of Oracle Traffic Director enables it to handle large volumes of application traffic with low latency. The product is optimized for use in Oracle Exalogic Elastic Cloud and Oracle SuperCluster. It can communicate with servers in the back end over Exalogic's InfiniBand fabric. For more information about Exalogic, see the Oracle Exalogic Elastic Cloud documentation, http://docs.oracle.com/cd/E18476_01/index.htm. Oracle Traffic Director is also certified with various Fusion Middleware products.

Oracle Traffic Director is easy to install, configure, and use. It includes a simple, wizard-driven graphical interface as well as a robust command-line interface to help you administer Oracle Traffic Director instances.

On engineered systems platforms, you can set up pairs of Oracle Traffic Director instances and leverage its built-in High Availability capability to setup either Active-Passive or Active-Active failover. As the volume of traffic to your network grows, you can easily scale the environment by reconfiguring Oracle Traffic Director with additional back-end servers to which it can route requests.

Depending on the needs of your IT environment, you can configure Oracle Traffic Director to apply multiple, complex rules when distributing requests to the back-end servers and when forwarding responses to clients.

This chapter provides information to help you understand and get started with Oracle Traffic Director. It contains the following sections:

1.1 What's New in this Release?

The following are the new features in Oracle Traffic Director 11.1.1.9.0:

- Support for external health check executables

  Oracle Traffic Director now supports a generic health check hook-up mechanism, so that customers can write their own health check programs/scripts to monitor the health of specific origin servers. An external executable is especially useful for a protocol-level health check monitor for the origin servers.

  For more information, see Section 14.3.1, "Using an External Health-Check Executable to Check the Health of a Server" in Chapter 14, "Configuring Oracle Traffic Director for High Availability".

- Oracle Traffic Director on Oracle Linux Server 6

  Oracle Traffic Director can now be installed on Oracle Linux Server release 6. For more information, see *Oracle Traffic Director Installation Guide*.

- Support for custom maintenance page when origin servers offline

  Oracle Traffic Director now allows you to serve a custom server pool maintenance response code, and HTML page, when all the back-end servers are detected offline. Providing this type of message is better than having a gateway time-out, or creating other resources to host static content.

  For more information, see Chapter 6.6, "Configuring a Custom Maintenance Page" in Chapter 6, "Managing Origin-Server Pools".

- Support for TLS 1.1 and TLS 1.2

  Oracle Traffic Director now supports TLS 1.1 and TLS 1.2. This supports the WLS Managed Service use case, where Oracle Traffic Director provides the Load Balancer as a Service role, and acts as the primary SSL termination end point.

  For more information, see Chapter 11.2.4, "Configuring SSL/TLS Ciphers for a Listener" in Chapter 11, "Managing Security".

- HTTP forward proxy support in origin server pools

  Oracle Traffic Director now supports an HTTP forward proxy server to be optionally associated with an origin server pool so that all member origin servers of the pool are communicated with via the configured HTTP forward proxy server. This feature supports an environment where access to intended origin servers are restricted through corporate proxy servers.

  For more information, see Specifying an HTTP Forward Proxy Server in Chapter 6, "Managing Origin-Server Pools" and the `create-origin-server-pool` command in *Oracle Traffic Director CLI Reference*.

- High-availability (HA) changes and support for multiple admin users for JaaS

  Oracle Traffic Director includes several changes to high-availability (HA) functionality, including removal of a subnetwork restriction wherein the VIP as well as both the primary and the backup nodes have to belong to the same subnet; and, in the keep-alive or persistent connection handling subsystem added support for unicast transmission.

- Monitoring enhancements

  Oracle Traffic Director includes new monitored attributes for the instance, connection queue, virtual server, and origin server; changes to monitoring of the

TCP proxy for SNMP; and a new failover monitoring that contains the statistics of each VIP in the server instance.

For more information, see Appendix A, "Metrics Tracked by Oracle Traffic Director".

- Enabling POST keep-alive by default

  Oracle Traffic Director now enables POST keep-alive by default (a default of true for `always-use-keep-alive`) if the origin is Oracle WebLogic Server (WLS).

- Solaris IPoIB high availability (HA) support

  Oracle Traffic Director can now be configured to provide high availability for IP over InfiniBand (IPoIB) on Solaris. There is no longer a restriction that Oracle Traffic Director must be installed in a global zone to provide high availability for Solaris.

- Oracle Traffic Director now monitors the status of origin servers, and displays the status in the origin servers page in the user interface. Oracle Traffic Director can also show status for dynamically discovered origin servers if available and enabled.

1.2 Features of Oracle Traffic Director

Oracle Traffic Director provides the following features:

- **Advanced methods for load distribution**

  You can configure Oracle Traffic Director to distribute client requests to servers in the back end by using one of the following methods:

  - Round robin
  - Least connection count
  - Least response time
  - Weighted round robin
  - Weighted least connection count
  - IP hash (this distribution method sticks requests to origin servers based on a hash derived from the client IP address)

- **Flexible routing and load control on back-end servers**

  - **Request-based routing**

    Oracle Traffic Director can be configured to route HTTP/S requests to specific servers in the back end based on information in the request URI: pattern, query string, domain, source and destination IP addresses, and so on.

  - **Content-based routing**

    Oracle Traffic Director can be configured to route HTTP/S requests to specific servers in the back end based on contents within a request. This way, web service requests such as XML or JSON can be easily routed to specific origin servers based on specific elements within the body content. Content-based routing is enabled by default.

  - **Request rate acceleration**

    Administrators can configure the rate at which Oracle Traffic Director increases the load (number of requests) for specific servers in the back end. By

using this feature, administrators can allow a server that has just been added to the pool, or has restarted, to perform startup tasks such as loading data and allocating system resources.

  - **Connection limiting**

    Oracle Traffic Director can be configured to limit the number of concurrent connections to a server in the back end. When the configured connection limit for a server is reached, further requests that require new connections are not sent to that server.

- **Controlling the request load and quality of service**
  - **Request rate limiting**

    Oracle Traffic Director can be set up to limit the rate of incoming requests from specific clients and for specific types of requests. This feature enables administrators to optimize the utilization of the available bandwidth, guarantee a certain level of quality of service, and prevent denial-of-service (DoS) attacks.

  - **Quality of service tuning**

    To ensure equitable utilization of the available network resources for incoming requests, you can configure Oracle Traffic Director virtual servers to limit the maximum number of concurrent connections to clients and the maximum speed at which data can be transferred to clients.

- **Support for WebSocket connections**

  Oracle Traffic Director handles WebSocket connections by default. WebSocket connections are long-lived and allow support for live content, games in real-time, video chatting, and so on. In addition, Oracle Traffic Director can be configured to ensure that only those clients that strictly adhere to RFC 6455 are allowed. For more information, see the section Section 8.4, "Configuring Routes" and the *Oracle Traffic Director Command-Line Reference*.

- **Integration with Oracle Fusion Middleware**
  - Oracle Traffic Director is designed to recognize and handle headers that are part of requests to, and responses from, Oracle WebLogic Server managed servers in the back end.
  - When an Oracle Traffic Director instance is configured to distribute client requests to clustered Oracle WebLogic Server managed servers, Oracle Traffic Director automatically detects changes in the cluster—such as the removal or addition of managed servers, and considers such changes while routing requests.
  - Patches that Oracle delivers for the Oracle Traffic Director software can be applied by using OPatch, a Java-based utility, which is the standard method for applying patches to Oracle Fusion Middleware products.

- **Easy-to-use administration interfaces**

  Administrators can use either a graphical user interface or a command-line interface to administer Oracle Traffic Director instances.

- **Security**

  Oracle Traffic Director enables and enhances security for your IT infrastructure in the following ways:

  - **Reverse proxy**

By serving as an intermediary between clients outside the network and servers in the back end, Oracle Traffic Director masks the names of servers in the back end and provides a single point at which you can track access to critical data and applications hosted by multiple servers in the back end.

- **Support for SSL 3.0 and TLS 1.0, 1.1, and 1.2**

  To secure data during transmission and to ensure that only authorized users access the servers in the back end, you can configure SSL/TLS-enabled HTTP and TCP listeners for Oracle Traffic Director instances.

  You can either use digital certificates issued by commercial CAs such as VeriSign or generate RSA- and Elliptic Curve Cryptography (ECC)-type self-signed certificates with key sizes of up to 4096 bits by using the administration console or the CLI.

- **Web Application Firewall**

  A Web application firewall enables you to apply a set of rules to an HTTP request, which are useful for preventing common attacks such as Cross-site Scripting (XSS) and SQL Injection. The Web Application Firewall module for Oracle Traffic Director supports open source ModSecurity 2.6.

- **HTTP Forward Proxy Support in Origin Server Pools**

  In an environment where access to intended origin servers are restricted through corporate proxy servers, you can optionally associate an HTTP forward proxy server with an origin server pool so that all member origin servers (of said pool) are communicated with via the configured HTTP forward proxy server.

- **High availability**

  Oracle Traffic Director provides high availability for your enterprise applications and services through the following mechanisms:

  - **Health checks for the back end**

    If a server in the back end is no longer available or is fully loaded, Oracle Traffic Director detects this situation automatically through periodic health checks and stops sending client requests to that server. When the failed server becomes available again, Oracle Traffic Director detects this automatically and resumes sending requests to the server.

  - **Backup servers in the back end**

    When setting up server pools for an Oracle Traffic Director instance, you can designate a few servers in the back end as backup servers. Oracle Traffic Director sends requests to the backup servers only when none of the primary servers is available. This feature ensures continued availability even when some servers in the back end fail.

  - **Failover for load balancing**

    Two Oracle Traffic Director instances can be deployed in an active-passive or active-active configuration. If the primary Oracle Traffic Director instance fails, the backup instance takes over.

  - **Dynamic reconfiguration**

    Most configuration changes to Oracle Traffic Director instances can be deployed dynamically, without restarting the instances and without affecting requests that are being processed.

- **Monitoring statistics**

  Administrators can monitor a wide range of statistics pertaining to the performance of Oracle Traffic Director instances through several methods: the administration console, the command-line interface, and a report in XML format.

- **High performance**
  - **SSL/TLS offloading**

    Oracle Traffic Director can be configured as the SSL/TLS termination point for HTTP/S and TCP requests. This reduces the processing of overhead on the servers in the back end.

  - **Content caching**

    Oracle Traffic Director can be configured to cache (in its process memory) content that it receives from origin servers. By caching content, Oracle Traffic Director helps reduce the load on servers in the back end and helps improve performance for clients.

  - **HTTP compression**

    Administrators can configure Oracle Traffic Director instances to compress the data received from servers in the back end and forward the compressed content to the requesting clients. This feature improves the response time for clients connected on slow connections.

- **Virtualization-enabled solution**

  Oracle Traffic Director can be deployed as a virtual appliance on cloud and virtual platforms.

  After deploying Oracle Traffic Director as a physical application, you can create a virtual appliance from an Oracle Traffic Director instance or create an assembly containing multiple such appliances. You can then deploy the appliance or assembly on the Oracle Virtual Machine hypervisor. To enable such a deployment, Oracle provides an Oracle Traffic Director plug-in as part of Oracle Virtual Assembly Builder, a tool that you can use to build virtual appliances and assemblies from physical applications.

  For more information about creating and deploying virtual assemblies containing Oracle Traffic Director instances, see the *Oracle Virtual Assembly Builder User's Guide*.

- **TCP load balancing**

  With TCP load balancing, Oracle Traffic Director accepts client connections and routes the requests to a pool of servers running TCP-based protocols.

1.3 Typical Network Topology

The network topology that you create for Oracle Traffic Director varies depending on your business requirements such as the number of back-end applications for which you want to use Oracle Traffic Director to balance requests, IT requirements such as security, and the features of Oracle Traffic Director that you want to use.

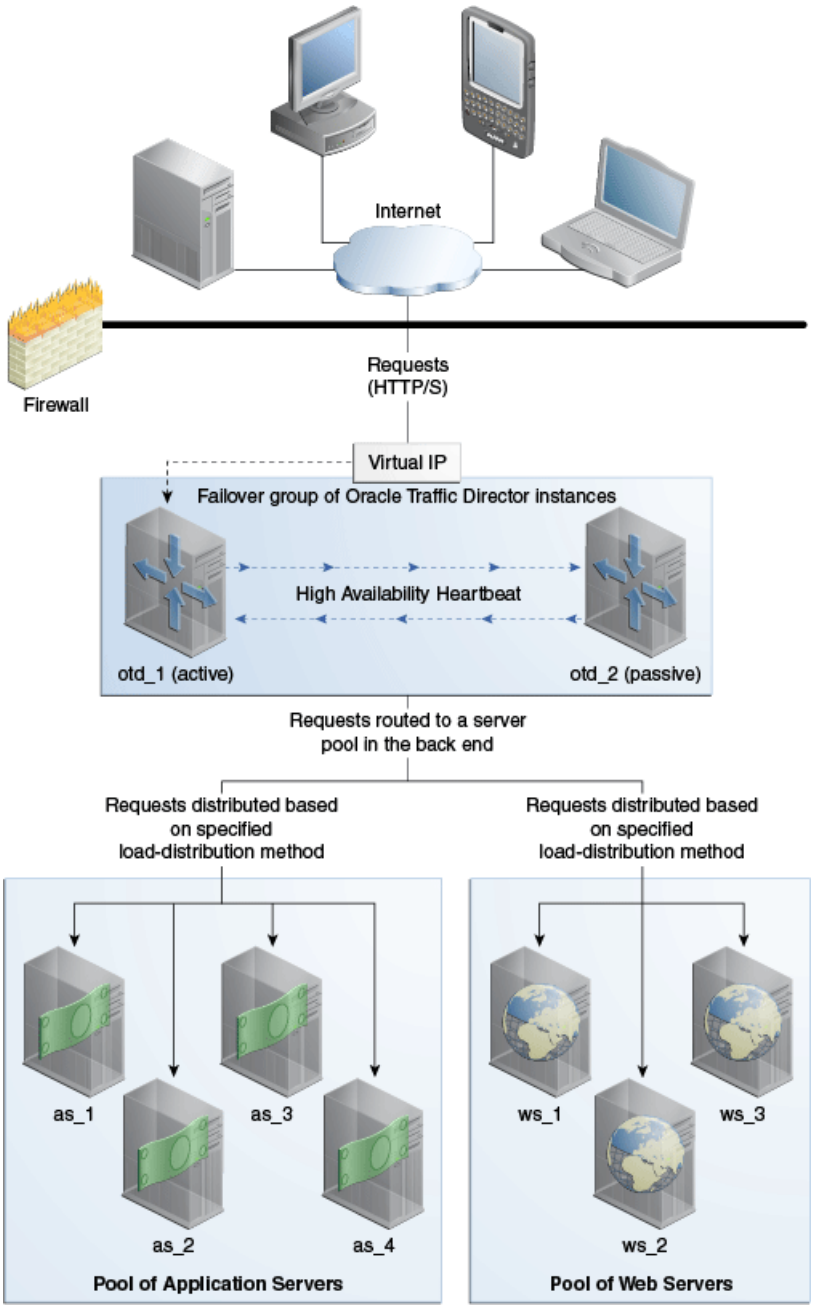
In the simplest implementation, you can have a single Oracle Traffic Director instance running on a dedicated compute node distributing client requests to a pool of servers in the back end.

To ensure that the node on which an Oracle Traffic Director instance runs does not become the single point of failure in the topology, you can have two homogenous

Oracle Traffic Director instances running on different nodes forming an active-passive failover pair.

Figure 1–1 shows a typical Oracle Traffic Director network topology for a high-availability use case in an active-passive mode.

Figure 1–1 Oracle Traffic Director Network Topology: Active-Passive Failover Mode



Oracle Traffic Director network topology for an active-passive failover mode. The figure is described in detail in the following text.

****

The topology shown in Figure 1–1 consists of two Oracle Traffic Director instances—`otd_1` and `otd_2`—forming an active-passive failover group and providing

a single virtual IP address for client requests. When the active instance (`otd_1` in this example) receives a request, it determines the server pool to which the request should be sent and forwards the request to one of the servers in the pool based on the load distribution method defined for that pool.

Note that Figure 1–1 shows only two server pools in the back end, but you can configure Oracle Traffic Director to route requests to servers in multiple pools.

In the active-passive setup described here, one node in the failover group is redundant at any point in time. To improve resource utilization, you can configure the two Oracle Traffic Director instances in active-active mode with two virtual IP addresses. Each instance caters to requests received on one virtual IP address *and* backs up the other instance.

For more information about configuring Oracle Traffic Director instances in failover groups, see Section 14.2, "Creating and Managing Failover Groups."

1.4 Oracle Traffic Director Terminology

An Oracle Traffic Director configuration is a collection of elements that define the run-time behavior of an Oracle Traffic Director instance. An Oracle Traffic Director configuration contains information about various elements of an Oracle Traffic Director instance such as listeners, origin servers, failover groups, and logs.

The following table describes the terms used in this document when describing administrative tasks for Oracle Traffic Director.

Term	Description
Configuration	A collection of configurable elements (metadata) that determine the run-time behavior of an Oracle Traffic Director instance. A typical configuration contains definitions for the listeners (IP address and port combinations) on which Oracle Traffic Director should listen for requests and information about the servers in the back end to which the requests should be sent. Oracle Traffic Director reads the configuration when an Oracle Traffic Director instance starts and while processing client requests.
Instance	An Oracle Traffic Director server that is instantiated from a configuration and deployed on an administration node.
Failover group	Two Oracle Traffic Director instances grouped by a virtual IP address (VIP), to provide high availability in active-passive mode. Requests are received at the VIP and routed to the Oracle Traffic Director instance that is designated as the primary instance. If the primary instance is not reachable, requests are routed to the backup instance. For active-active failover, two failover groups are required, each with a unique VIP, but both consisting of the same nodes with the primary and backup roles reversed. Each instance in the failover group is designated as the primary instance for one VIP and the backup for the other VIP.
Administration server	A specially configured Oracle Traffic Director instance that hosts the administration console and command-line interfaces, using which you can create and manage Oracle Traffic Director configurations, deploy instances on administration nodes, and manage the lifecycle of these instances. Note that you can deploy instances of Oracle Traffic Director configuration on the administration server. In this sense, the administration server can function as an administration node as well.

Term	Description
Administration node	A specially configured Oracle Traffic director instance that is registered with the remote administration server. The administration node running on a host acts as the agent of the remote administration server and assists the administration server in managing the instances running on the host. Note that, on a given node, you can deploy only one instance of a configuration.
INSTANCE_HOME	A directory of your choice, on the administration server or an administration node, in which the configuration data and binary files pertaining to Oracle Traffic Director instances are stored.
ORACLE_HOME	A directory of your choice in which you install the Oracle Traffic Director binaries.
Administration console	A web-based graphical interface on the administration server that you can use to create, deploy, and manage Oracle Traffic Director instances.
Client	Any agent—a browser or an application, for example—that sends HTTP, HTTPS and TCP requests to Oracle Traffic Director instances.
Origin server	A server in the back end, to which Oracle Traffic Director forwards the HTTP, HTTPS and TCP requests that it receives from clients, and from which it receives responses to client requests. Origin servers can be application servers like Oracle WebLogic Server managed servers, web servers, and so on.
Origin-server pool	A collection of origin servers that host the same application or service that you can load-balance by using Oracle Traffic Director. Oracle Traffic Director distributes client requests to servers in the origin-server pool based on the load-distribution method that is specified for the pool. Oracle Traffic Director can communicate with the origin servers in the origin-server pool directly, or through a configured HTTP forward proxy server.
Virtual server	A virtual entity within an Oracle Traffic Director server instance that provides a unique IP address (or host name) and port combination through which Oracle Traffic Director can serve requests for one or more domains. An Oracle Traffic Director instance on a node can contain multiple virtual servers. Administrators can configure settings such as the maximum number of incoming connections specifically for each virtual server. They can also customize how each virtual server handles requests.

1.5 Oracle Traffic Director Deployment Scenarios

Oracle Traffic Director can be used either as a physical application or as a virtual appliance.

- **Physical application**

  You can install Oracle Traffic Director on an Oracle Linux 5.6 or Oracle Linux 6 system and run one or more instances of the product to distribute client requests to servers in the back end.

  For information about installing Oracle Traffic Director as a physical application, see the *Oracle Traffic Director Installation Guide*.

- **Appliance running on a virtual platform**

After deploying Oracle Traffic Director as a physical application, you can create a virtual appliance from an Oracle Traffic Director instance or create an assembly containing multiple such appliances. You can then deploy the appliance or assembly on the Oracle Virtual Machine hypervisor. To enable such a deployment, Oracle provides an Oracle Traffic Director plug-in as part of Oracle Virtual Assembly Builder, a tool that you can use to build virtual appliances and assemblies from physical applications.

For more information about creating and deploying virtual assemblies containing Oracle Traffic Director instances, see the *Oracle Virtual Assembly Builder User's Guide*.

1.6 Administration Framework of Oracle Traffic Director

You can perform various administrative tasks—enabling a feature of Oracle Traffic Director, adjusting how the feature works, and instructing Oracle Traffic Director to handle requests and responses in specific ways—by using the administration interfaces provided by the *administration server*.

The following subsections describe the administration framework in detail:

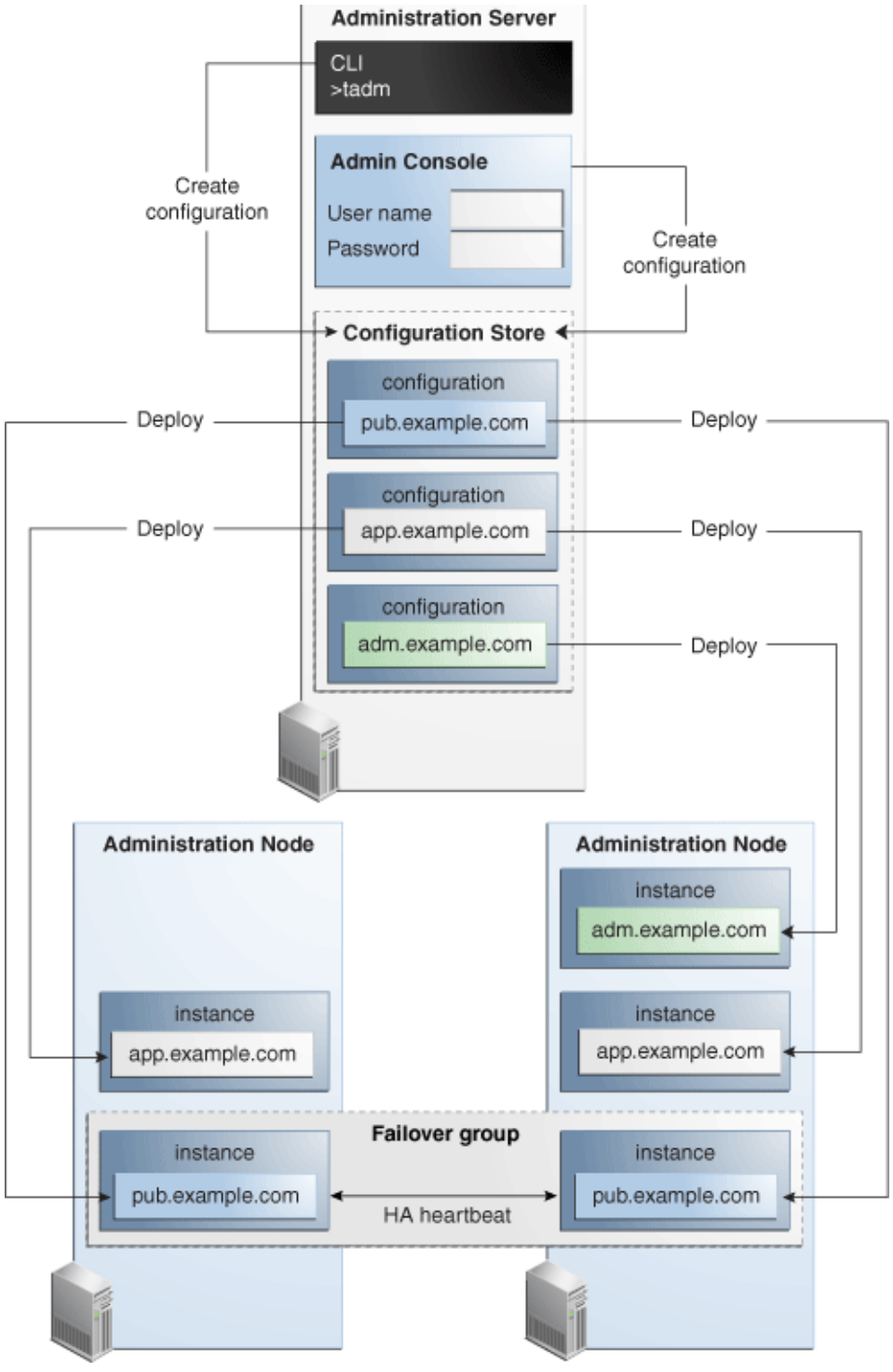
- Section 1.6.1, "Overview of the Administration Framework"
- Section 1.6.2, "Administration Server"
- Section 1.6.3, "Administration Node"
- Section 1.6.4, "Administration Interfaces"
- Section 1.6.5, "Configuration Store"
- Section 1.6.6, "Instance Configuration Files"

1.6.1 Overview of the Administration Framework

The settings that you define for Oracle Traffic Director instances are stored as configurations in a *configuration store* on the administration server. You can instantiate a configuration by deploying it as *instances* on one or more *administration nodes*.

Figure 1–2 depicts the administration framework of Oracle Traffic Director.

Figure 1–2 Administration Framework of Oracle Traffic Director



Administrative framework of Oracle Traffic Director

****

Figure 1–2 shows an administration server running on one machine, hosting the command-line interface and administration console applications. The administration interfaces are used to create three configurations—`pub.example.com`,

app.example.com, and adm.example.com, which are stored in the configuration store of the administration server.

- The adm.example.com configuration is deployed as an instance on one administration node.
- The app.example.com configuration is deployed as an instance on two administration nodes.
- The pub.example.com configuration is deployed as an instance on two administration nodes, with a high-availability heartbeat between the two nodes.

1.6.2 Administration Server

You can perform all of the administrative tasks for Oracle Traffic Director through the administration server, which is a specially configured Oracle Traffic Director instance.

The Oracle Traffic Director administration server is *not* created automatically when you install the product. You should create the administration server as described in Section 2.1, "Creating the Administration Server.".

1.6.3 Administration Node

An administration node is a physical host on which you can create Oracle Traffic Director instances.

To make a host an administration node, you should do the following:

1. Install Oracle Traffic Director on the host, or mount a remote installation of Oracle Traffic Director on a local directory on the host.
2. Register the host with the administration server by running the configure-server command. This command designates the host as an Oracle Traffic Director administration node and registers the administration node with a remote administration server.

You can now create instances of Oracle Traffic Director configurations on the administration node. Note that on an administration node, you can create only one instance of a particular configuration.

For more information about creating administration nodes and managing them, see Section 3, "Managing Administration Nodes."

1.6.4 Administration Interfaces

The administration server of Oracle Traffic Director provides the following interfaces through which you can create, modify, and manage Oracle Traffic Director instances:

- **Command-line interface**

  Oracle Traffic Director provides a command-line interface (CLI) that supports a wide range of administrative operations. The syntax of the command-line interface is easy to understand and use. While you use the interface, you can look up help for specific commands and options. For information about accessing the CLI, see Section 2.3.1, "Accessing the Command-Line Interface."

- **Administration console**

  The administration console is an web-based graphical interface consisting of a set of screens and wizards that you can use to create, monitor, and manage Oracle Traffic Director instances. For information about accessing the administration console, see Section 2.3.2, "Accessing the Administration Console."

1.6.5 Configuration Store

All of the configurable elements of an Oracle Traffic Director instance are stored as a configuration, which is a set of files created in a **configuration store** in the following directory:

INSTANCE_HOME/admin-server/**config-store**/*config_name*/config

config_name is the name that you specified for the configuration while creating it.

The files in the configuration store are meant for internal use by Oracle Traffic Director. They can be created, updated, and deleted only through the administration interfaces—administration console and command-line interface.

Caution: The files in the configuration store are updated automatically when you edit a configuration by using either the administration console or the CLI.

DO NOT edit the files in the configuration store manually.

1.6.6 Instance Configuration Files

When you create instances of an Oracle Traffic Director configuration, the configuration files that represent the configuration are copied from the administration server to the *INSTANCE_HOME*/net-*config_name*/config directory on each of the administration nodes.

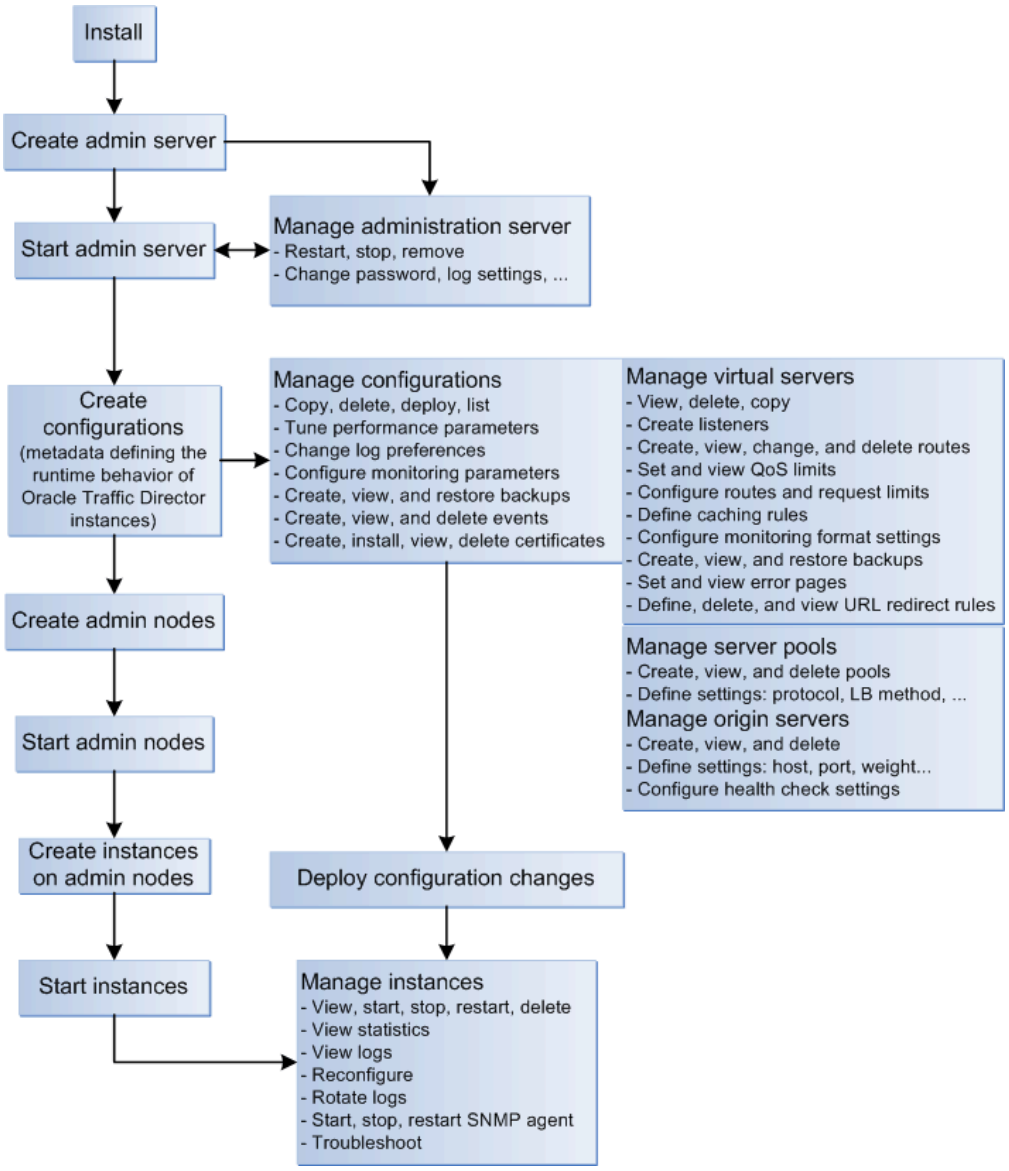
Oracle Traffic Director uses the configuration files in the *INSTANCE_HOME*/net-*config_name*/config directory when the instance starts and while it processes requests from clients.

For information about the content and structure of the configuration files, see the *Oracle Traffic Director Configuration Files Reference*.

1.7 Overview of Administration Tasks

Figure 1–3 shows the typical order of tasks that you should perform to create and manage Oracle Traffic Director instances.

Figure 1–3 Oracle Traffic Director Administration Workflow



Oracle Traffic Director administration workflow. The figure is described in detail in the following text.

****

> **Note:** As the administrator of Oracle Traffic Director, you might perform several additional tasks such as managing security, tuning for performance, and troubleshooting problems. These tasks are not shown in the flowchart because they are not necessarily performed at definite points in a fixed sequence. All of these additional tasks are described in other chapters of this document.

- Install the product

  You can install Oracle Traffic Director on Oracle Linux 5.6+ or Oracle Linux 6 on an x86_64 system, by using an interactive graphical wizard or in silent mode.

For more information, see the *Oracle Traffic Director Installation Guide*.

- Create the administration server

  After installing the product, you should create an administration server instance of Oracle Traffic Director. The administration server is a specially configured Oracle Traffic Director virtual server that you can use to administer Oracle Traffic Director instances.

  For more information, see "Creating the Administration Server Instance" in the *Oracle Traffic Director Installation Guide*.

- Manage the administration server

  At times, you might want to stop the administration server and restart it, or change settings such as the administrator user name and password.

  For more information, see Chapter 2, "Managing the Administration Server."

- Access the administration console and command-line interface

  You can use the administration console and command-line interface of Oracle Traffic Director to create, modify, and monitor Oracle Traffic Director instances.

  For information about accessing the administration console and command-line interface, see Section 2.3, "Accessing the Administration Interfaces."

- Create and manage administration nodes

  Administration nodes are physical hosts on which you can create Oracle Traffic Director instances.

  For information about managing administration nodes, see Chapter 3, "Managing Administration Nodes."

- Create and manage configurations

  After creating the administration nodes, create configurations that define your Oracle Traffic Director instances. A configuration is a collection of metadata that you can use to instantiate Oracle Traffic Director. Oracle Traffic Director reads the configuration when a server instance starts and while processing client requests.

  For information about managing configurations, see Chapter 4, "Managing Configurations."

- Create and manage instances

  After creating a configuration, you can create Oracle Traffic Director server instances by deploying the configuration on one or more hosts. You can view the current state of each instance, start or stop it, reconfigure it to reflect configuration changes, and so on.

  For information about managing instances, see Chapter 5, "Managing Instances."

- Define and manage origin-server pools

  For an Oracle Traffic Director instance to distribute client requests, you should define one or more origin-server pools or in the back end. For each origin-server pool, you can define the load-distribution method that Oracle Traffic Director should use to distribute requests. In addition, for each origin server in a pool, you can define how Oracle Traffic Director should control the request load.

  For more information, see Chapter 6, "Managing Origin-Server Pools" and Chapter 7, "Managing Origin Servers."

- Create and manage virtual servers and listeners

An Oracle Traffic Director instance running on a node contains one or more virtual servers. Each virtual server provides one or more listeners for receiving requests from clients. For each virtual server, you can configure parameters such as the origin-server pool to which the virtual server should route requests, the quality of service settings, request limits, caching rules, and log preferences.

For more information, see Chapter 8, "Managing Virtual Servers" and Chapter 10, "Managing Listeners."

- Manage security

  Oracle Traffic Director, by virtue of its external-facing position in a typical network, plays a critical role in protecting data and applications in the back end against attacks and unauthorized access from outside the network. In addition, the security and integrity of data traversing through Oracle Traffic Director to the rest of the network needs to be guaranteed.

  For more information, see Chapter 11, "Managing Security."

- Manage Logs

  Oracle Traffic Director records data about server events such as configuration changes, instances being started and stopped, errors while processing requests, and so on in log files. You can use the logs to troubleshoot errors and to tune the system for improved performance.

  For more information, see Chapter 12, "Managing Logs."

- Monitor statistics

  The state and performance of Oracle Traffic Director instances are influenced by several factors: configuration settings, volume of incoming requests, health of origin servers, nature of data passing through the instances, and so on. As the administrator, you can view metrics for all of these factors through the command-line interface and administration console, and extract the statistics in the form of XML files for detailed analysis. You can also adjust the granularity at which Oracle Traffic Director collects statistics.

  For more information, see Chapter 13, "Monitoring Oracle Traffic Director Instances."

- Set up Oracle Traffic Director instances for high availability

  In the event that an Oracle Traffic Director instance or the node on which it runs fails, you need to ensure that the load-balancing service that the instance provides continues to be available uninterrupted. You can achieve this goal by configuring a backup Oracle Traffic Director instance that can take over processing of requests when the primary instance fails.

  For more information, see Chapter 14, "Configuring Oracle Traffic Director for High Availability."

- Tune for performance

  Based on your analysis of performance statistics and to respond to changes in the request load profile, you might want to adjust the request processing parameters of Oracle Traffic Director to maintain or improve the performance. Oracle Traffic Director provides a range of performance-tuning controls and knobs that you can use to limit the size and volume of individual requests, control timeout settings, configure thread pool settings, SSL/TLS caching behavior, and so on.

  For more information, see Chapter 15, "Tuning Oracle Traffic Director for Performance."

- Diagnose and troubleshoot problems

  Despite the best possible precautions, you might occasionally run into problems when installing, configuring, and monitoring Oracle Traffic Director instances. You can diagnose and solve some of these problems based on the information available in error messages and logs. For complex problems, you would need to gather certain data that Oracle support personnel can use to understand, reproduce, and diagnose the problem.

  For more information, see Chapter 16, "Diagnosing and Troubleshooting Problems."

1.8 Setting Up a Simple Load Balancer Using Oracle Traffic Director

This section describes how you can set up a load-balanced service using Oracle Traffic Director with the minimum necessary configuration. The purpose of this section is to reinforce and illustrate the concepts discussed earlier in this chapter and to prepare you for the configuration tasks described in the remaining chapters.

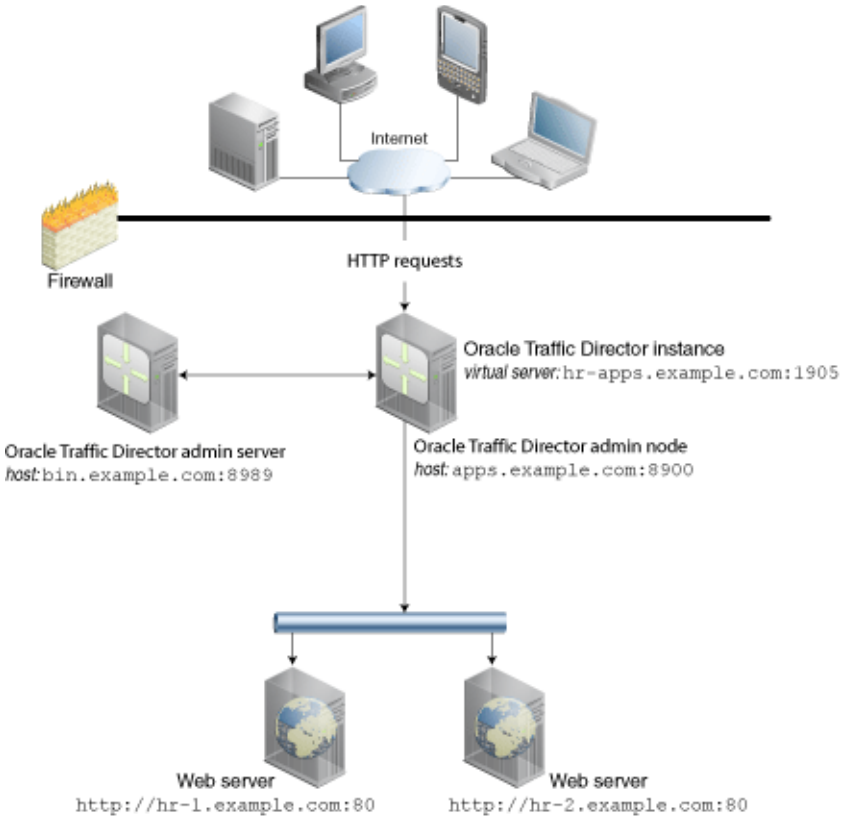
This section contains the following topics:

1.8.1 Example Topology

In this example, we will create a single instance of Oracle Traffic Director that will receive HTTP requests and distribute them to two origin servers in the back end, both serving identical content.

Figure 1–4 shows the example topology.

Figure 1–4 Oracle Traffic Director Deployment Example



Oracle Traffic Director example topology

****

The example topology is based on the following configuration:

- Administration server host and port: `bin.example.com:8989`
- Administration node host and port: `apps.example.com:8900`
- Virtual server host and port to receive requests from clients: `hr-apps.example.com:1905`
- Host and port of origin servers (web servers in this example):
  - `hr-1.example.com:80`
  - `hr-2.example.com:80`

  In the real world, both origin servers would serve identical content. But for this example, to be able to see load balancing in action, we will set up the `index.html` page to which the `DocumentRoot` directive of the web servers points, to show slightly different content, as follows:
  - For `hr-1.example.com:80`: "**Page served from origin-server 1**"
  - For `hr-2.example.com:80`: "**Page served from origin-server 2**"
- Load-balancing method: Round robin

1.8.2 Creating the Load Balancer for the Example Topology

This section describes how to set up the topology described in Section 1.8.1, "Example Topology."

1. Install Oracle Traffic Director on the hosts `bin.example.com` and `apps.example.com`, as described in the *Oracle Traffic Director Installation Guide*.
2. On `bin.example.com` create the administration server instance by using the `configure-server` CLI command.

```
> $ORACLE_HOME/bin/tadm configure-server --port=8989 --user=admin
 --instance-home=/production/otd/

This command will create an Administration Server. The password that is
 provided will be required to access the Administration Server.
Enter admin-user-password>
Enter admin-user-password again>
OTD-70214 The Administration Server has been configured successfully.
The server can be started by executing:
/production/otd/admin-server/bin/startserv
The Administration Console can be accessed at https://bin.example.com:8989
using user name 'admin'.
```

3. Start the administration server.

```
> /production/otd/admin-server/bin/startserv

Oracle Traffic Director 11.1.1.7.0 B01/14/2013 09:08
[NOTIFICATION:1] [OTD-80118] Using [Java HotSpot(TM) 64-Bit Server VM, Version
1.6.0_29] from [Sun Microsystems Inc.]
[NOTIFICATION:1] [OTD-80000] Loading web module in virtual server
[admin-server] at [/admin]
[NOTIFICATION:1] [OTD-80000] Loading web module in virtual server
[admin-server] at [/jmxconnector]
[NOTIFICATION:1] [OTD-10358] admin-ssl-port: https://bin.example.com:8989 ready
to accept requests
[NOTIFICATION:1] [OTD-10487] successful server startup
```

4. On the `apps.example.com` host, run the `configure-server` command to register the host with the remote administration server as an administration node.

```
> $ORACLE_HOME/bin/tadm configure-server --user=admin --port=8989
 --host=bin.example.com --admin-node --node-port=8900
 --instance-home=/home/otd-instances

This command will create an Administration Node and register it with the remote
Administration Server: https://bin.example.com:8989.
Enter admin-user-password>
OTD-70215 The Administration Node has been configured successfully.
The node can be started by executing:
/home/otd-instances/admin-server/bin/startserv
```

5. Start the administration node.

```
> /home/otd-instances/admin-server/bin/startserv

Oracle Traffic Director 11.1.1.7.0 B01/14/2013 09:08
[NOTIFICATION:1] [OTD-80118] Using [Java HotSpot(TM) 64-Bit Server VM, Version
1.6.0_29] from [Sun Microsystems Inc.]
[NOTIFICATION:1] [OTD-80000] Loading web module in virtual server
[admin-server] at [/jmxconnector]
```

```
[NOTIFICATION:1] [OTD-10358] admin-ssl-port: https://apps.example.com:8900
ready to accept requests
[NOTIFICATION:1] [OTD-10487] successful server startup
```

6. On the administration server (`bin.example.com`), create a configuration named `hr-config`, by using the `create-config` CLI command.

```
> $ORACLE_HOME/bin/tadm create-config --user=admin --port=8989
 --listener-port=1905 --server-name=hr-apps.example.com
 --origin-server=hr-1.example.com:80,hr-2.example.com:80 hr-config

Enter admin-user-password>
OTD-70201 Command 'create-config' ran successfully.
```

7. Create an instance of the configuration `hr-config` on the administration node `apps.example.com`, by running the `create-instance` CLI command from the administration server.

```
> $ORACLE_HOME/bin/tadm create-instance --user=admin --port=8989
 --config=hr-config apps.example.com

Enter admin-user-password>
OTD-70201 Command 'create-instance' ran successfully.
```

8. Start the Oracle Traffic Director instance that you just created on `apps.example.com`, by running the `start-instance` CLI command from the administration server.

```
> $ORACLE_HOME/bin/tadm start-instance --config=hr-config

CLI204 Successfully started the server instance.
```

Note: The steps in this procedure use only the CLI, but you can perform step 6 onward by using the administration console as well.

We have now successfully created an Oracle Traffic Director configuration, instantiated it on an administration node, and started the instance.

1.8.3 Verifying the Load-Balancing Behavior of the Oracle Traffic Director Instance

The Oracle Traffic Director instance that we created and started earlier is now listening for HTTP requests at the URL `http://hr-apps.example.com:1905`.

This section describes how you can verify the load-balancing behavior of the Oracle Traffic Director instance by using your browser.

Note:

- Make sure that the web servers `hr-1.example.com:80` and `hr-2.example.com:80` are running.
- If necessary, update the `/etc/hosts` file on the host from which you are going to access the Oracle Traffic Director virtual server, to make sure that the browser can resolve `hr-apps.example.com` to the correct IP address.

1. Enter the URL `http://hr-apps.example.com:1905` in your browser.

   A page with the following text is displayed:

   ```
   "Page served from origin-server 1"
   ```

   This indicates that the Oracle Traffic Director instance running on the `apps.example.com` administration node received the HTTP request that you sent from the browser, and forwarded it to the origin server `hr-1.example.com:80`.

2. Send another HTTP request to `http://hr-apps.example.com:1905` by refreshing the browser window.

   A page with the following text is displayed:

   ```
   "Page served from origin-server 2"
   ```

   This indicates that Oracle Traffic Director sent the second request to the origin server `hr-2.example.com:80`

3. Send a third HTTP request to `http://hr-apps.example.com:1905` by refreshing the browser window again.

   A page with the following text is displayed:

   ```
   "Page served from origin-server 1"
   ```

   This indicates that Oracle Traffic Director used the simple round-robin load-distribution method to send the third HTTP request to the origin server `hr-1.example.com:80`.

2

Managing the Administration Server

The administration server is a specially configured Oracle Traffic Director virtual server that you can use to create, monitor, and manage Oracle Traffic Director instances.

For information about the role of the administration server in the administrative framework of Oracle Traffic Director, see Section 1.6, "Administration Framework of Oracle Traffic Director."

This chapter describes how to create, remove, start, stop, and restart the administration server; and how to configure its settings. It also describes how to access the administration interfaces of Oracle Traffic Director—the administration console and the command-line interface.

This chapter contains the following sections:

2.1 Creating the Administration Server

The Oracle Traffic Director administration server provides the interfaces that you can use to perform all of the administrative tasks for Oracle Traffic Director.

To create the administration server, do the following:

1. Run the `configure-server` CLI command as shown in the following example:

```
> $ORACLE_HOME/bin/tadm configure-server --port=8989 --user=admin
 --instance-home=/production/otd/
```

`ORACLE_HOME` is the directory in which you installed Oracle Traffic Director.

> **Note:** The user name can contain a maximum of 100 characters and must not contain spaces.

The following message and prompt are displayed.

```
This command will create the administration server.
The password that is provided will be required to access the administration
server.
Enter admin-user-password>
```

2. Enter the administrator password.

   You will later use this password to log in to the Oracle Traffic Director administration console.

   A prompt to enter the password again is displayed.

   ```
   Enter admin-user-password again>
   ```

3. Confirm the administrator password by entering it again.

The files and resources pertaining to the administration server (configuration artifacts, executable commands, and so on) are located in the `admin-server` subdirectory within the `instance-home` directory specified with the `configure-server` command.

For each of the Oracle Traffic Director configurations that you subsequently instantiate on the host that contains the administration server, a directory named `net-`*`config_name`* is created within the `instance-home` directory, as shown in the following example:

```
/production/otd
   admin-server
   net-test
   net-soa
   net-adf
```

For more information about the `configure-server` command, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

2.2 Starting the Administration Server

To be able to use the administration interfaces—administration console and command-line interface, the administration server should be running.

To start the administration server, run the following command:

```
> $INSTANCE_HOME/admin-server/bin/startserv
```

`INSTANCE_HOME` is the directory that contains all the Oracle Traffic Director instances, including the administration server instance. This is the directory that you specified with the `instance-home` option while creating the administration server by using the `configure-server` command.

The `startserv` command starts the administration server using the port that you specified while creating the administration server.

Wait for the `successful server startup` message to be displayed, as shown in the following example:

```
Oracle Traffic Director 11.1.1.7.0 B01/14/2013 09:08
[NOTIFICATION:1] [OTD-80118] Using [Java HotSpot(TM) 64-Bit Server VM, Version
1.6.0_29] from [Sun Microsystems Inc.]
[NOTIFICATION:1] [OTD-80000] Loading web module in virtual server [admin-server]
at [/admin]
[NOTIFICATION:1] [OTD-80000] Loading web module in virtual server [admin-server]
at [/jmxconnector]
[NOTIFICATION:1] [OTD-10358] admin-ssl-port: https://bin.example.com:8989 ready to
```

```
accept requests
[NOTIFICATION:1] [OTD-10487] successful server startup
```

You can now use the administration interfaces of Oracle Traffic Director—administration console and command-line interface—to configure and manage Oracle Traffic Director instances.

To use the administration console and the command-line interface, you should log in by using the user name and password that you specified while creating the administration server. For more information, see Section 2.3, "Accessing the Administration Interfaces."

2.3 Accessing the Administration Interfaces

This section contains the following topics:

- Accessing the Command-Line Interface
- Accessing the Administration Console

> **Note:** To be able to use the administration interfaces, the administration server should be running. For information about starting the administration server, see Section 2.2, "Starting the Administration Server."

2.3.1 Accessing the Command-Line Interface

You can access the command-line interface (CLI) of Oracle Traffic Director by running the **`tadm`** command from the `ORACLE_HOME/bin` directory, as follows:

```
./tadm [subcommand] --user=admin_user --host=adminserver_host
[--password-file=path_to_file] --port=adminserver_port
```

The CLI uses password-based authentication to allow access to the administration server. If you do not specify the `--password-file` option, a prompt to enter the administrator user password is displayed. After you enter the password, the specified subcommand is executed.

The `tadm` command supports a comprehensive set of subcommands that you can use to create, view, update, and manage settings for all of the features of Oracle Traffic Director. If you run the `tadm` command without specifying the subcommand, you enter the *shell* mode of the CLI. In the shell mode, the options to connect to the administration server—`user`, `host`, `port`, and `password`—have already been specified; so you can run individual subcommands without specifying the connection options each time.

You can view help for a subcommand by running the subcommand with the `--help` option.

For more information about using the CLI, including the usage modes (standalone, shell, and file), the subcommands that the `tadm` command supports, and the options for each subcommand, see the *Oracle Traffic Director Command-Line Reference*.

2.3.2 Accessing the Administration Console

The administration console is a browser-based graphical interface that enables you create, configure, and monitor Oracle Traffic Director instances.

For information about the web browser versions that are supported for the Oracle Traffic Director administration console, see the Oracle Fusion Middleware Supported System Configurations at: http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html.

To access the administration console, do the following:

1. Go to the administration console URL.

   The URL for the administration console depends on the host name and the port number that you specified while creating the administration server, as described in "Creating the Administration Server Instance" in the *Oracle Traffic Director Installation Guide*.

   For example, if you created the administration server with port `1895` on the `admin.example.com` host, the URL for the administration console would be the following:

   ```
   https://admin.example.com:8989
   ```

   Note: Communication with the administration server takes place over SSL. If you try to use the `http://` schema in the URL to access the administration console, you are redirected automatically to the `https://` URL.

   The SSL-enabled administration server uses a self-signed digital certificate rather than one issued by a trusted certificate authority. So, the first time you attempt to access the administration console, an `invalid security certificate` message is displayed.

2. Proceed to the log-in page of the administration console by choosing to trust the certificate.

   The steps to be performed to trust a certificate vary depending on the browser you use. For example, in Mozilla Firefox 4.0, click on the **I Understand the Risks** link on the error page, then click the **Add Exception** button, and finally, on the resulting page, click the **Confirm Security Exception** button.

   The log-in page of the Oracle Traffic Director administration console is displayed.

Figure 2–1 Oracle Traffic Director Administration-Console Log-In Page

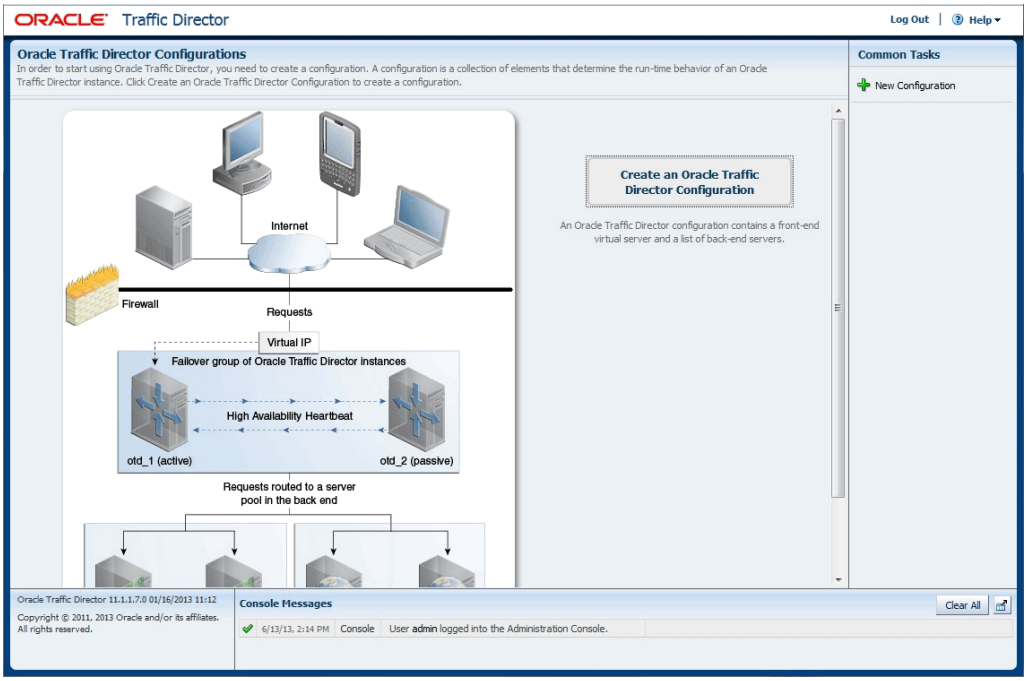
Oracle Traffic Director administration console log-in page

****

3. Enter the administrator user name and password that you specified while creating the administration server.

   The home page of the administration-console is displayed.

Figure 2–2 Oracle Traffic Director Administration-Console Home Page



Oracle Traffic Director administration-console home page

****

You can now create Oracle Traffic Director configurations and deploy them as instances on administration nodes. For more information, see Chapter 4, "Managing Configurations."

Note: If the administration-console browser session remains idle for 30 minutes, you will be logged out and the log-in page will be displayed.

2.4 Stopping and Restarting the Administration Server

At times, you might want to create the administration server instance afresh with new settings. Before attempting to re-create the administration server, you should stop the running administration server as described in this section. In some situations, such as when you change the administrator password or the administrator server port, for the changes to take effect, you should restart the administration server as described in this section.

You can stop and restart the administration server by using either the administration console or the CLI.

Note: If you stop the administration server, the administration console will not be available again until you restart the administration server.

Stopping and Restarting the Administration Server Using the Administration Console

To stop or restart the administration server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button near the upper left corner of the page.

   A list of available nodes is displayed.
3. From the list of nodes, select **Administration Server**.
4. In the Common Tasks pane, click **Restart** or **Stop**.

   A dialog box is displayed prompting you to confirm restarting or stopping the administration server. Click **OK**.

If you clicked **Restart**, then, after the administration server restarts, the log-in page of the administration console is displayed.

If you clicked **Stop**, then, after the administration server stops, a dialog box is displayed indicating that the browser is unable to communicate with the administration server. Start the administration server as described in Section 2.2, "Starting the Administration Server." Then, click the **Reload** button in the dialog box to bring up the log-in page of the administration console.

Stopping the Administration Server Using the CLI

To stop the administration server, run the `stop-admin` command:

```
> $ORACLE_HOME/bin/tadm stop-admin --user=admin_server_user --port=admin_server_
```

port node_host

node_host is the name or IP address of the host on which the administration server instance is deployed.

At the prompt, enter the administration user password.

After the administration server shuts down, the following message is displayed:

```
OTD-70201 Command 'stop-admin' ran successfully.
```

> **Note:** Stopping the administration server has no effect on the state of Oracle Traffic Director instances.

Restarting the Administration Server Using the CLI

To restart the administration server by using the CLI, run the following command:

```
> $ORACLE_HOME/bin/tadm restart-admin --user=admin_server_user --port=admin_
server_port node_host
```

node_host is the name or IP address of the host on which the administration server instance is deployed.

At the prompt, enter the administration user password.

After the administration server restarts, the following message is displayed:

```
OTD-70201 Command 'restart-admin' ran successfully.
```

> **Note:** Alternatively, you can use the following commands to stop and restart the administration server:
>
> ```
> > $INSTANCE_HOME/admin-server/bin/stopserv
> ```
>
> ```
> > $INSTANCE_HOME/admin-server/bin/restart
> ```

2.5 Viewing Administration Server Settings

You can view the settings of the administration server by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (tadm>). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing the Administration Server Settings Using the Administration Console

To view the current properties of the administration server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button that is situated near the upper left corner of the page.

   A list of available nodes is displayed.

> **Note:** The Nodes button is available only after you have created at least one new configuration.

3. From the list of nodes, select **Administration Server**.

   The General Settings page is displayed. You can view the authentication settings by clicking **Authentication** in the navigation pane.

Viewing the Administration Server Settings Using the CLI

To view the current properties of the administration server by using the CLI, run the following command:

```
tadm> get-admin-prop
```

The current properties of the administration server are displayed as shown in the following example:

```
instance-home=/production/otd
java-home=/production/otd/jdk
admin-node=false
server-version=Oracle Traffic Director 11.1.1.7.0 B01/14/2013 09:08
admin-user=admin
server-user=joe
ssl-port=8989
log-file=../logs/server.log
log-level=NOTIFICATION:1
access-log-file=../logs/access.log
host=adm.example.com
description=This is the Administration Server node
```

These are the properties that you specified, or were set by default, when you created the administration server by using the `configure-server` CLI command.

2.6 Changing Administration Server Settings

You can change the administration server settings by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Changing the Administration Server Settings Using the Administration Console

To change the properties of the administration server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button that is situated near the upper left corner of the page.

   A list of available nodes is displayed.
3. From the list of nodes, select **Administration Server**.

   The General Settings page is displayed. On this page you can do the following:

- Change the SSL port number on which the administration server communicates.
- Change the path to the JDK that the administration server process should use.
- Change the locations of the access and server logs, and the server log level.
- Change the user ID with which the administration server runs. Note that you can change the user ID only when the administration server is running as the `root` user and if there are no instances running on the administration server.

You can also set and configure a pin for the `internal` token for the administration server's certificates database, and change and configure the authentication mode for the administration server. For more information, see Section 11.1, "Securing Access to the Administration Server."

4. Specify the parameters that you want to change, and then click **Save**.

   A message is displayed in the Console Messages pane indicating that the updated settings are saved.

5. Restart the administration server by clicking **Restart** in the Common Tasks pane.

Changing the Administration Server Settings Using the CLI

To change the settings of the administrator server by using the CLI, run the following command:

```
tadm> set-admin-prop (property=value)+
```

You can specify one or more `property=value` pairs separated by spaces, as shown in the following example:

```
tadm> set-admin-prop ssl-port=8900 log-level=WARNING:1
```

For information about the properties that you can set by using the `set-admin-prop` command, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Note: For the changes to take effect, you must restart the administration server.

2.7 Removing the Administration Server Instance

To remove the administration server instance, run the following command from the shell prompt of the administration server host:

Caution: If you remove the administration server instance, the administration console and the CLI will no longer be available, any Oracle Traffic Director instances that are deployed on the administration server host will be removed, and you can no longer manage Oracle Traffic Director instances deployed on administration nodes.

```
$ORACLE_HOME/bin/tadm unconfigure-server --instance-home=absolute_path
```

After the administration server instance is removed, the following message is displayed:

```
OTD-70201 Command 'unconfigure-server' ran successfully
```

3 Managing Administration Nodes

After installing Oracle Traffic Director and creating the administration server on a particular host, you can create Oracle Traffic Director server instances on the same host. However, typically, you might want to deploy Oracle Traffic Director server instances on other hosts that are remote from the host on which the administration server runs. For example, to ensure high availability of the Oracle Traffic Director service, you can deploy instances of a configuration on two distinct hosts.

When you want to create Oracle Traffic Director server instances on hosts other than that on which you created the administration server, you must first designate those other hosts as administration nodes and register them with the administration server.

This chapter describes the procedure to create administration nodes and to start, stop, restart, and remove them.

This chapter contains the following sections:

3.1 Creating an Administration Node

To create an administration node, make sure that the administration server is running, and then run the `configure-server` command from the shell prompt of the host that you want to designate as the administration node.

For example, to designate `an.example.com` as an Oracle Traffic Director administration node that is registered with the administration server on the host `adm.example.com`, do the following:

1. Run the `configure-server` command from the `an.example.com` system.

```
$ORACLE_HOME/bin/tadm configure-server --user=admin --host=adm.example.com
 --port=8989 --admin-node --node-host=an.example.com --node-port=8900
 --instance-home=/home/otd/instances
```

   - `ORACLE_HOME` is the path to the directory or mount point containing the Oracle Traffic Director installation on the node that you want to designate as the administration node.

- The `--instance-home` option specifies the directory in which the instance directories should be created on the node.
- The `--admin-node` option specifies that the specified host should be configured as an administration node.

The following message is displayed.

```
This command will create an administration node and register it with the remote
administration server: https://adm.example.com:8989.
Enter admin-user-password>
```

2. Enter the password for the administration server user.

   The `configure-server` command attempts to connect to the remote administration server by using the specified administration server host, port, user, and password.

   Note: If the administration server is not reachable, the following error message is displayed.

   ```
   OTD-70104 Unable to communicate with the administration server:
    Unable to connect to admin-server. Please check if the server is
    up and running and that the host and port provided are correct.
   ```

   Start the administration server and run the `configure-server` command again.

   If this is the first time that the host on which you are creating the administration node is attempting to connect to the administration server, the server certificate of the administration server is displayed.

3. Enter `y` to trust the certificate.

   The following message is displayed:

   ```
   OTD-70215 The administration node has been configured successfully.
   The node can be started by executing:
   /home/otd/instances/admin-server/bin/startserv
   ```

For more information about `configure-server`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

For information about starting the administration node, see Section 3.3, "Starting an Administration Node."

After you start the administration node, you can create instances of Oracle Traffic Director configurations on the administration node. Note that on each administration node, you can create only one instance of a configuration.

3.2 Viewing a List of Administration Nodes

You can view a list of the administration nodes by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing a List of Administration Nodes Using the Administration Console

To view a list of the available administration nodes by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button near the upper left corner of the page.

   The administration server and the administration nodes that are registered with it are displayed as shown in Figure 3–1. For each node, the names of the configurations that have been instantiated on the node are also displayed.

Figure 3–1 List of Administration Nodes

List of administration nodes

****

To view the settings of an administration node in detail, click on the node.

Viewing a List of Administration Nodes Using the CLI

To view a list of the administration nodes, run the `list-nodes` command, as shown in the following example:

```
tadm> list-nodes --verbose --all
node-name         node-port  node-online  node-description
--------------------------------------------------------------
adm.example.com    8989       true         "This is the Administration Server
node"
an.example.com     8900       false        -
```

3.3 Starting an Administration Node

For the administration server to communicate with a remote administration node, the node must be running.

To start an administration node, run the following command on the node host:

```
$INSTANCE_HOME/admin-server/bin/startserv
```

The following messages are displayed:

```
Oracle Traffic Director 11.1.1.7.0 B01/14/2013 09:08
[NOTIFICATION:1] [OTD-80118] Using [Java HotSpot(TM) 64-Bit Server VM, Version
1.6.0_29] from [Sun Microsystems Inc.]
[NOTIFICATION:1] [OTD-80000] Loading web module in virtual server [admin-server]
at [/jmxconnector]
[NOTIFICATION:1] [OTD-10358] admin-ssl-port: https://an.example.com:8900 ready to
accept requests
[NOTIFICATION:1] [OTD-10487] successful server startup
```

3.4 Changing the Properties of an Administration Node

You can change the properties of an administration node by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Changing the Properties of an Administration Node Using the Administration Console

To change the properties of an administration node by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button that is situated near the upper left corner of the page.

   A list of available nodes is displayed.
3. From the list of nodes, select the node for which you want to change properties.

   The General Settings page is displayed.
4. Specify the parameters that you want to change, and then click **Save**.

   A message is displayed in the Console Messages pane indicating that the updated settings are saved.
5. Restart the administration server by clicking **Restart** in the Common Tasks pane.

Changing the Properties of an Administration Node Using the CLI

To change the properties of an administration node by using the CLI, run the following command:

```
tadm> set-admin-prop --node=node_name (property=value)+
```

You can specify one or more `property=value` pairs separated by spaces, as shown in the following example:

```
tadm> set-admin-prop --node=apps.example.com ssl-port=8900 log-level=warning
```

For information about the properties that you can set by using the `set-admin-prop` command, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

> **Note:** For the changes to take effect, you should restart the administration node as described in Section 3.5, "Stopping and Restarting an Administration Node."

3.5 Stopping and Restarting an Administration Node

You can stop and restart administration nodes by using either the administration console, CLI commands, or shell commands.

Note: For information about stopping and restarting the administration server, see Section 2.4, "Stopping and Restarting the Administration Server."

Stopping and Restarting an Administration Node Using the Administration Console

To stop or restart an administration node by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button near the upper left corner of the page.

   The administration server and all of the administration nodes that are registered with it are displayed.
3. From the list of nodes, select the node that you want to stop or restart.
4. In the Common Tasks pane, select **Restart** or **Stop**, as required.

Stopping and Restarting an Administration Node Using the CLI

- To stop an administration node, run the following command:

  ```
  tadm> stop-admin node_host
  ```

  The following message is displayed:

  ```
  OTD-70201 Command 'stop-admin' ran successfully.
  ```

- To restart an administration node, run the following command:

  ```
  tadm> restart-admin node_host
  ```

  The following message is displayed:

  ```
  OTD-70201 Command 'restart-admin' ran successfully.
  ```

For more information about the `stop-admin` and `restart-admin` commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

Stopping and Restarting an Administration Node Using Shell Commands

- To stop an administration node from the shell, run the following command:

  ```
  $INSTANCE_HOME/admin-server/bin/stopserv
  ```

  The following message is displayed:

  ```
  server has been shutdown
  ```

- To restart an administration node from the shell, run the following command:

  ```
  $INSTANCE_HOME/admin-server/bin/restart
  ```

3.6 Removing an Administration Node

To remove an administration node, do the following:

1. From the administration server host, run the `remove-node` command with the `--force` option.

   ```
   tadm> remove-node --force node_host
   ```

   If Oracle Traffic Director instances exist on the node, the `remove-node` command fails, unless you specify the `--force` option.

2. On the administration node host, run the `unconfigure-server` command, as shown in the following example:

   ```
   $ORACLE_HOME/bin/tadm unconfigure-server --instance-home=home/otd/instances
   ```

   This command stops and removes all instances of Oracle Traffic Director that are currently deployed on the administration node, and stops the administration node process on the host. You can no longer create Oracle Traffic Director instances on that host until you designate it again as an administration node.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

Part II

Basic Administration

Part II contains the following chapters:

- Chapter 4, "Managing Configurations" describes how to create and manage configurations, which are collections of metadata that determine the runtime behavior of Oracle Traffic Director instances.
- Chapter 5, "Managing Instances" describes how to create and manage Oracle Traffic Director instances.
- Chapter 6, "Managing Origin-Server Pools" describes how to create and manage pools of servers in the back end, to which Oracle Traffic Director instances can route client requests.
- Chapter 7, "Managing Origin Servers" describes how to add and manage servers in origin-server pools.
- Chapter 8, "Managing Virtual Servers" describes how to create and manage virtual servers to process client request, and how to create and manage route rules.
- Chapter 9, "Managing TCP Proxies" describes how to create and manage TCP proxies to handle TCP requests.
- Chapter 10, "Managing Listeners" describes how to create and manage HTTP listeners for virtual servers and TCP listeners for TCP proxies.

4 Managing Configurations

The first step toward creating a load-balanced service with Oracle Traffic Director is to create a configuration, which is a collection of metadata defining the run-time characteristics of an Oracle Traffic Director server. After creating a configuration, you can use it to create instances of Oracle Traffic Director servers on one or more administration nodes.

> **Note:** For the definitions of the Oracle Traffic Director terminology—*configuration*, *administration node*, and *instance*, see Section 1.4, "Oracle Traffic Director Terminology." For information about the relationship between configurations, administration nodes, and instances, see Figure 1–3 in Chapter 1, "Getting Started with Oracle Traffic Director."

This chapter contains the following topics:

- Creating a Configuration
- Viewing a List of Configurations
- Deploying a Configuration
- Modifying a Configuration
- Synchronizing Configurations Between the Administration Server and Nodes
- Copying a Configuration
- Deleting a Configuration
- Viewing a List of Configuration Backups
- Restoring a Configuration from a Backup

4.1 Creating a Configuration

You can create configurations by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Before You Begin

Before you begin creating a configuration, decide the following:

- A unique name for the configuration. Choose the name carefully; after creating a configuration, you cannot change its name.
- The user ID with which the Oracle Traffic Director instances of the configuration should run.

  Note: The server user that you specify for a configuration must meet the following requirements:

  - When the administration server is running as `root`, the server user of a configuration must either be `root` or belong to the same group as the user that installed Oracle Traffic Director.
  - When the administration server is running as a non-`root` user, the server user of a configuration must be the same as the administration server's server user.

  Note that the nodes to which a configuration is deployed must be homogenous in terms of the user accounts and groups configured on those systems.

- A unique listener `host:port` combination for the default virtual server that you will create as part of the configuration.
- `host:port` addresses of the servers in the origin-server pool that you will create as part of the configuration.
- (optional) Host names of the administration nodes on which you want to create instances of the configuration.

  Note: While creating a configuration by using the New Configuration wizard, you can choose to also instantiate the configuration on one or more administration nodes. The wizard enables you to do this by displaying the host names of the administration nodes that are registered with the administration server.

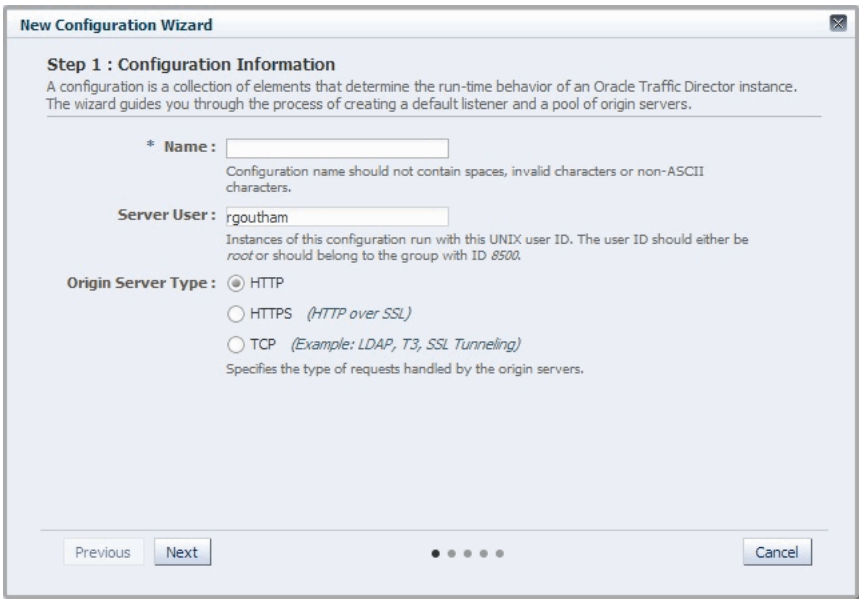
Creating a Configuration Using the Administration Console

To create a configuration by using the administration console, do the following tasks:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. In the Common Tasks pane, click **New Configuration**.

   The New Configuration wizard starts.

Figure 4–1 New Configuration Wizard



New Configuration wizard

****

3. Follow the on-screen prompts to complete creation of the configuration by using the details—listener port, origin-server addresses, and so on—that you decided earlier.

   After the configuration is created, the Results screen of the New Configuration wizard displays a message confirming successful creation of the configuration. If you chose to create instances of the configuration, then a message confirming successful creation of the instances is also displayed.

4. Click **Close** on the Results screen.

   In the New Configuration wizard, if you chose not to create an instance of the configuration, the message **Undeployed Configuration** is displayed, indicating that the configuration that you just created is yet to be deployed.

Creating a Configuration Using the CLI

To create a configuration, run the `create-config` command.

For example, the following command creates a configuration named `soa.example.com` with the virtual server and port `soa-app.example.com:1905` and two origin servers, `soa-1.example.com:80` and `soa-2.example.com:80`.

```
tadm> create-config --listener-port=1905 --server-name=soa-app.example.com
 --origin-server=soa-1.example.com:80,soa-2.example.com:80 soa.example.com
```

For more information about `create-config`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

4.2 Viewing a List of Configurations

At any time, you can view a list of the available configurations by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

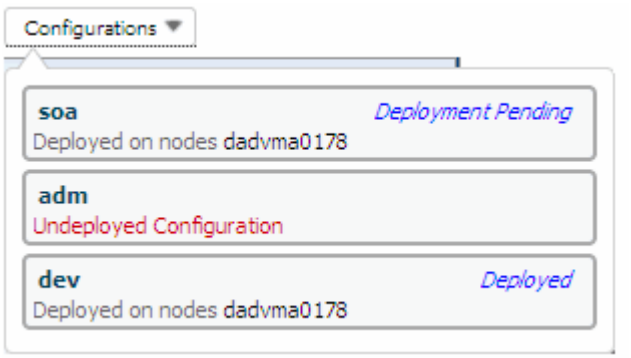
Viewing a List of Configurations Using the Administration Console

To view a list of the available configurations by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available Configurations is displayed, as shown in Figure 4–2.

Figure 4–2 List of Configurations



List of configurations

****

You can view the properties of a configuration by clicking on its name.

Viewing a List of Configurations Using the CLI

To view a list of the available configurations, run the `list-configs` command, as shown in the following example:

```
tadm> list-configs --verbose --all
config-name      deployment-status
---------------------------------
soa  deploy-pending
adm  undeployed
dev  deployed
org  instance-modified
```

Deployment Statuses

A configuration can be in one of the following statuses:

- **Deployed**: One or more instances of the configuration exist. All of the instances have the updated configuration settings.
- **Deployment pending**: One or more instances of the configuration exist. But the instances do not have the latest configuration settings.
- **Undeployed**: No instances exist for the configuration.

- **Instance modified**: The settings of one or more instances of the configuration have been manually modified.

4.3 Deploying a Configuration

You deploy a configuration to either create an instance of it on an administration node or update a previously created instance with new configuration settings. When you deploy a configuration, the running instances are reconfigured to reflect the configuration changes.

> **Note:** Certain configuration changes cannot be applied dynamically without restarting the instances. For the configuration changes that require instances to be restarted, the administration interfaces—CLI and administration console—display a prompt to restart the instances.

You can deploy a configuration by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Deploying a Configuration Using the Administration Console

To deploy a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration that you want to deploy.

   If there is no instance yet of the configuration that you selected to deploy, the message **Undeployed Configuration** is displayed at the top of the main pane. For the procedure to create one or more instances of an undeployed configuration, see Section 5.1, "Creating Oracle Traffic Director Instances."

   If instances of the configuration exist, but do not have the latest configuration settings, the message **Deployment Pending** is displayed at the top of the main pane. To update the instances with the latest configuration settings, do the following:

   a. Click **Deploy Changes**.

      A prompt to confirm deployment is displayed.

   b. Click **Deploy**.

      A message is displayed confirming that the updated configuration was successfully deployed.

   c. Click **Close**.

Deploying a Configuration Using the CLI

To deploy a configuration, run the `deploy-config` command.

For example, the following command updates all instances of the configuration `soa.example.com` with the latest configuration settings.

```
tadm> deploy-config soa.example.com
OTD-70201 Command 'deploy-config' ran successfully.
```

For more information about `deploy-config`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

After deploying an updated configuration, for the changes to take effect, you should restart the instance.

Note: For some parameters, you can reconfigure an instance without restarting it. For more information, see Section 5.4, "Updating Oracle Traffic Director Instances Without Restarting."

4.4 Modifying a Configuration

After you create a configuration and create instances from it, you might need to change some of the settings—log preferences, performance parameters, virtual server listener, origin-server pools, and so on.

You can modify a configuration by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Modifying a Configuration Using the Administration Console

To modify a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration that you want to modify.
4. In the navigation pane, you can select the following additional categories of settings for the configuration. The parameters relevant to the selected category are displayed on the main pane.
   - **SSL**
     - Settings for PKCS#11 Tokens. For more information, see Section 11.5, "Managing PKCS#11 Tokens."
     - Schedule and manage CRL-update events. For more information, see Section 11.6.2, "Installing CRLs Automatically."
     - SSL/TLS caching preferences. For more information, see Section 15.6.1, "SSL/TLS Session Caching."
   - **Logging**
     - Set and change parameters for the server log file—name and location of the log file, log level, date format, and so on.

  - Enable and disable the access log.
  - Set and change parameters for the access log file—name and location of the log file and log format
  - Schedule and manage events to rotate the server and access log files.
  - Configure access-log buffer settings to tune performance.

  For more information, see Chapter 12, "Managing Logs."

- **Advanced Settings**
  - Specify general settings: the server user ID, the temporary directory in which the process ID and socket information for the instances of the configuration are stored, and the localization preferences.
  - Configure DNS lookup and cache settings.

    For more information, see Section 15.5, "Tuning DNS Caching Settings."
  - Create, enable, disable, view, delete events for the configuration. For more information, see Section 5.6, "Controlling Oracle Traffic Director Instances Through Scheduled Events."
  - View a list of available backups for the configuration and restore from a backup configuration. For more information, see Section 4.9, "Restoring a Configuration from a Backup."
- **HTTP**, under **Advanced Settings**: Set and change parameters to tune the performance of the virtual servers defined for the configuration—such as, request buffer size, response buffer size, timeout thresholds for the request body and header, thread-pool settings, and keep-alive settings.

  For more information, see Section 15.4, "Tuning HTTP Request and Response Limits."
- **Monitoring**, under **Advanced Settings**
  - Enable and disable statistics collection, profiling, and the SNMP subagent.
  - Specify the statistics-collection interval.

  For more information, see Chapter 13, "Monitoring Oracle Traffic Director Instances."

Note: For information about modifying origin servers, origin-server pools, listeners, and virtual servers, see:

- Section 6.3, "Modifying an Origin-Server Pool"
- Section 7.3, "Modifying an Origin Server"
- Section 8.3, "Modifying a Virtual Server"
- Section 10.3, "Modifying a Listener"

---

5. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

6. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

> **Caution:** In the Advanced Settings page, if you change the **Temporary Directory** value, you should first stop all the instances of the configuration, deploy the changes, and then start the instances.
>
> If you deploy the changes without stopping the running instances, an error would occur when you attempt to stop the instances later. For information about solving this problem, see Section 16.2.8, "Unable to stop instance after changing the temporary directory."

Modifying a Configuration Using the CLI

The CLI provides several commands (see Table 4–1) that you can use to change specific parameters of a configuration.

> **Note:** For information about the CLI commands to change the properties of virtual servers, listeners, origin server pools, and origin servers in a configuration, see the following chapters:
>
> - Chapter 6, "Managing Origin-Server Pools"
> - Chapter 7, "Managing Origin Servers"
> - Chapter 8, "Managing Virtual Servers"
> - Chapter 10, "Managing Listeners"

Table 4–1 CLI Commands for Modifying a Configuration

Task	CLI Commands
Change the server user and temporary directory	`set-config-prop`
Change access-log buffer properties	`set-access-log-buffer-prop` `get-access-log-buffer-prop`
Change caching properties	`set-cache-prop` `get-cache-prop`
Change DNS properties	`set-dns-prop` `get-dns-prop`
Change DNS caching properties	`set-dns-cache-prop` `get-dns-cache-prop`
Change HTTP request properties	`set-http-prop` `get-http-prop`
Change keep-alive settings for client connections	`set-keep-alive-prop` `get-keep-alive-prop`

Table 4–1 (Cont.) CLI Commands for Modifying a Configuration

Task	CLI Commands
Change the default language	`set-localization-prop` `get-localization-prop`
Change error log settings	`set-log-prop` `get-log-prop`
Change PKCS #11 encryption settings	`set-pkcs11-prop` `get-pkcs11-prop`
Change QoS settings	`set-qos-prop` `get-qos-prop`
Enable SNMP	`set-snmp-prop` `set-snmp-prop`
Change SSL/TLS session caching properties	`set-ssl-session-cache-prop` `get-ssl-session-cache-prop`
Change statistics collection properties	`set-stats-prop` `get-stats-prop`
Change HTTP thread pool properties	`set-thread-pool-prop` `get-thread-pool-prop`

For example, the following command changes the location of the error log file for the configuration `soa` to `/home/log/errors.log`.

```
tadm> set-log-prop --config=soa log-file=/home/log/errors.log
OTD-70201 Command 'set-log-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

4.5 Synchronizing Configurations Between the Administration Server and Nodes

When you create a configuration, the configuration files—`server.xml` and *`vs_name`*`-obj.conf`—are created in the configuration store on the administration server.

When you create instances of the configuration, the configuration files are copied from the configuration store to the instance-specific configuration directories on the nodes, as in the following examples:

`INSTANCE_HOME``/net-`*`soa.example.com`*`/config`
`INSTANCE_HOME``/net-`*`dev.example.com`*`/config`

So the configuration files in the instance-specific configuration directories are usually identical to the configuration files stored in the configuration store on the administration server. In the following situations, a configuration stored on the administration server can be different from that of its instances.

- You modified a configuration on the administration server, by using the administration console or CLI, but have not yet deployed the modified configuration to its instances.

  You can rectify this out-of-sync situation by deploying the configuration as described in Section 4.3, "Deploying a Configuration."

- You changed the configuration of an instance manually, by editing a configuration file directly in the instance's `config` directory.

If you change the configuration of an instance manually by editing a configuration file—`server.xml` or *`vs_name`*`-obj.conf` in the `INSTANCE_HOME/net-`*`config_name`*`/config` directory, the next time you log in to the administration console and view the configuration, the alert **Instance Configuration Modified** is displayed. The alert indicates that the configuration stored on the administration server is different from the current configuration settings of one or more instances. The alert continues to be displayed till you synchronize the configuration stored on the administration server with that of all its instances, by doing one of the following:

- *Pull* the modified configuration from one of the modified instances back into the configuration store of the administration server.
- Discard the instance-specific configurations by redeploying the configuration from the administration server to the modified instances.

You can synchronize a configuration on the administration server with its instances by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Synchronizing Configurations on the Administration Server and Administration Nodes Using the Administration Console

To synchronize a configuration stored on the administration server with that of its instances by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which one or more instances have been modified. You can identify the configuration from its status, which would be Instance Configuration Modified.

   The Instances page of the configuration is displayed, with the **Instance Configuration Modified** button at the top of the main pane.
4. Click the **Instance Configuration Modified** button.

   The Deploy Configuration dialog box is displayed.

   - If you want to discard all of the instance-specific configurations and deploy the configuration that is currently stored on the administration server to all the instances, select the **Discard Instance Changes** option.

- If you want to pull the configuration of a modified instance to the administration server, select **Pull and Deploy Configuration from Node**, and select the appropriate administration node.

  For each administration node, you can review the names of the configuration files that are different from the configuration store on the administration server, by clicking **View Details**.

5. Click **OK**.

   A message is displayed confirming that the configuration was successfully deployed.

6. Click **Close**.

Synchronizing Configurations on the Administration Server and Administration Nodes Using the CLI

To discard the instance-specific configurations and deploy the configuration that is currently stored on the administration server to all the instances, run the following command:

```
tadm> deploy-config -- force config_name
```

To pull configuration files from an instance to the configuration store on the administration server, run the following commands:

```
tadm> pull-config --config=config_name node

tadm> deploy-config config_name
```

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

4.6 Copying a Configuration

When you want to create a configuration that is similar to an existing configuration, you can copy the existing configuration and make the required changes later.

You can copy a configuration by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Copying a Configuration Using the Administration Console

To copy a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration that you want to copy.
4. In the Common Tasks pane, click **Duplicate Configuration**.

5. In the resulting dialog box, enter a name for the new configuration, and then click **Duplicate**

   A message is displayed confirming that the configuration was copied.

6. Click **Close**.

Copying a Configuration Using the CLI

To copy a configuration, run the `copy-config` command.

For example, the following command copies the configuration `soa.example.com` to a new configuration named `soa2.example.com`.

```
tadm> copy-config --config=soa.example.com soa2.example.com
OTD-70201 Command 'copy-config' ran successfully.
```

For more information about `copy-config`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

4.7 Deleting a Configuration

You can delete a configuration by using either the administration console or the CLI.

Note:

- To delete a configuration that has one or more failover groups, you should first delete the failover groups. For more information, see Section 14.2.4, "Managing Failover Groups."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Deleting a Configuration Using the Administration Console

To delete a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration that you want to delete.
4. In the Common Tasks pane, click **Delete Configuration**.
   - If there are no instances of the configuration that you want to delete, a prompt to confirm deletion of the configuration is displayed.
     a. Click **Delete**.

        A message is displayed confirming that the configuration was deleted.

     b. Click **Close**.
   - If there are instances of the configuration that you want to delete, a dialog box is displayed listing the administration nodes on which the configuration is deployed. The list also indicates whether the instances are running.

a. If you want to proceed with the deletion, you can choose to save the log files of the instances by selecting the **Save Instance Logs** check box.

   To confirm deletion, click **Delete**.

   A message is displayed confirming that the configuration and its instances were deleted.

b. Click **Close**.

Note: If you selected the **Save Instance Logs** check box, the server access and error logs for the instances that were deleted are retained in the *INSTANCE_HOME*/net-*config_name*/logs directory.

---

5. Click the **Delete** button corresponding to the configuration that you want to delete.

Deleting a Configuration Using the CLI

Note: You cannot delete a configuration by using the CLI if instances of the configuration are deployed to administration nodes, regardless of whether the instances are running or stopped.

To delete such a configuration by using the CLI, you must first delete all of its instances.

To delete a configuration, run the `delete-config` command, as shown in the following example:

```
tadm> delete-config --config=soa.example.com
OTD-70201 Command 'delete-config' ran successfully.
```

For more information about `delete-config`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

4.8 Viewing a List of Configuration Backups

When you redeploy a modified configuration to its instances, a backup of the previous configuration is stored in a zip file in the configuration store on the administration server.

You can view a list of the backups of a configuration by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing a List of Configuration Backups Using the CLI

To view a list of the backups of a configuration by using the CLI, run the following command:

```
tadm> list-backups --config=config_name --verbose --all
```

The following is an example of the output of the `list-backups` command.

```
backup-id       backup-date
--------------------------
20110712_025354 "Jul 12, 2011 2:53:54 AM"
20110712_024410 "Jul 12, 2011 2:44:10 AM"
20110712_004743 "Jul 12, 2011 12:47:43 AM"
20110711_231826 "Jul 11, 2011 11:18:26 PM
```

As shown in the example output, each backup configuration has a unique ID, which is assigned automatically when it is created. The ID indicates the date and time when the backup was created.

Viewing a List of Configuration Backups Using the Administration Console

To view a list of the backups of a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view a list of backups.
4. In the navigation pane, expand **Configuration Settings**, and select **Backups**.

   The Configuration Backups page is displayed. It lists the dates when backups of the configuration were created.

4.9 Restoring a Configuration from a Backup

When you redeploy a modified configuration to its instances, a backup of the previous configuration is created. For information about viewing a list of the available backups, see Section 4.8, "Viewing a List of Configuration Backups."

You can restore a configuration from a backup by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Restoring a Configuration from a Backup Using the Administration Console

To restore a configuration from a backup, by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view a list of backups.
4. In the navigation pane, select **Advanced Settings**.

   The Advanced Settings page is displayed.

5. Scroll down to the Configuration Backups section of the page.
6. Identify the backup that you want to restore, and click the icon displayed in the **Restore** column.

Restoring a Configuration from a Backup Using the CLI

To restore a configuration from a backup by using the CLI, run the following command:

```
tadm> restore-config --config=config_name --verbose --all
```

5 Managing Instances

An instance is an Oracle Traffic Director server running on an administration node, or on the administration server, and listening on one or more ports for requests from clients.

This chapter contains the following sections:

5.1 Creating Oracle Traffic Director Instances

You can create Oracle Traffic Director instances of a configuration by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Prerequisites for Creating Oracle Traffic Director Instances

To be able to create an instance, you should have done the following:

- Defined a configuration (see Section 4.1, "Creating a Configuration").
- Created one or more administration nodes (see Section 3.1, "Creating an Administration Node"). Note that you *can* create instances on the administration server as well.

Creating Oracle Traffic Director Instances Using the Administration Console

To create Oracle Traffic Director instances of a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

A list of the available configurations is displayed.

3. Select the configuration for which you want to create an instance.
4. In the Common Tasks pane, click **New Instance**.

   The New Instance wizard is displayed. The wizard lists the available administration nodes.

   Note:

   - For a host to be listed as an available administration node, it should be designated as an administration node as described in Section 3.1, "Creating an Administration Node"
   - On an administration node, you can create only one instance of a particular configuration. So if an instance of the configuration that you are trying to deploy already exists on the administration node, the node is not displayed.

5. Select the check boxes corresponding to the administration nodes on which you want to create instances of the configuration. Then, click **Next**.
6. On the resulting screen of the wizard, review the list of administration nodes that you selected. Then, click **Create Instance**.

   A message is displayed confirming the successful creation of the instance.
7. Click **Close**.

   The Instances page is displayed, showing the instance that you just created.

Creating an Oracle Traffic Director Instance Using the CLI

To create one or more Oracle Traffic Director instances, run the `create-instance` command.

For example, the following command creates an instance of the configuration named `soa` on each of the nodes, `apps2` and `apps2`.

```
tadm> create-instance --config=soa apps1 apps2
OTD-70201 Command 'create-instance' ran successfully.
```

For more information about `create-instance`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

For each Oracle Traffic Director configuration that you instantiate on an administration node, a subdirectory named `net-`*`config_name`* is created in the `INSTANCE_HOME` subdirectory.

5.2 Viewing a List of Oracle Traffic Director Instances

You can view a list of Oracle Traffic Director instances by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing a List of Oracle Traffic Director Instances Using the Administration Console

To view a list of the Oracle Traffic Director instances of a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view instances.

   The Instances page is displayed, showing the instances of the configuration, as shown in Figure 5–1.

Figure 5–1 List of Instances

List of instances

****

You can view the properties of an instance by clicking on its name.

Viewing a List of Oracle Traffic Director Instances Using the CLI

To view a list of the Oracle Traffic Director instances of a configuration, run the `list-instances` command, as shown in the following example:

```
tadm> list-instances --config=soa.example.com --verbose --all

node-name            instance-status   has-service
-------------------------------------------------
soa.example.com      started           false
adf.example.com      stopped           false
```

5.3 Starting, Stopping, and Restarting Oracle Traffic Director Instances

You can start, stop, and restart Oracle Traffic Director instances by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Starting, Stopping, and Restarting Oracle Traffic Director Instances Using the Administration Console

To start, stop, or restart Oracle Traffic Director instances by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to start, stop, or restart instances.
4. In the navigation pane, select **Instances**.
5. Click the **Start/Restart** or **Stop** button, as required, for the instance that you want to start, restart, or stop.

   Note: To start or restart *all* instances of the selected configuration, click **Start/Restart Instances** in the Common Tasks pane. To stop all instances of the configuration, click **Stop Instances**.
6. If a PKCS#11 token that provides the interface to the certificates database is protected with a pin (see Section 11.5, "Managing PKCS#11 Tokens"), when you click **Start/Restart** or **Start/Restart Instances**, a prompt to enter the token pin is displayed. To proceed with starting the instances, you should enter the pins for the tokens that are protected with pins.

   A message is displayed in the Console Messages pane confirming that the instances were started, stopped, or restarted.

Starting, Stopping, and Restarting Oracle Traffic Director Instances Using the CLI

To start, stop, or restart one or more Oracle Traffic Director instances of a configuration, run the `start-instance`, `stop-instance`, or `restart-instance` command.

For example, the following three commands start, restart, and stop the instances of the configuration `soa` on the nodes `apps1.example.com` and `apps2.example.com`.

```
tadm> start-instance --config=soa apps1.example.com apps2.example.com

tadm> restart-instance --config=soa apps1.example.com apps2.example.com

tadm> stop-instance --config=soa apps1.example.com apps2.example.com
```

If a PKCS#11 token that provides the interface to the certificates database is protected with a pin (see Section 11.5, "Managing PKCS#11 Tokens"), when you run `start-instance`, a prompt to enter the token pin is displayed, as shown in the following example:

```
Enter password for token "internal">
```

To proceed with starting the instances, you should enter the pins.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

Note: Alternatively, you can use the following commands from within the instance directory to start, restart, and stop the instances:

```
> $INSTANCE_HOME/net-config_name/bin/startserv

> $INSTANCE_HOME/net-config_name/bin/restart

> $INSTANCE_HOME/net-config_name/bin/stopserv
```

5.4 Updating Oracle Traffic Director Instances Without Restarting

When you make changes to some configuration parameters, the running Oracle Traffic Director instances of the configuration need not be restarted for the changes in the configuration to take effect. You can dynamically *reconfigure* the Oracle Traffic Director instances to reflect the new configuration.

For a list of the parameters that support dynamic reconfiguration, see "Dynamic Reconfiguration" in the *Oracle Traffic Director Configuration Files Reference*.

You can dynamically reconfigure the running instances of a configuration by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Reconfiguring an Oracle Traffic Director Instance Using the Administration Console

To reconfigure an Oracle Traffic Director instance by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to reconfigure instances.
4. In the navigation pane, select **Instances**.
5. Click the **Reconfigure** button for the instance that you want to update dynamically.

   A message is displayed in the Console Messages pane confirming that the instance was reconfigured.

Reconfiguring Oracle Traffic Director Instances Using the CLI

To reconfigure one or more Oracle Traffic Director instances of a configuration, run the `reconfig-instance` command.

For example, the following command reconfigures the instances of the configuration `soa` on the nodes `apps1` and `apps2`.

```
tadm> reconfig-instance --config=soa apps1 apps2
```

For more information about `reconfig-instance`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Note: Alternatively, you can use the following command from within the instance directory:

```
> $INSTANCE_HOME/net-config_name/bin/reconfig
```

5.5 Deleting Oracle Traffic Director Instances

You can delete instances of a configuration by using either the administration console or the CLI.

Deleting an Oracle Traffic Director Instance Using the Administration Console

To delete an Oracle Traffic Director instance by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to delete instances.
4. In the navigation pane, select **Instances**.
5. Click the **Delete** button for the instance that you want to delete.

Note: To delete an instance that is part of a failover group, you should first remove the instance from the failover group. For more information, see Section 14.2.4, "Managing Failover Groups."

A message is displayed in the Console Messages pane confirming that the instance was deleted.

Deleting Oracle Traffic Director Instances Using the CLI

To delete Oracle Traffic Director instances of a configuration, run the `delete-instance` command.

For example, the following command deletes the instance of the configuration `soa` running on nodes `apps1` and `apps2`.

```
tadm> delete-instance --config=soa apps1 apps2
```

Note: You can delete an instance that is part of a failover group by using the `--force` option, but if you do so, all the failover groups to which the instance belongs will also be deleted.

For more information about `delete-instance`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

5.6 Controlling Oracle Traffic Director Instances Through Scheduled Events

As an administrator, if you have to manage a large number of configurations and their instances, repetitive tasks such as restarting and reconfiguring instances of each configuration individually can become tedious. You can schedule *events* for administrative tasks to be performed automatically at defined intervals; or on specific days of the week, times of the day, or dates of the month.

You can create and manage events by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Managing Events Using the Administration Console

To manage events by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to schedule events.
4. In the navigation pane, select **Advanced Settings**.

   The Advanced Settings page is displayed.
5. Scroll down to the Scheduled Events section of the page.

   It lists events that are currently scheduled for the configuration.

   - To enable or disable an event, select the **Enable/Disable** check box.
   - To delete an event, click the **Delete** icon.
   - To create an event, click **New Event**.

     The New Configuration Event dialog box is displayed.

     Select the event that you want to schedule, and specify the interval or time at which the event should be performed, and then click **OK**.

   A message, confirming the change, is displayed in the Console Messages pane.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Managing Events Using the CLI

- **Creating an event**

  To create an event, run the `create-event` command, as shown in the following examples.

```
tadm> create-event --config=soa --command=restart --interval=86400
```

```
tadm> create-event --config=soa --command=reconfigure --time=14:00
```

The first command schedules an event to restart all the instances of the configuration `soa` after every 86400 seconds.

The second command schedules an event to reconfigure all the instances of the configuration `soa` at 2 p.m. every day.

Note: For the scheduled events to take effect, you should redeploy the configuration.

- **Viewing a list of events**

  To view a list of scheduled events, run the `list-events` command.

  For example, the following command displays the events scheduled for instances of the configuration `soa`.

```
tadm> list-events --config=soa --verbose --all
command        enabled    day-of-month   month   day-of-week   time    interval
-------------------------------------------------------------------------------
restart        false      -              -       -             -       60
reconfig       true       -              -       -             -       60
```

- **Disabling an event**

  When you create an event, it is enabled automatically, unless you specified the `--no-enabled` option.

  To disable an event, run the `disable-event` command, as shown in the following example:

```
tadm> disable-event --config=soa --command=restart --interval=86400
```

- **Enabling an event**

  To enable an event, run the `enable-event` command, as shown in the following example:

```
tadm> enable-event --config=soa --command=restart --interval=86400
```

- **Deleting an event**

  To delete an event, run the `delete-event` command, as shown in the following example:

```
tadm> delete-event --config=soa --command=restart --interval=86400
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

6 Managing Origin-Server Pools

An *origin server* is a back-end server to which Oracle Traffic Director forwards requests that it receives from clients, and from which it receives responses to client requests. The origin servers could, for example, be Oracle WebLogic Server instances or Oracle iPlanet Web Server instances. A group of origin servers providing the same service or serving the same content is called an *origin-server pool*. You can define several such origin-server pools in a configuration, and then configure each virtual server in an Oracle Traffic Director instance to route client requests to a specific pool.

This chapter describes how to create and manage origin-server pools. It contains the following sections:

6.1 Creating an Origin-Server Pool

You can create an origin-server pool by using either the administration console or the CLI.

Note:

- When you create an origin-server pool, you are, in effect, modifying a configuration. So for the settings of the new origin-server pool to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Before You Begin

Before you begin creating an origin-server pool, decide the following:

- A unique name for the origin-server pool. Choose the name carefully; after creating an origin-server pool, you cannot change its name.

- `host:port` combinations for the servers in the origin-server pool.

  Note: If the origin servers for which you want to create a pool are Oracle WebLogic Server managed servers in a cluster, it is sufficient to create the pool with any *one* of the managed servers as the origin server. You can then configure Oracle Traffic Director to *discover* the other managed servers in the pool dynamically. For more information, see Section 6.5, "Configuring an Oracle WebLogic Server Cluster as an Origin-Server Pool."

- The communication protocol—HTTP, HTTPS or TCP—of the servers in the pool.
- The address family that the servers in the origin-server pool use to listen for requests.

  The supported address families are:

  - `inet` (IPv4)
  - `inet6` (IPv6)
  - `inet-sdp` (Sockets Direct Protocol): Select this family if the servers in the origin-server pool are on the InfiniBand fabric and listen on an SDP interface, such as Oracle WebLogic Servers deployed on Oracle Exalogic machines.

    Note: For Oracle Traffic Director to communicate with WebLogic Server over SDP, further configuration steps are required on the WebLogic Server. For more information about these configuration steps, see "Enabling Cluster-Level Session Replication Enhancements" in the *Oracle Fusion Middleware Exalogic Enterprise Deployment Guide*.

Creating an Origin-Server Pool Using the Administration Console

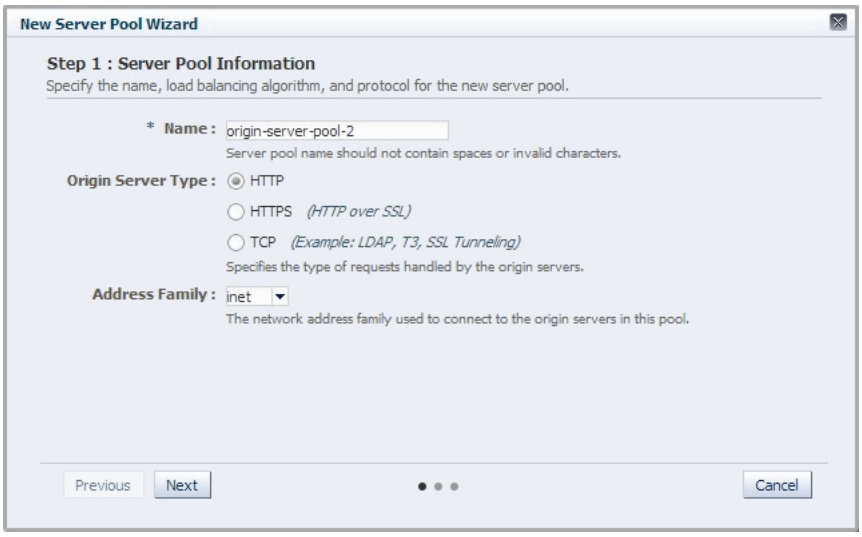
To create an origin-server pool by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create a virtual server.
4. In the Common Tasks pane, click **New Origin-Server Pool**.

   The New Origin-Server Pool wizard starts.

Figure 6–1 New Origin-Server Pool Wizard



New Origin-Server Pool wizard

****

5. Follow the on-screen prompts to complete creation of the origin-server pool by using the details—name, load balancing method, origin servers, and so on—that you decided earlier.

   After the origin-server pool is created, the Results screen of the New Origin-Server Pool wizard displays a message confirming successful creation of the origin-server pool.

6. Click **Close** on the Results screen.

   - The details of the origin-server pool that you just created are displayed on the Origin-Server Pools page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Creating an Origin-Server Pool Using the CLI

To create an origin-server pool, run the `create-origin-server-pool` command.

For example, the following command creates an origin-server pool `osp-soa` containing two origin servers `http://soa.example.com:1901` and `http://soa.example.com:1902` in the configuration `soa`.

```
tadm> create-origin-server-pool --config=soa --type=http
--origin-server=soa.example.com:1901,soa.example.com:1902 osp-soa
OTD-70201 Command 'create-origin-server-pool' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `create-origin-server-pool`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Specifying an HTTP Forward Proxy Server

The `create-origin-server-pool` command takes `proxy-server` as an optional option which you can use to specify a HTTP forward proxy server to be associated with an origin server pool so that all member origin servers of the pool are communicated with via the configured HTTP forward proxy server. The type must be `http` or `https`.

For example:

```
tadm> create-origin-server-pool --config=soa --type=http
--origin-server=soa.example.com:1901,soa.example.com:1902 osp-soa
--proxy-server=www.example.com
OTD-70201 Command 'create-origin-server-pool' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `create-origin-server-pool`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

6.2 Viewing a List of Origin-Server Pools

You can view a list of origin-server pools by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing a List of Origin-Server Pools Using the Administration Console

To view a list of origin-server pools by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view origin-server pools.
4. In the navigation pane, select **Origin-Server Pools**.

   The Origin-Server Pools page is displayed. It shows a list of the origin-server pools defined for the configuration.

You can view the properties of an origin-server pool in detail by clicking on its name.

Viewing a List of Origin-Server Pools Using the CLI

To view a list of origin-server pools, run the `list-origin-server-pools` command, as shown in the following example:

```
tadm> list-origin-server-pools --config=soa --verbose --all
name            type            load-distribution
------------------------------------------------
osp1            http            least-connection-count
osp2            http            round robin
osp3            https           least-connection-count
```

You can view the general properties and health-check settings of an origin-server pool by running the `get-origin-server-pool-prop` and `get-health-check-prop` command respectively.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

6.3 Modifying an Origin-Server Pool

You can change the properties of an origin-server pool by using either the administration console or the CLI.

Note:

- When you modify an origin-server pool, you are, in effect, modifying a configuration. So for the updated origin-server pool settings to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Changing the Properties of an Origin-Server Pool Using the Administration Console

To change the properties of an origin-server pool by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to modify origin-server pools.
4. In the navigation pane, select **Server Pools**.

   The Origin Server Pools page is displayed. It shows a list of the origin-server pools that are defined for the configuration.
5. Click the name of the origin-server pool that you want to modify.

   The Origin Server Pool Settings page is displayed. On this page, you can do the following:

   - Change the network protocol—IPv4, IPv6, or SDP—for the servers in the pool.
   - Set a proxy server via the **Connect to Origin Servers via Proxy Server** section. This setting specifies a HTTP forward proxy server to be associated with an origin server pool so that all member origin servers of the pool are communicated with via the configured HTTP forward proxy server.
   - Change the load-balancing method that Oracle Traffic Director should use to distribute client requests to the pool.

- **Least connection count** (default): When processing a request, Oracle Traffic Director assesses the number of connections that are currently active for each origin server, and forwards the request to the origin server with the least number of active connections.

  The least connection count method works on the premise that origin servers that are faster have fewer active connections, and so can take on more load. To further adjust the load distribution based on the capacities of the origin servers, you can assign relative weights to the origin servers.

Note: WebSocket connections affect the least connection count load balancing algorithm because WebSocket connections are potentially long lasting and will be counted as active connections until they are closed.

- **Least response time**: Though least connection count works well on most workloads, there could be situations when the response time of origin servers in a given pool for the same amount of load could differ. For example:

  - When origin servers of a given pool are deployed on machines that differ in hardware specification.

  - When some origin server nodes are used for other services.

  - When network connectivity for different nodes is not uniform or some network interfaces are more loaded than others.

  Least response time is useful in such scenarios because it is a dynamic weighted least connection algorithm and it calculates weights based on the response time. These weights are continuously adjusted based on how the origin servers respond. Least response time helps you avoid manual tuning of weights in the least connection algorithm.

- **Round robin**: Oracle Traffic Director forwards requests sequentially to the available origin servers—the first request to the first origin server in the pool, the second request to the next origin server, and so on. After it sends a request to the last origin server in the pool, it starts again with the first origin server.

  Though the round-robin method is simple, predictable, and low on processing overhead, it ignores differences in the origin servers' capabilities. So, over time, requests can accumulate at origin servers that are significantly slow. To overcome this problem, you can use a *weighted* round-robin method, by assigning relative weights to the origin servers.

For more information about assigning weights to origin servers, see Section 7.3, "Modifying an Origin Server."

- Configure health-check settings. For more information, see Section 14.3, "Configuring Health-Check Settings for Origin-Server Pools."
- Specify whether Oracle Traffic Director should dynamically discover Oracle WebLogic Server managed servers in a cluster. For more information, see Section 6.5, "Configuring an Oracle WebLogic Server Cluster as an Origin-Server Pool."

Note: You can add, modify, and remove origin servers in the pool, by selecting **Origin Servers** in the navigation pane. For more information, see Chapter 7, "Managing Origin Servers."

6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.

   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Changing the Properties of an Origin-Server Pool Using the CLI

- To change the network protocol and load-balancing method for an origin-server pool, run the `set-origin-server-pool-prop` command.

  For example, the following command changes the load-balancing method for the origin-server pool `osp1` in the configuration `soa` to the round-robin method.

  ```
  tadm> set-origin-server-pool-prop --config=soa --origin-server-pool=osp1
  load-distribution=round-robin
  OTD-70201 Command 'set-origin-server-pool-prop' ran successfully.
  ```

- To change the health-check parameters for an origin-server pool, run the `set-health-check-prop` command.

  For example, the following command changes the health-check ping interval for servers in the origin-server pool `osp1` of the configuration `soa` to 60 seconds.

  ```
  tadm> set-health-check-prop --config=soa --origin-server-pool=osp1 interval=60
  OTD-70201 Command 'set-origin-server-pool-prop' ran successfully.
  ```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For a list of the properties that you can set or change by using the `set-origin-server-pool-prop` and `set-health-check-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

6.4 Deleting an Origin-Server Pool

You can delete an origin-server pool by using either the administration console or the CLI.

Note:

- You cannot delete an origin-server pool that is associated with one or more routes in virtual servers.

  To delete an origin-server pool that is associated with routes, you must first delete the referring routes, as described in Section 8.4, "Configuring Routes."

- When you delete an origin-server pool, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."

- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Deleting an Origin-Server Pool Using the Administration Console

To delete an origin-server pool by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to delete origin-server pools.
4. In the navigation pane, select **Origin-Server Pools**.

   The Origin-Server Pools page is displayed.
5. Click the **Delete** icon for the origin-server pool that you want to delete.
   - If the origin-server pool is associated with one or more routes in virtual servers, a message is displayed indicating that you cannot delete the pool.
   - If the origin-server pool is not associated with any virtual server, a prompt to confirm the deletion is displayed.
6. Click **OK**.

   A message is displayed in the Console Message pane confirming that the origin-server pool was deleted.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Deleting an Origin-Server Pool Using the CLI

To delete an origin-server pool, run the `delete-origin-server-pool` command, as shown in the following example:

```
tadm> delete-origin-server-pool --config=soa osp1
OTD-70201 Command 'delete-origin-server-pool' ran successfully.
```

Note: If the specified origin-server pool is associated with one or more routes in virtual servers, the following error message is displayed:

```
OTD-67108 Cannot delete the origin-server pool. It is referred by
virtual server(s): vs1_name,vs1_name,[...]
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `delete-origin-server-pool`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

6.5 Configuring an Oracle WebLogic Server Cluster as an Origin-Server Pool

Note: Oracle Traffic Director has built-in support for some common functionality offered by the WebLogic Server plug-in. Hence Oracle Traffic Director does not require any other plug-in to inter-operate with WebLogic Server.

If you want to create an origin-server pool that represents a cluster of Oracle WebLogic Server managed servers, you need not specify each managed server in the cluster as an origin server. It is sufficient to specify *any one* of the managed servers as the sole origin server in the pool. You can configure Oracle Traffic Director to *discover* the presence of other Oracle WebLogic Server instances in the cluster dynamically, and distribute client requests to the managed server that is configured as an origin server *and* to the dynamically discovered managed servers in the same cluster.

So when dynamic discovery is enabled, if any of the managed servers in the cluster is stopped, added, or removed, you need not update the definition of the origin-server pool. However, for detecting changes in the Oracle WebLogic Server cluster, Oracle Traffic Director sends health-check requests at a specified interval, which causes some overhead.

6.5.1 How Dynamic Discovery Works

When dynamic discovery is enabled for an origin-server pool, Oracle Traffic Director discovers the remaining Oracle WebLogic Server managed servers in the cluster, by doing the following:

1. **When an Oracle Traffic Director instance starts**, it checks whether the origin servers specified in the pool are Oracle WebLogic Server managed servers and whether the servers belong to a cluster, by sending an HTTP health-check request to each configured origin server.

   The origin server's response indicates whether the server is an Oracle WebLogic Server managed server. If the origin server is an Oracle WebLogic Server managed server that belongs to a cluster, the response also includes a list of the managed servers in the cluster.

2. Oracle Traffic Director uses the information in the response from the origin server to update the configuration with the discovered managed servers.

The dynamically discovered origin servers inherit all of the properties—weight, maximum connections, and so on—that are specified for the configured origin server.

3. **Subsequently, at each health-check interval (default: 30 seconds) configured for the origin-server pool**, Oracle Traffic Director attempts to detect changes in the cluster, by sending dynamic-discovery health-check requests to the Oracle WebLogic Server instances that are configured as origin servers in the pool.

   If the response indicates a change—removal or addition of a managed server—in the cluster since the previous health check, Oracle Traffic Director updates the configuration with the new set of dynamically discovered origin servers.

Note:

- Dynamically discovered origin servers are not stored permanently in the origin-server pool definition of the instance's configuration. So when you restart an Oracle Traffic Director instance, the process of dynamic discovery starts afresh.
- The HTTP request type that Oracle Traffic Director sends for dynamic discovery is the health-check request type that is currently configured for the origin-server pool—`OPTIONS` (default) or `GET`. For more information, see Section 14.3, "Configuring Health-Check Settings for Origin-Server Pools."

6.5.2 Enabling Dynamic Discovery

When you create an origin-server pool, dynamic discovery of Oracle WebLogic Server managed servers in a cluster is *not* enabled by default. You can enable dynamic discovery by using either the administration console or the CLI.

Note:

- When you modify an origin-server pool, you are, in effect, modifying a configuration. So for the updated origin-server pool settings to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Enabling Dynamic Discovery Using the Administration Console

To enable dynamic discovery of WebLogic Server managed servers in a cluster by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to enable dynamic discovery.

4. In the navigation pane, expand **Server Pools** and select the origin-server pool for which you want to enable dynamic discovery.

   The Server Pool Settings page is displayed.

5. Go to the **Advanced Settings** section of the page.

6. Under the Health Check subsection, make sure that the **Protocol** is HTTP, select the **Dynamic Discovery** check box.

7. Click **Save**.

   Note: If the current health-check protocol is TCP, an error message is displayed indicating that the protocol must be changed to HTTP in order to enable dynamic discovery.

   A message is displayed in the Console Message pane confirming that the updated health-check settings were saved.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Enabling Dynamic Discovery Using the CLI

To enable dynamic discovery of Oracle WebLogic Server managed servers in a cluster, run the `set-health-check-prop` command.

For example, the following command enables dynamic discovery of managed servers in the Oracle WebLogic Server cluster that the `wls-1` origin-server pool represents.

```
tadm> set-health-check-prop --config=soa.example.com --origin-server-pool=wls-1
dynamic-server-discovery=true
OTD-70201 Command 'set-health-check-prop' ran successfully.
```

Note: If the current health-check protocol is TCP, an error message is displayed indicating that the protocol must be changed to HTTP in order to enable dynamic discovery.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `set-health-check-prop`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

6.6 Configuring a Custom Maintenance Page

Oracle Traffic Director allows you to serve a custom server pool maintenance response code, and HTML page, when all the back-end servers are detected offline. Providing this type of message is better than having a gateway time-out, or creating other resources to host static content.

When maintenance is enabled for an origin server pool, then:

- requests landing on it are aborted with a 503 response code, if both response-code and response-file are not configured.

- requests landing on it are aborted with response-code value as the response code, if only response-code is specified.
- requests landing on it are not aborted, but are responded to with a response-file content and response-code value as the response code, if both are specified.
- running of a health-check on its origin servers is disabled.

When maintenance is not enabled for an origin server pool but no origin servers are configured or enabled, then:

- requests landing on it are aborted with a 503 response code.
- running of a health-check on its origin servers is disabled.

Monitoring of Statistics for Origin Server Pool in Maintenance

If the origin-server pool is in a maintenance state, there will be no statistics for the origin server pool and the origin servers belonging to the pool. Statistics will be available only for active origin server pools and active origin servers.

Enabling or Disabling Maintenance for an Origin-Server Pool Using the CLI

To enable maintenance for an origin-server pool, run the `enable-maintenance` command.

For example, the following command enables maintenance for the `http-pool-1` origin-server pool, and specifies a response-code of 505. This command takes `response-code` and `response-file` as optional properties. A response-code of 200 is not allowed without a response-file.

```
tadm> enable-maintenance --config=test --origin-server-pool=http-pool-1
--response-code=505
```

To disable maintenance, use the `disable-maintenance` command:

```
tadm> disable-maintenance --config=test --origin-server-pool=http-pool-1
```

To return the enabled, response-file and response-code properties for the origin-server pool, use the `get-maintenance-prop` command:

```
tadm> get-maintenance-prop --config=test --origin-server-pool=http-pool-1
```

For information about `enable-maintenance`, `disable-maintenance`, and `get-maintenance-prop`, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

7

Managing Origin Servers

An *origin server* is a back-end server to which Oracle Traffic Director forwards requests that it receives from clients, and from which it receives responses to client requests. The origin servers could, for example, be Oracle WebLogic Server instances or Oracle iPlanet Web Server instances. A group of origin servers providing the same service is called an *origin server pool*.

This chapter describes how to create and manage origin servers. It contains the following sections:

7.1 Adding an Origin Server to a Pool

You can add an origin server to an origin-server pool by using either the administration console or the CLI.

Note:

- When you add an origin server to a pool, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Before You Begin

Before you begin adding an origin server to a pool, decide the following:

- The origin-server pool to which you want to add the origin server.
- The **host** name or IP address of the origin server. It is recommended that the IP address that you provide is the InfiniBand interface IP address (IPoIB) or Socket Director Protocol (SDP) address.

> **Note:** SDP is a native Infiniband protocol. With SDP, performance is very specific to work load. Hence, it is important to evaluate and compare the performance with SDP and IPoIB, and then select the one that meets your requirement.

- The **port** number at which the origin server listens for requests.
- Whether the server is a **backup** origin server.

  Oracle Traffic Director forwards requests to a backup origin server only when the health check indicates that none of the primary origin servers is available.
- The proportion of the total request load that Oracle Traffic Director should distribute to the origin server. You define this proportion as a **weight** number that is relative to the weights assigned to the other origin servers in the pool.

  You can use weights to get Oracle Traffic Director to distribute the request load based on the relative capacities of the origin servers in a pool.

  Consider a pool consisting of three origin servers—`os1`, `os2`, and `os3`, with the weights 1, 2, and 2 respectively. The total of the weights assigned to all the servers in the pool is 1+2+2=5. Oracle Traffic Director distributes a fifth (1/5) of the total load to `os1`, and two-fifths (2/5) of the load to each of `os2` and `os3`.

Adding an Origin Server to a Pool Using the Administration Console

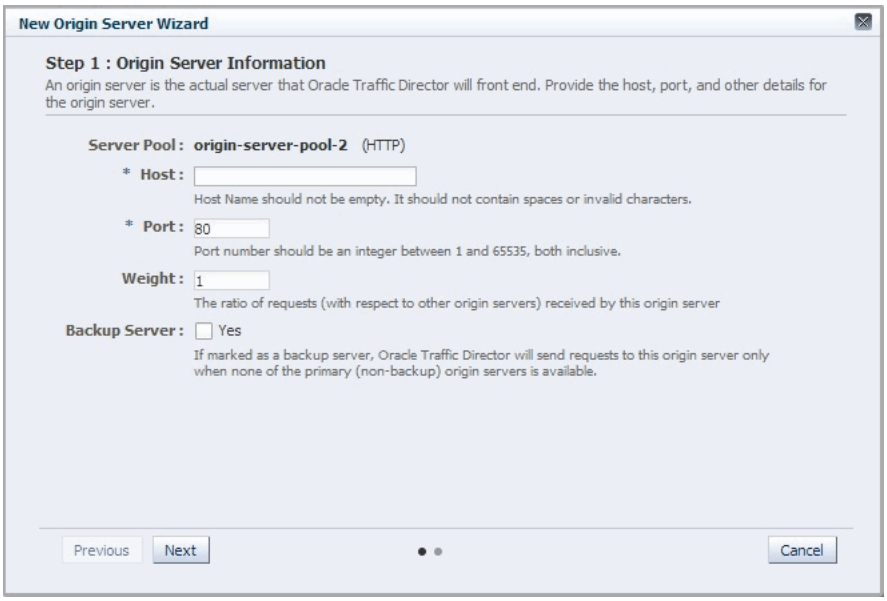
To add an origin server to a pool by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create an origin server.
4. In the Common Tasks pane, click **New Origin Server**.

   The New Origin Server wizard starts.

Figure 7–1 New Origin Server Wizard



New Origin Server wizard

****

5. Follow the on-screen prompts to complete creation of the origin-server pool by using the details—origin-server pool, host, port, and so on—that you decided earlier.

   After the origin server is created, the Results screen of the New Origin Server wizard displays a message confirming successful creation of the origin server.

6. Click **Close** on the Results screen.

   - The details of the origin server that you just defined are displayed on the Origin Servers page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Adding an Origin Server to a Pool Using the CLI

To add an origin server to a pool, run the `create-origin-server` command.

For example, the following commands adds `soa-app.example.com:80` as origin server `os1` in the pool `osp1` of the configuration `soa`.

```
tadm> create-origin-server --config=soa --origin-server-pool=osp1
soa-app.example.com:80
OTD-70201 Command 'create-origin-server' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `create-origin-server`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

7.2 Viewing a List of Origin Servers

You can view a list of origin servers by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing a List of Origin Servers Using the Administration Console

To view a list of origin servers by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view origin servers.
4. In the navigation pane, expand **Origin-Server Pools** and select **Origin Servers**.

   The Origin Servers page is displayed. It shows a list of the origin servers in the selected pool.

You can view and edit the properties of an origin server by clicking on its name.

Viewing a List of Origin Servers Using the CLI

To view a list of origin servers, run the `list-origin-servers` command as shown in the following example:

```
tadm> list-origin-servers --config=soa --origin-server-pool=osp1 --verbose --all
name                      weight          enabled         backup
-----------------------------------------------------------------
soa-app1.example.com:80   1               true            false
soa-app2.example.com:80   1               true            false
soa-app3.example.com:80   1               true            true
```

You can view the properties of an origin server in detail by running the `get-origin-server-prop` command.

For more information about the `list-origin-servers` and `get-origin-server-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

7.3 Modifying an Origin Server

This section describes how you can do the following:

- Change the properties—host, port, weight, and so on—that you defined while creating the origin server. For more information about those properties, see the "Before You Begin" section.
- Enable or disable the origin server.
- Specify the maximum number of connections that the origin server can handle concurrently.

- Specify the duration (ramp-up time) over which Oracle Traffic Director should increase the request-sending rate to the origin server. You can use this parameter to ensure that the request load, on origin servers that have just come up after being offline, is increased *gradually* up to the capacity of the server.

You can change the properties of an origin server by using either the administration console or the CLI.

Note:

- When you change the properties of an origin server in a pool, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Changing the Properties of an Origin Server Using the Administration Console

To change the properties of an origin server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to modify an origin server.
4. In the navigation pane, expand **Origin-Server Pools**, and select **Origin Servers**.

   The Origin Servers page is displayed.
5. Click the name of the origin server that you want to modify.

   The Editing Origin Server dialog box is displayed. In this dialog box, you can do the following:

   - Enable and disable the origin server
   - Change the host and port
   - Change the relative weight
   - Mark the origin server as a backup server
   - Set the maximum number of connections that the origin server can handle concurrently
   - Set the time that Oracle Traffic Director should take to ramp up the request-forwarding rate to the full capacity of the origin server.
6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Changing the Properties of an Origin Server Using the CLI

To change the properties of an origin server, run the `set-origin-server-prop` command.

For example, the following command changes the relative weight to 2 for the origin server `soa-app1.example.com:1900` in the pool `osp1` of the configuration `soa`.

```
tadm> set-origin-server-prop  --config=soa --origin-server-pool=osp1
 --origin-server=soa-app1.example.com:1900 weight=2
OTD-70201 Command 'set-origin-server-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For a list of the properties that you can change by using `set-origin-server-prop`, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

7.4 Managing Ephemeral Ports

In a topology that includes a client, OTD and Oracle WebLogic Server (WLS), OTD receives external requests at the configured HTTP listener port. OTD then opens up another connection while communicating and proxying the request to the WLS/origin server.

As part of this connection, OTD leverages *ephemeral ports* so that WLS/origin server can send data back to OTD. An ephemeral port is a short-lived transport protocol port for Internet Protocol (IP) communications allocated automatically from a predefined range by the IP software. In Linux, you can limit or restrict these ephemeral ports.

Note: OTD relies on having sufficient ephemeral ports available so that it can have sufficient pool of connections established with WLS/origin server. Not having enough ephemeral ports will cause delays processing the requests.

7.5 Removing an Origin Server from a Pool

You can remove an origin server from a pool by using either the administration console or the CLI.

Note:

- When dynamic discovery is enabled (see Section 6.5), if you delete an origin server that is an Oracle WebLogic Server instance in a cluster, and then reconfigure the Oracle Traffic Director instance, the instance might not start if no valid origin servers remain in the pool.
- When you remove an origin server from a pool, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Removing an Origin Server from a Pool Using the Administration Console

To remove an origin server from a pool by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to delete origin servers.
4. In the navigation pane, expand **Origin-Server Pools**, expand the pool for which you want to delete origin servers, and select **Origin Servers**.

   A list of the origin servers in the selected pool is displayed.
5. Click the **Delete** icon for the origin server that you want to delete.

   A message is displayed in the Console Message pane confirming that the origin server was deleted.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Removing an Origin Server from a Pool Using the CLI

To remove an origin server from a pool, run the `delete-origin-server` command, as shown in the following example:

```
tadm> delete-origin-server --config=soa --origin-server-pool=osp1
soa-app2.example.com:1900
OTD-70201 Command 'delete-origin-server' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `delete-origin-server`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

8

Managing Virtual Servers

You can use multiple virtual servers within a single Oracle Traffic Director instance to provide several entry points—domain names and IP addresses—for client requests, and to offer differentiated services for caching, quality of service, and so on. You can bind virtual servers to one or more listeners—HTTP or HTTPS—and configure them to forward requests to different origin-server pools.

You can configure caching, compression, routing, quality of service, log-file and web application firewall settings individually for each virtual server.

This chapter describes how to create, view, modify, and delete virtual servers. It contains the following sections:

8.1 Creating a Virtual Server

When you create a configuration, a virtual server is created automatically with the same name as that of the configuration and is associated with the HTTP listener that was specified while creating the configuration. A default routing rule is also created for the virtual server, to distribute all requests received at the associated HTTP listener to the origin servers that were specified while creating the configuration.

You can create additional virtual servers in a configuration by using either the administration console or the CLI.

Note:

- When you create a virtual server, you are, in effect, modifying a configuration. So for the new virtual-server to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Before You Begin

Before you begin creating a virtual server, decide the following:

- A unique name for the virtual server. Choose the name carefully; after creating a virtual server, you cannot change its name.
- One or more unique listen ports. For information about creating listeners, see Chapter 10, "Managing Listeners."
- The names of the hosts, or the host patterns, for which the virtual server will handle requests.

  When a request is received, Oracle Traffic Director determines the virtual server that should process it, by comparing the `Host` header in the request with the host patterns defined for each virtual server in the configuration.

  - The request is routed to the first virtual server that has a host pattern matching the `Host` header in the request.
  - If the `Host` header in the request does not match the host patterns defined for any of the virtual servers, or if the request does not contain the `Host` header, the request is routed to the default virtual server that is associated with the HTTP listener through which the request was received.

  Note: When Strict SNI Host Matching is enabled for an HTTP listener, and if for that listener at least one of the virtual servers has certificates, then Oracle Traffic Director returns a `403-Forbidden` error to the client, if any of the following conditions is true:

  - The client did not send the SNI host extension during the SSL/TLS handshake.
  - The request does not have the `Host:` header.
  - The host name sent by the client in the SNI host extension during the SSL/TLS handshake does not match the `Host:` header in the request.

  For more information, see Section 11.2.6, "About Strict SNI Host Matching."

- The name of the origin-server pool to which the virtual server should forward requests. For information about creating origin-server pools, see Chapter 6, "Managing Origin-Server Pools."

Creating a Virtual Server Using the Administration Console

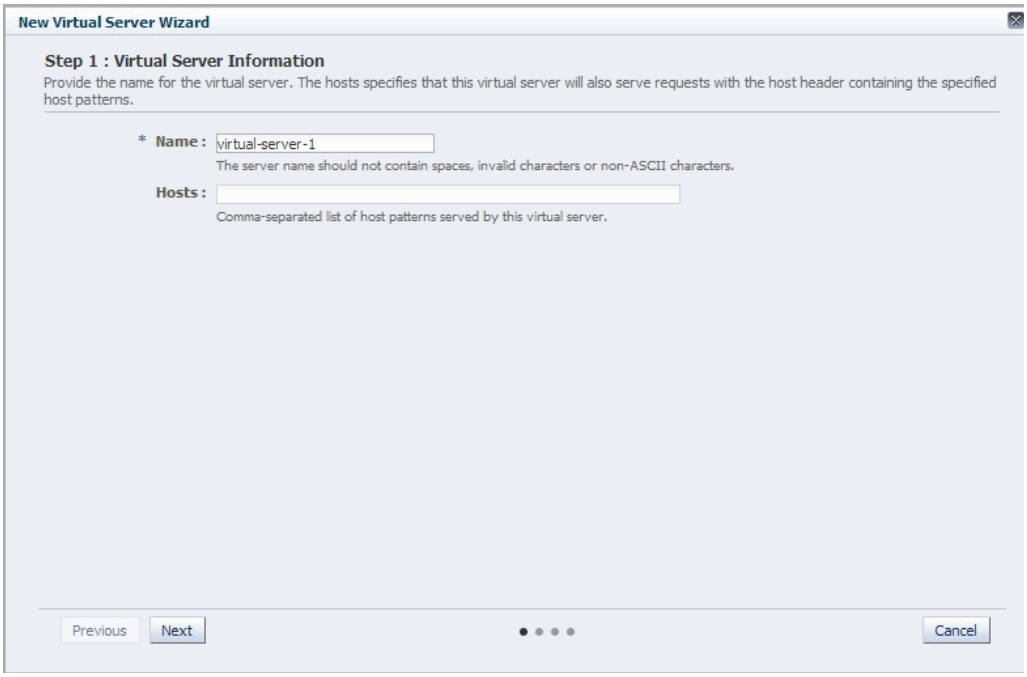
To create a virtual server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create a virtual server.
4. In the Common Tasks pane, click **New Virtual Server**.

   The New Virtual Server wizard starts.

Figure 8–1 New Virtual Server Wizard



New Virtual Server wizard

****

5. Follow the on-screen prompts to complete creation of the virtual server by using the details—listener, origin-server pool, and so on—that you decided earlier.

   After the virtual server is created, the Results screen of the New Virtual Server wizard displays a message confirming successful creation of the virtual server.
6. Click **Close** on the Results screen.
   - The details of the virtual server that you just created are displayed on the Virtual Servers page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Creating a Virtual Server Using the CLI

To create a virtual server, run the `create-virtual-server` command.

For example, the following command creates a virtual server named `vs_soa` associated with the listener `hl1` for the configuration `soa.example.com`, and configures the virtual server to forward client requests to the origin-server pool `soa-pool`.

```
tadm> create-virtual-server --config=soa.example.com --http-listener-name=hl1
--origin-server-pool=soa-pool vs_soa
OTD-70201 Command 'create-virtual-server' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `create-virtual-server`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

8.2 Viewing a List of Virtual Servers

You can view a list of virtual servers by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing List of Virtual Servers Using the Administration Console

To view a list of virtual servers by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view virtual servers.
4. In the navigation pane, select **Virtual Servers**.

   The Virtual Servers page is displayed. It shows a list of the virtual servers defined for the configuration.

You can view the properties of a virtual server by clicking on its name.

Viewing a List of Virtual Servers Using the CLI

To view a list of virtual servers, run the `list-virtual-servers` command, as shown in the following example:

```
tadm> list-virtual-servers --config=soa.example.com
name            http-listener-name
---------------------------------
soa             http-listener-1
adf             adf-listener
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

You can view the properties of a virtual server in detail by running the `get-virtual-server-prop` command.

For more information about the `list-virtual-servers` and `get-virtual-server-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

8.3 Modifying a Virtual Server

You can modify virtual servers by using either the administration console or the CLI.

Note:

- When you modify a virtual server, you are, in effect, modifying a configuration. So for the new virtual-server settings to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Modifying a Virtual Server Using the Administration Console

To modify a virtual server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to modify virtual servers.
4. In the navigation pane, select **Virtual Servers**.

   The Virtual Servers page is displayed. It shows a list of the virtual servers defined for the configuration.
5. Select the virtual server that you want to modify.

   The Virtual Server Settings page is displayed. On this page, you can do the following:

   - Enable and disable the virtual server.
   - Add, remove, and change host patterns served by the virtual server. For more information about how Oracle Traffic Director uses host patterns, see the "Before You Begin" section.
   - Add and remove HTTP listeners. For information about creating HTTP listeners, see Section 10.1, "Creating a Listener."
   - Enable SSL/TLS, by associating an RSA or an ECC certificate (or both) with the virtual server. For more information, see Section 11.2.3, "Associating Certificates with Virtual Servers."
   - Configure the virtual server to serve instance-level statistics in the form of XML and plain-text reports that users can access through a browser. Note that

the statistics displayed in the XML and plain-text reports are for the Oracle Traffic Director instance as a whole and not specific to each virtual server. For more information, see Section 13.3, "Configuring URI Access to Statistics Reports."

- The default language for messages is English. If required, this can be set to other languages that Oracle Traffic Director supports.
- Specify error pages that the virtual server should return to clients for different error codes. This is necessary only if you do not wish to use the default error pages and would like to customize them.

  To specify error codes and error pages of your choice, first create html pages that you would like displayed for specific error codes. Next, on the Virtual Server Settings page, in the Error Pages section, click **New Error Page**.

  In the **New Error Page** dialog box that appears, select an error code and enter the full path to the error page for that particular error code. In addition to the error codes that are provided, you can create your own custom error code by clicking **Custom Error Code** and entering a value for the same. When done, click **Create Error Page**.
- Enable and quality of service limits—the maximum speed at which the virtual server should transfer data to clients and the maximum number of concurrent connections that the virtual server can support.

In the navigation pane, under the **Virtual Servers** node, you can select the following additional categories of settings for the virtual server. The parameters relevant to the selected category are displayed in the main pane.

- **Routes**: Create, change, and delete rules for routing requests to origin servers. For more information, see Section 8.4, "Configuring Routes."
- **Caching**: Create, change, and delete rules for caching responses received from origin servers. For more information, see Section 8.10, "Configuring Caching Parameters."
- **Request Limits**: Create, change, and delete rules for limiting the number and rate of requests received by the virtual server. For more information, see Section 11.9, "Preventing Denial-of-Service Attacks."
- **Compression**: Create, change, and delete rules for compressing responses from origin servers before forwarding them to the clients. For more information, see Section 15.8, "Enabling and Configuring Content Compression."
- **Logging**: Define a server log file and location that is specific to the virtual server. For more information, see Section 12.3, "Configuring Log Preferences."
- **Webapp Firewall Ruleset**: Enable or disable webapp firewall rule set, specify rule set patterns and install rule set files. For more information, see Section 11.7, "Managing Web Application Firewalls."

6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.

- A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
- In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Modifying a Virtual Server Using the CLI

The CLI provides several commands (see Table 8–1) that you can use to change specific parameters of a virtual server.

Table 8–1 CLI Commands for Modifying a Virtual Server

Task/s	CLI Command/s
Enable or disable a virtual server; change the host, the HTTP listener, name and location of the log file; enable SSL/TLS by associating an RSA, or an ECC certificate, or both (see also: Section 11.2.3, "Associating Certificates with Virtual Servers" and Section 12.3, "Configuring Log Preferences"	`set-virtual-server-prop`

Create and manage caching rules (see Section 8.10, "Configuring Caching Parameters"	`create-cache-rule`
	`list-cache-rules`
	`delete-cache-rule`
	`get-cache-rule-prop`
	`set-cache-rule-prop`

Create and manage compression rules (see Section 15.8, "Enabling and Configuring Content Compression")	`create-compression-rule`
	`set-compression-rule-prop`
	`delete-compression-rule`
	`list-compression-rules`
	`get-compression-rule-prop`

Change QoS settings	`set-qos-limits-prop` `get-qos-limits-prop`

Change request limiting settings (see Section 11.9, "Preventing Denial-of-Service Attacks")	`create-request-limit`
	`delete-request-limit`
	`get-request-limit-prop`
	`list-request-limits`
	`set-request-limit-prop`

Create and manage routes (see Section 8.4, "Configuring Routes")	`create-route`
	`list-routes`
	`delete-route`
	`set-route-prop`
	`get-route-prop`

Create and manage error pages	`create-error-page`
	`delete-error-page`
	`list-error-pages`

For example, the following command changes the location of the error log file for the virtual server `soa` to `/home/log/errors.log`.

```
tadm> set-virtual-server-prop --config=soa --vs=soa log-file=/home/log/errors.log
OTD-70201 Command 'set-virtual-server-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

8.4 Configuring Routes

When you create a configuration, a virtual server is automatically created with the listener that you specified while creating the configuration. For the automatically created virtual server, as well as for any virtual server that you add subsequently in the configuration, a default route is created. The default route rule specifies that all requests to the virtual server should be routed to the origin-server pool that you specified while creating the virtual server. The default route of a virtual server cannot be deleted, but you can change its properties.

You can create additional routes for the virtual server, to route requests that satisfy specified conditions to specific origin-server pools. For example, in a banking software solution, if customer transactions for loans and deposits are processed by separate applications, you can host each of those applications in a separate origin-server pool behind an Oracle Traffic Director instance. To route customer requests to the appropriate origin-server pool depending on whether the request pertains to the loans or deposits applications, you can set up two routes as follows:

- Route 1: If the request URI starts with `/loan`, send the request to the origin-server pool that hosts the loans application.
- Route 2: If the request URI starts with `/deposit`, send the request to the origin-server pool that hosts the deposits application.

When a virtual server that is configured with multiple routes receives a request, it checks the request URI against each of the available routes. The routes are checked in the order in which they were created.

- If the request satisfies the condition in a route, Oracle Traffic Director sends the request to the origin-server pool specified for that route.
- If the request does not match the condition in any of the defined routes, Oracle Traffic Director sends the request to the origin-server pool specified in the default route.

WebSocket upgrade is enabled by default. In the Administration Console, use the **WebSocket Upgrade** check box to enable or disable WebSocket protocol for a route. Similarly, WebSocket protocol can also be enabled or disabled using the `websocket-upgrade-enabled` property, which can be set using the set-route-prop CLI command. For more information, see *Oracle Traffic Director Command-Line Reference*.

You can configure routes in a virtual server by using either the administration console or the CLI.

Note:

- When you modify a virtual server, you are, in effect, modifying a configuration. So for the new virtual-server settings to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Configuring Routes Using the Administration Console

To configure routes by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to configure routes.
4. In the navigation pane, expand **Virtual Servers**, expand the name of the virtual server for which you want to configure routes, and select **Routes**.

   The Routes page is displayed. It lists the routes that are currently defined for the virtual server.

   Creating a Route

   a. Click **New Route**.

      The New Route dialog box is displayed.

      In the **Name** field, enter a name for the new route.

      In the **Origin Server Pool** field, select the origin-server pool to which requests that satisfy the specified condition should be routed.

   b. Click **Next**.

      In the Condition Information pane, select a Variable/Function and an Operator from the respective drop-down lists, and provide a value in the **Value** field.

      Select the `and/or` operator from the drop-down list when configuring multiple expressions. Similarly, use the `Not` operator when you want the route to be applied only when the given expression is not true.

      Click **Ok**.

      To enter a condition manually, click **Cancel** and then click **Edit Manually**. In the **Condition** field, specify the condition under which the routing rule should be applied. For information about building condition expressions, click the help button near the Condition field or see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

   c. Click **Next** and then click **Create Route**.

      The route that you just created is displayed on the Routes page.

In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Editing a Route

To change the settings of a route, do the following:

a. Click the **Name** of the route.

The Route Settings page is displayed.

b. Specify the parameters that you want to change.

On-screen help and prompts are provided for all of the parameters.

When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

At any time, you can discard the changes by clicking the **Reset** button.

c. After making the required changes, click **Save**.

A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.

In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Deleting a Route Rule

To delete a route rule, click the **Delete** button. At the confirmation prompt, click **OK**.

Configuring Routes Using the CLI

- To create a route, run the `create-route` command.

  Examples:

  – The following command creates a route named `loan-route` in the virtual server `soa.example.com` of the configuration `soa`, to send requests for which the URI matches the pattern `/loan` to the origin-server pool `loan-app`.

  ```
  tadm> create-route --config=soa --vs=soa.example.com
  --condition="$uri='/loan'" --origin-server-pool=loan-app loan-route
  OTD-70201 Command 'create-route' ran successfully.
  ```

  – The following command creates a route named `images-route` in the virtual server `soa.example.com` of the configuration `soa`, to send requests for which the URI path matches the pattern `/images` to the origin-server pool `images-repo`.

  ```
  tadm> create-route --config=soa --vs=soa.example.com
  --condition="$path='/images/*'" --origin-server-pool=images-repo
  images-route
  OTD-70201 Command 'create-route' ran successfully.
  ```

  – The following command creates a route named `subnet-route` in the virtual server `soa.example.com` of the configuration `soa`, to send requests from any client in the subnet `130.35.46.*` to the origin-server pool `dedicated-osp`.

```
tadm> create-route --config=soa --vs=soa.example.com
--condition="$ip='130.35.45.*'" --origin-server-pool=dedicated-osp
subnet-route
OTD-70201 Command 'create-route' ran successfully.
```

– The following command creates a route named `body-route` in the virtual server `soa.example.com` of the configuration `soa`, to route requests to the origin-server pool `dedicated-osp` if the request body contains the word *alpha*.

```
tadm> create-route --config=soa --vs=soa.example.com --condition="$body
='alpha'" --origin-server-pool=dedicated-osp body-route
OTD-70201 Command 'create-route' ran successfully.
```

Note that the value of the `--condition` option should be a regular expression. For information about building condition expressions, see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

- To view a list of the routes defined for a virtual server, run the `list-routes` command, as shown in the following example:

```
tadm> list-routes --config=soa --vs=soa.example.com
route           condition
------------------------
loan-route       "$uri = '/loan'"
default-route   -
```

- To view the properties of a route, run the `get-route-prop` command, as shown in the following example:

```
tadm> get-route-prop --config=soa --vs=soa.example.com --route=loan-route
keep-alive-timeout=15
sticky-cookie=JSESSIONID
condition="$uri = '/loan'"
validate-server-cert=true
always-use-keep-alive=true
origin-server-pool=origin-server-pool-1
sticky-param=jsessionid
route-header=Proxy-jroute
rewrite-headers=location,content-location
use-keep-alive=true
route=loan-route
log-headers=false
route-cookie=JROUTE
timeout=300
```

- To change the properties of a route, run the `set-route-prop` command.

  Examples:

  – The following command changes the keep-alive timeout setting for the route named `loan-route` in the virtual server `soa.example.com` of the configuration `soa` to 30 seconds.

```
tadm> set-route-prop --config=soa --vs=soa.example.com --route=loan-route
keep-alive-timeout=30
```

  – The following command enables logging of the headers that Oracle Traffic Director sends to, and receives from, the origin servers associated with the route named `default-route` in the virtual server `soa.example.com` of the configuration `soa`.

```
tadm> set-route-prop --config=soa --vs=soa.example.com
--route=default-route log-headers=true
```

- To delete a route, run the `delete-route` command, as shown in the following example:

```
tadm> delete-route --config=soa --vs=soa.example.com loan-route
OTD-70201 Command 'delete-route' ran successfully.
```

- To disable WebSocket support, run the `set-route-prop` command with the `websocket-upgrade-enabled` property, as shown in the following example:

```
tadm> set-route-prop --config=soa --vs=soa.example.com --route=default-route
websocket-upgrade-enabled=false
OTD-70201 Command 'set-route-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

8.5 Copying a Virtual Server

You can copy a virtual server by using either the administration console or the CLI.

Note:

- When you copy a virtual server, you are, in effect, modifying a configuration. So for the new virtual server to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Copying a Virtual Server Using the Administration Console

To copy a virtual server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to copy virtual servers.
4. In the navigation pane, select **Virtual Servers**.

   The Virtual Servers page is displayed. It shows a list of the virtual servers defined for the configuration.
5. Click the **Duplicate** icon for the virtual server that you want to copy.

   The Duplicate Virtual Server dialog box is displayed.

6. Enter a name for the new virtual server, and click **Duplicate**.

   A message is displayed confirming that the new virtual server was created.

7. Click **Close**.

   The virtual server that you just created is displayed on the Virtual Servers page.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Copying a Virtual Server Using the CLI

To copy a virtual server, run the `copy-virtual-server` command.

For example, the following command creates a copy (`vs2`) of the virtual server `vs1`.

```
tadm> copy-virtual-server --config=soa --vs=vs1 vs2
OTD-70201 Command 'copy-virtual-server' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `copy-virtual-server`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

8.6 Deleting a Virtual Server

You can delete virtual servers by using either the administration console or the CLI.

Note:

- When you delete a virtual server, you are, in effect, modifying a configuration. So for the configuration changes to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Deleting a Virtual Server Using the Administration Console

To delete a virtual server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration for which you want to delete virtual servers.
4. In the navigation pane, select **Virtual Servers**.

   The Virtual Servers page is displayed. It shows a list of the virtual servers defined for the configuration.

5. Click the **Delete** icon for the virtual server that you want to delete.

   A prompt to confirm the deletion is displayed.

6. Click **OK**.

   A message is displayed in the Console Message pane confirming that the virtual server was deleted.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Deleting a Virtual Server Using the CLI

To delete a virtual server, run the `delete-virtual-server` command, as shown in the following example:

```
tadm> delete-virtual-server --config=soa vs1
OTD-70201 Command 'delete-virtual-server' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `delete-virtual-server`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

8.7 Caching in Oracle Traffic Director

Caching frequently requested data reduces the time that clients have to wait for responses. In addition, when frequently accessed URLs are stored in memory, the load on the origin servers is significantly reduced.

To enable caching, you must configure caching rules.

- Both static and dynamically generated content from origin servers are cached.
- Only complete responses (response code: `200`) are cached.
- Responses to only HTTP GET and HEAD requests are cached.
- Oracle Traffic Director caches the response body and all of the response headers except `Dest-IP`, `Proxy-Agent`, `Proxy-Connection`, `Server`, `Set-Cookie`, `State-Info`, and `Status`.
- Oracle Traffic Director honors `Cache-Control` directives from origin servers, including directives to revalidate content and to not cache certain headers.
- You can configure one or more caching rules specific to each virtual server, subject to the overall limits—maximum heap space, maximum entries, and maximum object size—specified for the configuration.

  You can configure the caching rules to be applicable either to all requests or to only those requests that match a specified condition.
- Cached data is held in the process memory (heap), separately for each virtual server. When the instance is stopped or restarted, the cache becomes empty.
- WebSocket upgrade requests are not cached.

When a client first requests an object, Oracle Traffic Director sends the request to an origin server. This request is a *cache miss*. If the requested object matches a caching rule, Oracle Traffic Director caches the object. For subsequent requests for the same

object, Oracle Traffic Director serves the object from its cache to the client. Such requests are *cache hits*.

The caching behavior in Oracle Traffic Director is consistent with the specification in section 13 of RFC 2616. For more information, see http://www.w3.org/Protocols/rfc2616/rfc2616-sec13.html.

8.8 Reviewing Caching Settings and Metrics for an Instance

Viewing Caching Settings

- To view the current caching settings for a configuration, run the `get-cache-prop` command, as shown in the following example:

```
tadm> get-cache-prop --config=soa
enabled=true
replacement=lru
max-heap-space=10485760
max-entries=1024
max-heap-object-size=524288
```

- To view a list of the caching rules defined for a virtual server, run the `list-cache-rules` command, as shown in the following example:

```
tadm> list-cache-rules --config=soa --vs=soa --verbose --all
rule            condition
------------------------
cache-rule-2    "$uri = '^/images'
cache-rule-1    -
```

- To view the current settings of a virtual server-specific caching rule, run the `get-cache-rule-prop` command, as shown in the following example:

```
tadm> get-cache-rule-prop --config=soa --vs=soa --rule=cache-rule-2
enabled=true
last-modified-factor=0
min-object-size=1
cache-https-response=true
condition="$uri = '^/images"
min-reload-interval=0
rule=cache-rule-2
query-maxlen=0
compression=true
max-reload-interval=3600
```

Viewing Caching Metrics

You can view the current cache-hit rate, the cache heap usage, and the rate of successful revalidation of cache entries in the plain-text `perfdump` report, as shown in the following example:

```
Proxy Cache:
--------------------------
Proxy Cache Enabled             yes
Object Cache Entries            42
Cache lookup (hits/misses)      183/79
Requests served from Cache      22
Revalidation (successful/total) 30/38 (  78.95%)
Heap space used                 16495
```

- `Proxy Cache Enabled` indicates whether caching is enabled for the instance.

- `Object Cache Entries` is the number of entries (URIs) currently in the cache.
- `Cache lookup (hits/misses)`
  - The first number is the number of times an entry was found in the cache for the requested URI.
  - The second number is the number of times the requested URI was not found in the cache.
- `Requests served from Cache` is the number of requests that Oracle Traffic Director served from the cache.
- `Revalidation (successful/total)`
  - The first number is the number of times revalidation of cached content was successful.
  - The second number is the total number of times Oracle Traffic Director attempted to revalidate cached content.
  - The percentage value is the ratio of successful revalidations to the total number of revalidation attempts.
- `Heap space used` is the amount of cache heap space that is currently used.

8.9 Tunable Caching Parameters

Caching can be considered effective in reducing the response time for clients when the cache-hit rate is high; that is, a relatively large number of requests are served from the cache instead of being sent to origin servers. For a high cache-hit rate, there should be sufficient memory to store cacheable responses from origin servers and the entries in the cache should be validated regularly.

> **Note:** Dynamic content is generally not cacheable. So if the application or content being served by the origin servers consists mostly of dynamic content, the cache-hit rate is bound to be low. In such cases, enabling and tuning caching might not yield a significant performance improvement.

To improve the cache-hit rate, you can tune the following caching parameters:

- **Cache-entry replacement method**

  When the cache becomes full—that is, the number of entries reaches the maximum entries limit, or the cache heap size reaches the maximum cache heap space—further entries in the cache can be accommodated only if existing entries are removed. The cache-entry replacement method specifies how Oracle Traffic Director determines the entries that can be removed from the cache.

  - The default replacement method is Least Recently Used (`lru`). When the cache is full, Oracle Traffic Director discards the least recently used entries first.
  - The other available method is Least Frequently Used (`lfu`). When the cache is full, Oracle Traffic Director discards the least frequently used entry first.

  In either method, every time Oracle Traffic Director serves content from the cache, it needs to track usage information—the time the content was served in the case of the `lru` replacement method, and the number of times the content was served in the case of `lfu`. So the time saved by serving content directly from the cache instead of sending the request to the origin server, is offset to a certain extent by

the latency caused by the need to track usage information. Between the two methods, `lru` requires marginally lower computing resources.

You can disable cache-entry replacement by specifying `false` as the replacement method.

- **Maximum cache heap space**

  If only a small portion of the available heap space is used, it is possible that responses are not being cached because the virtual server-specific caching rules are defined too narrowly.

  The optimal cache heap size depends upon how much system memory is free. With a large cache heap, Oracle Traffic Director can cache more content and therefore obtain a better hit ratio. However, the heap size should not be so large that the operating system starts paging cached content.

- **Maximum number of entries in the cache**

  If the number of entries in the cache, as shown in the `perfdump` report, is consistently near, or at, the maximum number of entries, it is an indication that the cache might not be large enough. Consider increasing the maximum number of entries.

  If the number of entries in the cache is very low when compared with the maximum allowed entries, it is possible that responses are not being cached because the virtual server-specific caching rules are defined too narrowly.

- **Maximum size of cacheable object**

  To conserve system resources, you can limit the size of objects that are cached, even if the objects fulfill other caching rules.

  If you observe that objects that are larger than the maximum cached object size are requested frequently, consider increasing the limit.

In a caching rule for a specific virtual server, you can specify the following parameters:

- Minimum and maximum size of objects that can be cached
- Minimum and maximum interval between cache-validation checks
- Maximum number of characters in a query string that can be cached
- Whether to compress content before caching
- Whether to cache HTTPS responses

8.10 Configuring Caching Parameters

You can configure caching settings by using either the administration console or the CLI.

Configuring Caching Settings Using the Administration Console

To configure caching settings by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration that you want to modify.

4. In the navigation pane, select **Advanced Settings**.

   The Advanced Settings page is displayed.

5. Go to the **Cache** section on the page.

6. Specify the caching parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.

   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Virtual Server-Specific Caching Rules Using the Administration Console

To create virtual server-specific caching rules by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."

2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration for which you want to create virtual server-specific caching rules.

4. In the navigation pane, expand **Virtual Servers**, expand the name of the virtual server for which you want to create caching rules, and select **Caching**.

   The Caching page is displayed. It lists the caching rules that are currently defined for the virtual server, and indicates whether the rules are enabled.

   Creating a Caching Rule

   a. Click **New Caching Rule**.

      The New Cache Rule dialog box is displayed.

      In the **Name** field, enter a name for the new caching rule.

   b. Click **Next**.

      If this is the first caching rule for the virtual server, the New Caching Rule dialog box gives you the option to choose whether the rule should be applied to all requests. Select **All Requests**.

      If you wish to apply the rule to only those requests that satisfy a condition, create a new condition by selecting **Create a new condition**. In the condition builder, select a Variable/Function and an Operator from the respective drop-down lists and provide a value in the **Value** field.

Select the `and`/or `operator` from the drop-down list when configuring multiple expressions. Similarly, use the `Not` operator when you want the route to be applied only when the given expression is not true.

To enter a condition manually, click **Cancel** and then click **Edit Manually**. In the **Condition** field, specify the condition under which the caching rule should be applied. For information about building condition expressions, click the help button near the Condition field or see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

c. Click **Next** and then click **Create Caching Rule**.

The caching rule that you just created is displayed on the Caching page.

In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Editing a Caching Rule

To enable or disable a caching rule, or to change the settings of a rule, do the following:

1. Click the **Name** of the caching rule that you want to edit.

   The Edit Cache Rule dialog box is displayed.

   Note: To access the condition builder to edit conditions, select **Requests satisfying the condition** and click **Edit**. The condition builder enables you to delete old expressions and add new ones.

2. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   For information about building condition expressions, click the help button near the Condition field or see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

3. After making the required changes, click **Save**.

   A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Deleting a Caching Rule

To delete a caching rule, click the **Delete** button. At the confirmation prompt, click **OK**.

Configuring Caching Settings Using the CLI

- To change the caching properties for a configuration, run the `set-cache-prop` command.

  For example, the following command changes the maximum cache heap space to 20 MB.

  ```
  tadm> set-cache-prop --config=soa max-heap-space=20971520
  OTD-70201 Command 'set-cache-prop' ran successfully.
  ```

- To create a caching rule for a virtual server, run the `create-cache-rule` command.

  For example, the following command creates a rule named `cache-rule-images` for the virtual server `soa.example.com` in the configuration `soa`, to cache the requests for which the expression `$uri='^/images'` evaluates to true.

  ```
  tadm> create-cache-rule --condition="\$uri='^/images'" --config=soa
  --vs=soa.example.com cache-rule-images
  OTD-70201 Command 'create-cache-rule' ran successfully.
  ```

  Note that the value of the `--condition` option should be a regular expression. For information about building condition expressions, see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

- To change a caching rule, run the `set-cache-rule-prop` command.

  For example, the following command disables compression of content for the caching rule `cache-rule-images`.

  ```
  tadm> set-cache-rule-prop --config=soa --vs=soa.example.com
  --rule=cache-rule-images compression=false
  OTD-70201 Command 'set-cache-rule-prop' ran successfully.
  ```

- To delete a caching rule, run the `delete-cache-rule` command, as shown in the following example.

  ```
  tadm> delete-cache-rule --config=soa --vs=soa.example.com cache-rule-images
  OTD-70201 Command 'delete-cache-rule' ran successfully.
  ```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

9 Managing TCP Proxies

A TCP Proxy handles TCP requests through TCP listeners for traffic tunnelling. While a TCP Proxy can have several TCP listeners associated with it, a TCP listener can be associated with only one TCP Proxy.

This chapter describes how to create, view, modify, and delete TCP proxies. It contains the following topics:

9.1 Creating a TCP Proxy

You can create TCP proxies by using either the administration console or the CLI.

Note:

- When you create a TCP Proxy, you are, in effect, modifying a configuration. So for the new TCP Proxy settings to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Before You Begin

Before you begin creating a TCP Proxy, decide the following:

- A unique name for the proxy. Choose the name carefully; after creating a proxy, you cannot change its name.
- A unique IP address (or host name) and port number combinations for the listener.

  You can define multiple TCP listeners with the same IP address combined with different port numbers, or with a single port number combined with different IP addresses. So each of the following IP address and port number combinations would be considered a unique listener:

```
10.10.10.1:80
10.10.10.1:81
```

```
10.10.10.2:80
10.10.10.2:81
```

- The name of the origin-server pool to which the TCP Proxy should forward requests. For information about creating origin-server pools, see Chapter 6, "Managing Origin-Server Pools."

Creating a TCP Proxy Using the Administration Console

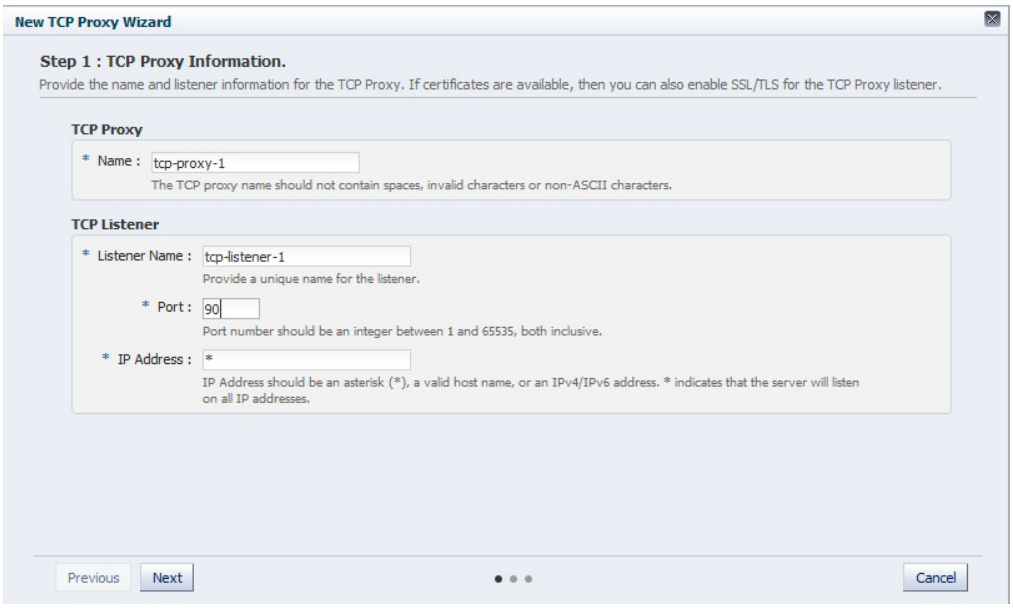
To create a TCP Proxy by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create a TCP Proxy.
4. In the Common Tasks pane, click **New TCP Proxy**.

   The New TCP Proxy wizard starts.

Figure 9–1 New TCP Proxy Wizard



New TCP Proxy wizard

****

5. Follow the on-screen prompts to complete creation of the TCP Proxy by using the details—proxy name, listener name, IP address, port, and so on—that you decided earlier.

> **Note:** If the TCP traffic on the port is over SSL, for example T3S, then select the **SSL/TLS** check box on the first screen of the New TCP Proxy wizard and select the certificate to be used. For more information, see Section 11.2.2, "Configuring SSL/TLS for a Listener."

After the proxy is created, the Results screen of the New TCP Proxy wizard displays a message confirming successful creation of the proxy.

6. Click **Close** on the Results screen.
   - The details of the TCP Proxies that you just created are displayed on the TCP proxies page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Creating a TCP Proxy Using the CLI

To create a TCP Proxy, run the `create-tcp-proxy` command.

For example, the following command creates a TCP Proxy named `tcp_proxy1` for the configuration `soa.example.com` with the port as `1910` and the origin-server-pool as `soa-pool`.

```
tadm> create-tcp-proxy --config=soa.example.com --origin-server-pool=soa-pool
--port=1910 tcp_proxy1
OTD-70201 Command 'create-tcp-proxy' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `create-tcp-proxy`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

9.2 Viewing a List of TCP Proxies

You can view a list of TCP proxies by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing a List of TCP Proxies Using the Administration Console

To view a list of TCP proxies by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view TCP proxies.
4. In the navigation pane, select **TCP Proxies**.

   The TCP Proxies page is displayed. It shows a list of the TCP proxies defined for the configuration.

You can view the properties of a proxy in detail by clicking on its name.

Viewing a List of TCP Proxies Using the CLI

To view a list of TCP proxies, run the `list-tcp-proxies` command, as shown in the following example:

```
tadm> list-tcp-proxies --config=soa --verbose --all
name            session-idle-timeout        origin-server-pool-name
-------------------------------------------------------------------
tcp_proxy1      300                         soa-pool1
tcp_proxy2      400                         soa-pool2
```

You can view the properties of a TCP Proxy in detail by running the `get-tcp-proxy-prop` command.

For more information about the `list-tcp-proxies` and `get-tcp-proxy-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

9.3 Modifying a TCP Proxy

You can modify TCP proxies by using either the administration console or the CLI.

Note:

- When you modify a TCP Proxy, you are, in effect, modifying a configuration. So for the new settings of the TCP Proxy to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Modifying a TCP Proxy Using the Administration Console

To modify a TCP Proxy by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to modify TCP proxies.
4. In the navigation pane, select **TCP Proxies**.

   The TCP Proxies page is displayed. It shows a list of the TCP proxies defined for the configuration.
5. Click the name of the TCP Proxy that you want to modify.

   The TCP Proxy Settings page is displayed. On this page, you can do the following:

   - Enable and disable the TCP Proxy.
   - Change the origin server pool and idle timeout.
   - Add and remove TCP listeners. For information about creating TCP listeners, see Section 10.1, "Creating a Listener."

6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.

   - A message, confirming that the updated proxy was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Modifying a TCP Proxy Using the CLI

- To change the properties of a TCP Proxy, run the `set-tcp-proxy-prop` command. For example, the following command changes the session idle timeout of the proxy `tcp_proxy1` in the configuration `soa` to `500`.

```
tadm> set-tcp-proxy-prop --config=soa --tcp-proxy=tcp_proxy1
session-idle-timeout=500
OTD-70201 Command 'set-tcp-proxy-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For a list of the properties that you can set or change by using the `set-tcp-proxy-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

9.4 Deleting a TCP Proxy

You can delete TCP proxies by using either the administration console or the CLI.

Note:

- When you delete a TCP Proxy, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Deleting a TCP Proxy Using the Administration Console

To delete a TCP Proxy by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

A list of the available configurations is displayed.

3. Select the configuration for which you want to delete TCP proxies.
4. In the navigation pane, select **TCP Proxies**.

   The TCP Proxies page is displayed. It shows a list of the TCP proxies defined for the configuration.
5. Click the **Delete** icon for the TCP Proxy that you want to delete.

   A prompt to confirm deletion of the proxy is displayed. If the proxy is associated with any listeners, the prompt shows the names of those listeners.
6. To proceed with the deletion, click **OK**.

   A message is displayed in the Console Message pane confirming that the TCP Proxy was deleted.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Deleting a TCP Proxy Using the CLI

To delete a TCP Proxy, run the `delete-tcp-proxy` command, as shown in the following example:

```
tadm> delete-tcp-proxy --config=soa tcp_proxy1
OTD-70201 Command 'delete-tcp-proxy' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `delete-tcp-proxy`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

10 Managing Listeners

Connections between the clients and Oracle Traffic Director instances are created through HTTP and TCP listeners. Each listener is a unique combination of an IP address (or host name) and a port number.

This chapter describes how to create, view, modify, and delete listeners. It contains the following topics:

10.1 Creating a Listener

You can create listeners by using either the administration console or the CLI.

Note:

- When you create a listener, you are, in effect, modifying a configuration. So for the new listener settings to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Before You Begin

Before you begin creating an listener, decide the following:

- A unique name for the listener. Choose the name carefully; after creating a listener, you cannot change its name.
- A unique IP address (or host name) and port number combinations for the listener.

  You can define multiple listeners with the same IP address combined with different port numbers, or with a single port number combined with different IP addresses. So each of the following IP address and port number combinations would be considered a unique listener:

```
10.10.10.1:80
10.10.10.1:81
```

```
10.10.10.2:80
10.10.10.2:81
```

- For HTTP listeners: The default virtual server for the listener.

  Oracle Traffic Director routes requests to the default virtual server if it cannot match the `Host` value in the request header with the host patterns specified for any of the virtual servers bound to the listener.

  For information about specifying the host patterns for virtual servers, see Section 8.1, "Creating a Virtual Server."

- For HTTP listeners: The server name to be included in any URLs that are generated automatically by the server and sent to the client. This server name should be the virtual host name, or the alias name if your server uses an alias. If a colon and port number are appended to the server name then that port number is used in the autogenerated URLs.

- For TCP listeners: TCP proxy for the listener.

  A TCP proxy handles TCP requests through TCP listeners for traffic tunnelling. A TCP proxy can have several TCP listeners associated with it. You can associate TCP listeners and configure TCP proxy settings from this page.

  For more information about creating TCP proxies, see Section 9.1, "Creating a TCP Proxy."

Creating an HTTP Listener Using the Administration Console

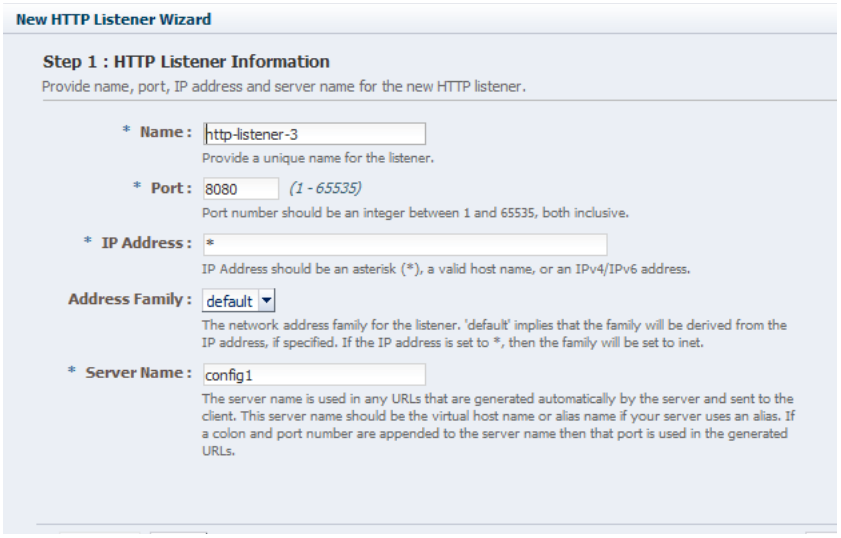
To create an HTTP listener by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create an HTTP listener.
4. In the Common Tasks pane, click **New HTTP Listener**.

   The New HTTP Listener wizard starts.

Figure 10–1 New HTTP Listener Wizard



New HTTP Listener wizard

****

5. Follow the on-screen prompts to complete creation of the HTTP listener by using the details—listener name, IP address, port, and so on—that you decided earlier.

   Note: If certificates are available in the configuration, in the second screen of the wizard, an **SSL/TLS** check box will be available. If you want the new listener to receive HTTPS requests, click the check box to enable **SSL/TLS** and then select the appropriate certificate from the drop-down list.

   After the HTTP listener is created, the Results screen of the New HTTP Listener wizard displays a message confirming successful creation of the listener.

6. Click **Close** on the Results screen.

   - The details of the listener that you just created are displayed on the Listeners page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Creating a TCP Listener Using the Administration Console

To create a TCP listener by using the administration console, do the following:

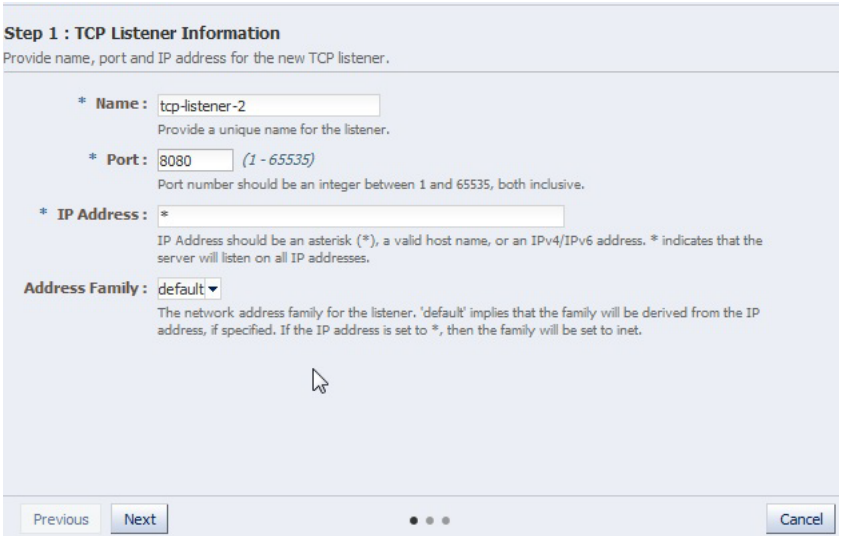
1. Perform steps 1, 2, and 3 of Section , "Creating an HTTP Listener Using the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create a TCP listener.

4. In the Common Tasks pane, click **New TCP Listener**.

   The New TCP Listener wizard starts.

Figure 10–2 New TCP Listener Wizard



New TCP Listener wizard

****

5. Follow the on-screen prompts to complete creation of the TCP listener by using the details—listener name, IP address, port, and so on—that you decided earlier.

   > **Note:** If certificates are available in the configuration, in the second screen of the wizard, an **SSL/TLS** check box will be available. If you want the new listener to receive T3S requests, click the check box to enable **SSL/TLS** and then select the appropriate certificate from the drop-down list.

   After the TCP listener is created, the Results screen of the New TCP Listener wizard displays a message confirming successful creation of the listener.

6. Click **Close** on the Results screen.

   - The details of the listener that you just created are displayed on the Listeners page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Creating a Listener Using the CLI

- To create an HTTP listener, run the `create-http-listener` command.

  For example, the following command creates an HTTP listener named `listener_soa` for the configuration `soa.example.com` with the port as `1910` and the default virtual server as `soa`.

```
tadm> create-http-listener --config=soa.example.com --listener-port=1910
```

```
--server-name=soa.example.com --default-virtual-server-name=soa listener_soa
OTD-70201 Command 'create-http-listener' ran successfully.
```

- To create a TCP listener, run the `create-tcp-listener` command.

  For example, the following command creates a TCP listener named `tcp_listener_soa` for the configuration `soa.example.com` with the port as `1920` and the TCP Proxy as `tcp_proxy1`.

```
tadm> create-tcp-listener --config=soa.example.com --listener-port=1920
--server-name=soa.example.com --tcp-proxy=tcp_proxy1 listener_soa
OTD-70201 Command 'create-tcp-listener' ran successfully.
```

For more information about `create-http-listener` and `create-tcp-listener`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

10.2 Viewing a List of Listeners

You can view a list of HTTP or TCP listeners by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing a List of Listeners Using the Administration Console

To view a list of HTTP or TCP listeners by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view HTTP or TCP listeners.
4. In the navigation pane, expand **Listeners**, and select a listener.

   The Listeners page is displayed. It shows a list of the listeners defined for the configuration.

> **Note:** HTTP and TCP listeners can also be identified by their icons.

You can view the properties of a listener in detail by clicking on its name.

Viewing a List of Listeners Using the CLI

- To view a list of HTTP listeners, run the `list-http-listeners` command, as shown in the following example:

```
tadm> list-http-listeners --config=soa --verbose --all
name          ip       port    ssl-enabled     default-virtual-server
-------------------------------------------------------------------------
```

```
listener-1    *        1904       false          vs1
listener-2    *        80         false          vs1
```

You can view the properties of an HTTP listener in detail by running the `get-http-listener-prop` command.

For more information about the `list-http-listeners` and `get-http-listener-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

- To view a list of TCP listeners, run the `list-tcp-listeners` command, as shown in the following example:

```
tadm> list-tcp-listeners --config=soa --verbose --all
name          ip         port         ssl-enabled         tcp-proxy-name
-------------------------------------------------------------------------
listener-1    *          9090         false               tcp_proxy1
listener-2    *          9092         false               tcp_proxy1
```

You can view the properties of an TCP listener in detail by running the `get-tcp-listener-prop` command.

For more information about the `list-tcp-listeners` and `get-tcp-listener-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

10.3 Modifying a Listener

You can modify listeners by using either the administration console or the CLI.

Note:

- When you modify a listener, you are, in effect, modifying a configuration. So for the new settings of a listener to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Modifying a Listener Using the Administration Console

To modify an HTTP or TCP listener by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to modify listeners.
4. In the navigation pane, click **Listeners**.

   The Listeners page is displayed. It shows a list of the HTTP/TCP listeners defined for the configuration.

5. Click the name of the listener that you want to modify.

   The Listener Settings page is displayed. On this page, you can do the following:

   - Enable and disable the listener.
   - Change the listener port number and IP address.
   - For HTTP listeners: Change the server name and the default virtual server.
   - For TCP listeners: Change the TCP proxy.
   - If server certificates have been created for the configuration, you can enable SSL/TLS and configure SSL/TLS settings for the listener. For more information, see Section 11.2.2, "Configuring SSL/TLS for a Listener."
   - Change the protocol family—IPv4, IPv6, or SDP—for which the listener should accept requests.
   - For HTTP listeners: Configure parameters to tune the performance of the virtual server—the number of acceptor threads, the listen queue size, receive buffer size, and so on. For more information, see Section 15.2.2, "Tuning HTTP Listener Settings."

6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.

   - A message, confirming that the updated listener was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Modifying a Listener Using the CLI

- To change the properties of an HTTP listener, run the `set-http-listener-prop` command. For example, the following command changes the port number of the listener `ls1` in the configuration `soa` to `1911`.

```
tadm> set-http-listener-prop --config=soa --http-listener=ls1 port=1911
OTD-70201 Command 'set-http-listener-prop' ran successfully.
```

  To change the SSL/TLS settings of an HTTP listener, run the `set-ssl-prop` command. For example, the following command enables SSL 3.0 support for the listener `ls1` in the configuration `soa`.

```
tadm> set-ssl-prop --config=soa --http-listener=ls1 ssl3=true
OTD-70201 Command 'set-ssl-prop' ran successfully.
```

  To change the properties of a TCP listener, run the `set-tcp-listener-prop` command. For example, the following command changes the port number of the listener `tcp_ls1` in the configuration `soa` to `1911`.

```
tadm> set-tcp-listener-prop --config=soa --tcp-listener=tcp_ls1
listen-queue-size=238
OTD-70201 Command 'set-tcp-listener-prop' ran successfully.
```

To change the SSL/TLS settings of an TCP listener, run the `set-ssl-prop` command. For example, the following command enables SSL 3.0 support for the listener `tcp_ls1` in the configuration `soa`.

```
tadm> set-ssl-prop --config=soa --tcp-listener=ls1 ssl3=true
OTD-70201 Command 'set-ssl-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For a list of the properties that you can set or change by using the `set-tcp-listener-prop` and `set-ssl-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

10.4 Deleting a Listener

You can delete HTTP or TCP listeners by using either the administration console or the CLI.

Note:

- When you delete a listener, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Deleting a Listener Using the Administration Console

To delete an HTTP or TCP listener by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to delete listeners.
4. In the navigation pane, select **Listeners**.

   The Listeners page is displayed. It shows a list of the listeners defined for the configuration.
5. Click the **Delete** icon for the listener that you want to delete.

   A prompt to confirm deletion of the listener is displayed.

   Note: For HTTP listeners: If the HTTP listener is associated with any virtual servers, the prompt shows the names of those virtual servers.
6. To proceed with the deletion, click **OK**.

A message is displayed in the Console Message pane confirming that the HTTP/TCP listener was deleted.

In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Deleting a Listener Using the CLI

- To delete an HTTP listener, run the `delete-http-listener` command, as shown in the following example:

```
tadm> delete-http-listener --config=soa http-listener-1
OTD-70201 Command 'delete-http-listener' ran successfully.
```

To delete an TCP listener, run the `delete-tcp-listener` command, as shown in the following example:

```
tadm> delete-tcp-listener --config=soa tcp-listener-1
OTD-70201 Command 'delete-tcp-listener' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `delete-http-listener` and `delete-tcp-listener`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Part III

Advanced Administration

Part III contains the following chapters:

- Chapter 11, "Managing Security" describes how to secure access to the administration server; how to enable SSL/TLS for Oracle Traffic Director virtual servers, manage certificates; and how to manage certificates, PKCS#11 tokens, and certificate revocation lists.
- Chapter 12, "Managing Logs" provides an overview of the access and server logs; and describes how you can view logs, configure log preferences, and rotate logs.
- Chapter 13, "Monitoring Oracle Traffic Director Instances" describes the methods you can use to monitor Oracle Traffic Director instances.
- Chapter 14, "Configuring Oracle Traffic Director for High Availability" describes the high-availability features of Oracle Traffic Director. It describes how to configure Oracle Traffic Director instances in a failover group and set up Oracle Traffic Director to monitor the health of the origin servers in the back end.
- Chapter 15, "Tuning Oracle Traffic Director for Performance" describes the various parameters that you can tune to improve the performance of Oracle Traffic Director instances.
- Chapter 16, "Diagnosing and Troubleshooting Problems" provides information to help you understand and solve problems that you might encounter while using Oracle Traffic Director.
- Appendix A, "Metrics Tracked by Oracle Traffic Director" lists the names of the various metrics that Oracle Traffic Director tracks.
- Appendix B, "Web Application Firewall Examples and Use Cases" provides some basic information about how the web application firewall works.
- Appendix C, "Securing Oracle Traffic Director Deployment" provides information about the steps that you can take to secure your Oracle Traffic Director deployment.

11 Managing Security

This chapter describes how you can secure access to the Oracle Traffic Director administration server and enable SSL/TLS for Oracle Traffic Director virtual servers. It also describes how to configure client authentication and how you can use Oracle Traffic Director to secure access to origin servers.

This chapter contains the following sections:

- Securing Access to the Administration Server
- Configuring SSL/TLS Between Oracle Traffic Director and Clients
- Configuring SSL/TLS Between Oracle Traffic Director and Origin Servers
- Managing Certificates
- Managing PKCS#11 Tokens
- Managing Certificate Revocation Lists
- Managing Web Application Firewalls
- Configuring Client Authentication
- Preventing Denial-of-Service Attacks

> **Note:** For information about some steps that you can take to secure Oracle Traffic Director in your environment, see Appendix C, "Securing Oracle Traffic Director Deployment."

11.1 Securing Access to the Administration Server

The administration server instance of Oracle Traffic Director hosts the administration console and command-line interface. So it is important to secure access to the administration server.

User access to the administration server interfaces is controlled through password-based authentication.

- By default, the administration server enables only one administrator user account, which you specify while creating the administration server. For information about changing the administrator user name and password, see Section 11.1.1, "Changing the Administrator User Name and Password."
- To allow multiple users to log in to the administration server, you can enable LDAP authentication. For more information, see Section 11.1.2, "Configuring LDAP Authentication for the Administration Server."

SSL authentication of the Oracle Traffic Director administration server with clients as well as with administration nodes is enabled, by default, through the use of two self-signed certificates—`Admin-Client-Cert` and `Admin-Server-Cert`.

- The self-signed administration-server certificates are created automatically when you create the administration server and are valid for 12 months. For information about renewing the administration-server certificates, see Section 11.1.4, "Renewing Administration Server Certificates."
- You can secure access to the software database containing the self-signed administration-server certificates by defining a password for the token named `internal`, which provides the interface to the certificates database. For more information, see Section 11.1.3, "Enabling the Pin for the Administration Server's PKCS#11 Token."

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

11.1.1 Changing the Administrator User Name and Password

You can change the administrator user name and password by using either the administration console or the CLI.

Changing the Administrator User Name and Password Using the Administration Console

To change the administrator user name and password by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button that is situated near the upper left corner of the page.

   A list of available nodes is displayed.
3. From the list of nodes, select **Administration Server**.
4. In the navigation pane, select **Authentication**.

   The Authentication page is displayed.
5. Specify the user name and password, and then click **Save**.

   > **Note:** The user name can contain a maximum of 100 characters and must not contain spaces.

   A message is displayed in the Console Messages pane indicating that the updated settings are saved.
6. Restart the administration server by clicking **Restart** in the Common Tasks pane.

Changing the Administrator User Name and Password Using the CLI

- To change the administrator user name, run the `set-admin-prop` command.

```
tadm> set-admin-prop admin-user=user_name
OTD-70213 The administration server must be restarted for the changes to take
effect.
```

The user name can contain a maximum of 100 characters and must not contain spaces.

- To change the password, do the following:

  1. Run one of the following commands:

     ```
     tadm> set-admin-prop --set-password
     ```

     or

     ```
     tadm> reset-admin-password
     ```

     The following prompt is displayed:

     ```
     Enter admin-password>
     ```

  2. Enter the new password.

     A prompt to re-enter the new password is displayed:

     ```
     Enter admin-password again>
     ```

  3. Re-enter the new password.

     The following message is displayed.

     ```
     OTD-70201 Command 'reset-admin-password' ran successfully.
     ```

For the new user name and password to take effect, you should restart the administration server as described in Section 2.4, "Stopping and Restarting the Administration Server."

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.1.2 Configuring LDAP Authentication for the Administration Server

If you need more than one user to be able to log in to the administration server, you can store the user IDs and passwords in a directory server, and you can configure Oracle Traffic Director to access the directory server by using the Lightweight Directory Access Protocol (LDAP).

You can enable and configure LDAP authentication for the administration server by using either the administration console or the CLI.

Before You Begin

Before you start configuring Oracle Traffic Director to use LDAP authentication, keep the following information ready. This information is required for constructing the `ldap(s)://`*`host:port/baseDN`* URL that Oracle Traffic Director should use to access the LDAP directory server and for the directory server to search for the required user record.

- The name of the host on which the directory server runs.
- The port number at which the directory server listens for requests from LDAP clients.
- The base Distinguished Name (DN), which is the location within the directory information tree at which the directory server should start searching for the required user record.

- The bind DN, which is the user ID and password that Oracle Traffic Director provides to authenticate itself to the LDAP directory server.

  Note: If your directory server allows searches by anonymous clients, you need not specify the bind DN.

- The user groups whose members can access the administration server.

  By default, the administration server allows only users belonging to the group `wsadmin` to log in. While enabling LDAP authentication, you can specify a list of groups other than `wsadmin` whose members can log in.

Configuring LDAP Authentication for the Administration Server Using the Administration Console

To configure LDAP authentication for the administration server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button that is situated near the upper left corner of the page.

   A list of available nodes is displayed.
3. From the list of nodes, select **Administration Server**.
4. In the navigation pane, select **Authentication**.

   The Authentication page is displayed.
5. Select **LDAP Authentication**.
6. Specify the mandatory parameters—host name, port, base DN, and allowed groups—and the optional parameters, as required.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.

   A message is displayed in the Console Messages pane indicating that the updated settings are saved.
8. Restart the administration server by clicking **Restart** in the Common Tasks pane.

Configuring LDAP Authentication for the Administration Server Using the CLI

- To enable LDAP authentication, run `enable-admin-ldap-auth`, as shown in the following example:

```
> tadm enable-admin-ldap-auth
 --ldap-url=ldap://ldap.example.com:3950/dc=example,dc=com
 --bind-dn=cn="Directory Manager" --allow-groups=sys,adm,mgr
OTD-70213 The administration server must be restarted for the changes to take
effect.
```

  This command configures Oracle Traffic Director as an LDAP client for the directory server `ldap.example.com:3950`. Oracle Traffic Director authenticates itself to the directory server by using the bind DN `cn="Directory Manager"`, and

the directory server starts the search for the required user record at the base DN `dc=example,dc=com`.

- To disable LDAP authentication, run `disable-admin-ldap-auth`, as shown in the following example:

```
> tadm disable-admin-ldap-auth
OTD-70213 The administration server must be restarted for the changes to take
effect.
```

- To view the currently configured LDAP authentication properties, run `get-admin-ldap-auth-prop`, as shown in the following example:

```
> tadm get-admin-ldap-auth-prop
enabled=true
ldap-url="ldap://ldap.example.com:3950/dc=example,dc=com"
search-filter=uid
group-search-filter=uniquemember
group-search-attr=CN
timeout=10
allow-group=sys,adm,mgr
```

For more information about the `enable-admin-ldap-auth`, `disable-admin-ldap-auth`, and `get-admin-ldap-auth-prop` CLI commands, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.1.3 Enabling the Pin for the Administration Server's PKCS#11 Token

The administration server's self-signed certificates are stored in a Public-Key Cryptography Standards (PKCS) 11-compliant security database. Access to the certificates database is provided through a token named `internal`. To secure access to the administration server's certificates database, you can enable the pin for the `internal` token.

If you enable the pin for the `internal` PKCS#11 token in the administration server configuration, a prompt to enter the token pin is displayed when you perform the following tasks:

- Start the administration server.
- Renew the administration server certificates.

You can set, change, or disable the pin for the `internal` token by using either the administration console or the CLI.

Setting the PKCS#11 Token Pin for the Administration Server Using the Administration Console

To set the PKCS#11 token pin for the administration server by using the administration console, do the following

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button that is situated near the upper left corner of the page.

   A list of the available nodes is displayed.
3. Select the **Administration Server** node.

   The General Settings page is displayed.
4. In the **Token Pin Management** section, select the **Edit Token Pin** check box.

- To set the pin, enter the pin and confirm it in the **New Pin** and **New Pin Again** fields respectively.
- To change the pin, enter the current pin in the **Current Pin** field. Then, enter the new pin and confirm it in the **New Pin** and **New Pin Again** fields respectively.
- To disable pin protection for the token, select **Unset Pin**.

When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

At any time, you can discard the changes by clicking the **Reset** button.

5. After making the required changes, click **Save**.

   A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.

6. Stop the administration server by clicking **Stop** in the Common Tasks pane.
7. Start the administration server, by running the following command:

   ```
   > $INSTANCE_HOME/admin-server/bin/startserv
   ```

8. At the prompt to enter the token pin, enter the pin that you specified in step 4.

Setting the PKCS#11 Token Pin for the Administration Server Using the CLI

1. Run the `set-token-pin` command, as shown in the following example:

   ```
   tadm> set-token-pin --config=admin-server --token=internal
   ```

   If the token is already protected with a pin, a prompt to enter the current pin is displayed. Enter the current pin, and when prompted, enter the new pin and confirm it.

2. Restart the administration server as described in Section 2.4, "Stopping and Restarting the Administration Server."

For more information about `set-token-pin`, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.1.4 Renewing Administration Server Certificates

To extend the validity of the self-signed administration server certificates, run the `renew-admin-certs` CLI command.

For example, the following command sets the expiry date of the self-signed administration server certificates to 24 months after the current date.

```
tadm> renew-admin-certs --validity=24
OTD-70216 The administration server and the administration nodes need to be
restarted for the changes to take effect.
```

If you do not specify the `--validity` option, the expiry date is set to 12 months after the current date.

The `renew-admin-certs` command also attempts to update the certificates on the running nodes that are currently accessible. If a node is offline—not running or not accessible due to network problems—during the certificates renewal process, you can subsequently *pull* the renewed certificates from the administration server by running the `renew-node-certs` command on that node.

For the renewed certificates take effect, you should restart the administration server and nodes

> **Note:** After renewing the administration server certificates, the first time you access the CLI or administration console, a message is displayed indicating that the server's identity cannot be verified because the certificate is from an untrusted source. To continue, you should choose to trust the self-signed certificate.

If the PKCS#11 token that provides the interface to the certificates database is protected with a pin (see Section 11.1.3), a prompt to enter the token pin is displayed.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.2 Configuring SSL/TLS Between Oracle Traffic Director and Clients

This section describes how you can use SSL/TLS to secure communication between clients and Oracle Traffic Director instances. The information in this section is aimed at readers who are familiar with the concepts of SSL/TLS, certificates, ciphers, and keys. For basic information about those concepts, see Section 11.2.7, "SSL/TLS Concepts."

This section contains the following subsections:

- Section 11.2.1, "Overview of the SSL/TLS Configuration Process"
- Section 11.2.2, "Configuring SSL/TLS for a Listener"
- Section 11.2.3, "Associating Certificates with Virtual Servers"
- Section 11.2.4, "Configuring SSL/TLS Ciphers for a Listener"
- Section 11.2.5, "Certificate-Selection Logic"
- Section 11.2.6, "About Strict SNI Host Matching"
- Section 11.2.7, "SSL/TLS Concepts"

11.2.1 Overview of the SSL/TLS Configuration Process

To enable SSL/TLS for an Oracle Traffic Director instance, you must associate an RSA or ECC certificate, or both, with one more listeners of the instance. Additionally, you can associate an RSA or ECC certificate, or both, directly with virtual servers. The process of configuring SSL/TLS for Oracle Traffic Director instances involves the following steps:

1. Obtain the required certificates, which could be self-signed, issued by a third-party Certificate Authority (CA) like VeriSign or a certificate that you generated.

   For more information, see the following sections:

   - Section 11.4.1, "Creating a Self-Signed Certificate"
   - Section 11.4.2, "Obtaining a CA-Signed Certificate"

2. Install the certificates as described in Section 11.4.3, "Installing a Certificate."
3. Associate the certificates with the required HTTP or TCP listeners as described in Section 11.2.2, "Configuring SSL/TLS for a Listener."

You can also associate certificates directly with virtual servers as described in Section 11.2.3, "Associating Certificates with Virtual Servers." For information about the logic that Oracle Traffic Director uses to select the certificate to be sent to a client during the SSL/TLS handshake, see Section 11.2.5, "Certificate-Selection Logic."

4. Configure ciphers supported for the HTTP or TCP listeners as described in Section 11.2.4, "Configuring SSL/TLS Ciphers for a Listener."

11.2.2 Configuring SSL/TLS for a Listener

You can configure a listener to receive HTTPS or TCP requests by using either the administration console or the CLI. Before you start, obtain the required certificates and install them as described in sections Section 11.4.1, Section 11.4.2, and Section 11.4.3.

Note:

- When you modify listeners, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- If you associate new certificates with a listener or remove previously associated certificates, for the changes to take effect, you must restart the instances. It is not sufficient to merely deploy the updated configuration.
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Configuring SSL/TLS for a Listener Using the Administration Console

To configure SSL/TLS for an HTTP or TCP listener by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to configure SSL/TLS-enabled listeners.
4. In the navigation pane, expand **Listeners** and select the listener for which you want to enable and configure SSL/TLS.

   The Listener Settings page is displayed.
5. In the SSL Settings section, select the **SSL Enabled** check box.
6. In the **RSA Certificate** and **ECC Certificate** fields, select the certificates that you want to use to authenticate the server.

   If you associate a listener with an RSA certificate *and* with an ECC certificate, the certificate that the server eventually presents to the client is determined during the

SSL/TLS handshake, based on the cipher suite that the client and the server negotiate to use.

You can also specify the following advanced SSL/TLS settings in the Advanced Settings section of the Listener Settings page:

- Enable and disable settings for client authentication. For more information, see Section 11.8, "Configuring Client Authentication."
- Enable and disable strict SNI host matching. For more information, see the Section 11.2.6, "About Strict SNI Host Matching." section.
- Enable and disable the following TLS-specific features:
  - **Version Rollbacks**

    Select this check box if you want Oracle Traffic Director to detect and block attempts at rolling back the TLS version. For example, if the client requested TLS 1.0, but an attacker changed it to a lower version (say, SSL 3.0), Oracle Traffic Director can detect and block the rollback even if it supports the lower version.
  - **Session Ticket Extension**

    If enabled, TLS sessions can be resumed without storing the session state of each client on the server. Oracle Traffic Director encapsulates the session state of each client in a ticket and forwards the ticket to the client. The client can subsequently resume the TLS session by using the previously obtained session ticket.
- Enable and disable SSL and TLS ciphers. For more information, see Section 11.2.4, "Configuring SSL/TLS Ciphers for a Listener."

On-screen help and prompts are provided for all of the parameters.

When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.

   A message, confirming that the updated listener was saved, is displayed in the Console Messages pane.
8. Click the **Deployment Pending** button that is displayed at the top of the main pane, and on the resulting dialog box, confirm the deployment by clicking **Deploy**.
9. Restart the instances of the configuration by clicking **Start/Restart Instances** in the Common Tasks pane.

Configuring SSL/TLS for a Listener Using the CLI

- To view the SSL/TLS properties of an HTTP or TCP listener, run the `get-ssl-prop` command, as shown in the following example:

```
tadm> get-ssl-prop --config=soa --http-listener=ls1
enabled=false
strict-sni-vs-host-match=false
client-auth=false
tls=true
max-client-auth-data=1048576
tls-session-tickets-enabled=false
ssl3=true
```

```
tls-rollback-detection=true
client-auth-timeout=60
```

- To configure SSL/TLS for an HTTP or TCP listener, run the `set-ssl-prop` command, as shown in the following example:

  ```
  tadm> set-ssl-prop --config=soa --http-listener=ls1 enabled=true
  server-cert-nickname=rsa-cert1
  OTD-70201 Command 'set-ssl-prop' ran successfully.
  ```

  For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by running the `deploy-config` command, and restart the instances by running the `restart-instance` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Note: When you enable SSL/TLS for an HTTP or TCP listener, initialization of PKCS#11 cryptographic tokens for the certificates database in the configuration is enabled automatically. For more information about configuring PKCS#11 tokens, see Section 11.5, "Managing PKCS#11 Tokens."

11.2.3 Associating Certificates with Virtual Servers

You can associate one RSA and one ECC certificate with each virtual server, by using either the administration console or the CLI. For information about the logic that Oracle Traffic Director uses to select the certificate to be sent to a client during the SSL/TLS handshake, see Section 11.2.5, "Certificate-Selection Logic."

Before you start, obtain the required certificates and install them as described in sections Section 11.4.1, Section 11.4.2, and Section 11.4.3.

Note:

- When you modify virtual servers, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- If you associate new certificates with a virtual server or remove previously associated certificates, for the changes to take effect, you must restart the instances. It is not sufficient to merely deploy the updated configuration.
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Associating Certificates with Virtual Servers Using the Administration Console

To associate certificates with virtual servers by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."

2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration for which you want to associate certificates with virtual servers.

4. In the navigation pane, expand **Virtual Servers** and select the virtual server for which you want to associate certificates.

   The Virtual Server Settings page is displayed.

5. Go to the **Certificates** section of the Virtual Server Settings page.

6. In the **RSA Certificate** and **ECC Certificate** fields, select the certificates that you want to use to authenticate the server.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.

   A message, confirming that the updated listener was saved, is displayed in the Console Messages pane.

8. Click the **Deployment Pending** button that is displayed at the top of the main pane, and on the resulting dialog box, confirm the deployment by clicking **Deploy**.

9. Restart the instances of the configuration by clicking **Start/Restart Instances** in the Common Tasks pane.

Associating Certificates with Virtual Servers Using the CLI

- To view the certificates that are currently associated with a virtual server, run the `get-virtual-server-prop` command, as shown in the following example:

```
tadm> get-virtual-server-prop --config=soa --vs=soa.example.com
server-cert-nickname
cert-rsa-soa
```

- To associate a certificate with a virtual server, run the `set-virtual-server-prop` command, as shown in the following example:

```
tadm> set-virtual-server-prop --config=soa --vs=soa.example.com
server-cert-nickname=cert-ecc-soa
OTD-70201 Command 'set-virtual-server-prop' ran successfully.
```

  For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by running the `deploy-config` command, and restart the instances by running the `restart-instance` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Note:

- If you associate a virtual server with an RSA certificate *and* with an ECC certificate, the certificate that the server eventually sends to the client is determined during the SSL/TLS handshake, based on the cipher suite that the client and the server negotiate to use.
- Make sure that a certificate of the same type—ECC or RSA—that you want to associate with the virtual server, is also associated with the listeners to which the virtual server is bound.

11.2.4 Configuring SSL/TLS Ciphers for a Listener

During the SSL/TLS handshake, the client and server inform each other about the SSL and TLS ciphers that they support and then negotiate the cipher—typically, the strongest—that they will use for the SSL/TLS session. For basic conceptual information about ciphers, see "About Ciphers".

You can configure the ciphers that Oracle Traffic Director supports for a listener by using either the administration console or the CLI.

Configuring Ciphers for a Listener Using the Administration Console

To configure the ciphers supported for an HTTP or TCP listener by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to configure ciphers.
4. In the navigation pane, expand **Listeners** and select the listener for which you want to configure ciphers.

   The Listener Settings page is displayed.
5. Go to the **Advanced Settings** section of the page and scroll down to the **SSL** subsection.
6. In the **SSL3/TLS Ciphers** field, select the check boxes for the ciphers that you want to enable for the listener, and deselect the check boxes for the ciphers that you want to disable.
7. After making the required changes, click **Save**.

   A message, confirming that the updated listener was saved, is displayed in the Console Messages pane.
8. Click the **Deployment Pending** button that is displayed at the top of the main pane, and on the resulting dialog box, confirm the deployment by clicking **Deploy**.
9. For the cipher changes to take effect, restart the instances of the configuration by clicking **Start/Restart Instances** in the Common Tasks pane.

Configuring Ciphers for a Listener Using the CLI

- To view the ciphers that are currently enabled for an HTTP or TCP listener, run the `list-ciphers` command, as shown in the following example:

  ```
  tadm> list-ciphers --config=soa
  --http-listener=http-listener-1|--tcp-listener=tcp-listener-1
  ```

  This command returns a list of all the ciphers that Oracle Traffic Director supports and indicates whether they are enabled for the listener.

- To enable specific ciphers for a listener, run the `enable-ciphers` command.

  For example, the following command enables two additional ciphers—`TLS_RSA_WITH_AES_128_CBC_SHA` and `TLS_RSA_WITH_AES_256_CBC_SHA`—for the listener `http-listener-1` in the configuration `soa`.

  ```
  tadm> enable-ciphers --config=soa --http-listener=http-listener-1 TLS_RSA_WITH_
  AES_128_CBC_SHA TLS_RSA_WITH_AES_256_CBC_SHA
  ```

- To disable ciphers for a listener, run the `disable-ciphers` command, as shown in the following example:

  ```
  tadm> disable-ciphers --config=soa --http-listener=http-listener-1  TLS_RSA_
  WITH_AES_256_CBC_SHA
  OTD-70201 Command 'disable-ciphers' ran successfully.
  ```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

Cipher Suites Supported by Oracle Traffic Director

During the SSL/TLS handshake, Oracle Traffic Director and clients negotiate the cipher suites to be used. Table 11–1 lists the cipher suites supported in Oracle Traffic Director. You can view this list by running the `list-ciphers` CLI command, as described earlier in this section. The name of each cipher suite indicates the key-exchange algorithm, the hashing algorithm, and the encryption algorithm, as depicted in.

- **Protocols supported**
  - `TLS`: TLS 1.0, 1.1, and 1.2
  - `SSL`: SSL 3 and TLS 1.0, 1.1, and 1.2
- **Key exchange algorithms supported**
  - `RSA`
  - `RSA_EXPORT`
  - `RSA_EXPORT1024`
  - `RSA_FIPS`
  - `ECDHE_RSA`
  - `ECDH_RSA`
  - `ECDH_ECDSA`
  - `ECDHE_ECDSA`

- **Encryption algorithms supported**
  - `AES_256_CBC`: 256-bit key
  - `CAMELLIA_256_CBC`: 256-bit key
  - `3DES_EDE_CBC`: 168-bit key
  - `AES_128_CBC`: 128-bit key
  - `CAMELLIA_128_CBC`: 128-bit key
  - `RC4_128`: 128-bit key
  - `SEED_CBC`: 128-bit key
  - `DES_CBC`: 56-bit key
  - `RC4_56`: 56-bit key
  - `RC4_40` and `RC2_CBC_40`: 128-bit key but only 40 bits have cryptographic significance
  - `NULL`: No encryption
- **Message Authentication Code (MAC) algorithms supported**
  - `SHA`: 160-bit hash
  - `MD5`: 128-bit hash
  - `NULL`: No hashing

Table 11–1 Cipher Suites Supported in Oracle Traffic Director

Cipher Suite	Exportable?
SSL_RSA_WITH_RC4_128_SHA	
SSL_RSA_WITH_3DES_EDE_CBC_SHA	
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA	
TLS_ECDH_RSA_WITH_RC4_128_SHA	
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA	
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA	
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA	
TLS_ECDH_ECDSA_WITH_RC4_128_SHA	
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_ SHA	
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA	
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_ SHA256	
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_ SHA256	
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA	
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_ SHA	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	

Table 11–1 (Cont.) Cipher Suites Supported in Oracle Traffic Director

Cipher Suite	Exportable?
TLS_ECDHE_RSA_WITH_RC4_128_SHA	
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	
TLS_RSA_WITH_AES_128_CBC_SHA	
TLS_RSA_WITH_AES_128_CBC_SHA256	
TLS_RSA_WITH_AES_128_GCM_SHA256	
TLS_RSA_WITH_AES_256_CBC_SHA	
TLS_RSA_WITH_AES_256_CBC_SHA256	
TLS_RSA_WITH_SEED_CBC_SHA	
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA	
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA	

Source for FIPS 40-compliance information:
http://www.mozilla.org/projects/security/pki/nss/ssl/fips-ssl-ciphersuites.html

11.2.5 Certificate-Selection Logic

When an HTTPS request is received, the certificate that Oracle Traffic Director sends to the client during the SSL/TLS handshake could be one of the following:

- A certificate associated with a virtual server bound to a configured HTTP/TCP listener
- A certificate associated with the default virtual server of the listener
- A certificate associated with the listener

Oracle Traffic Director uses the following logic to determine the certificate that should be sent to the client during the SSL/TLS handshake.

Table 11–2 Certificate-Selection Logic

Condition	Case A	Case B	Case C	Case D
Client sent SNI host extension	Yes	Yes	Yes	No
A *matching*[1] virtual server is found.	Yes	No	No	--
The matching virtual server has a certificate of a type—RSA or ECC— that matches the ciphers sent by the client.	Yes	--	--	--
The default virtual server of the listener has a certificate of a type—RSA or ECC— that matches the ciphers sent by the client.	--	Yes	--	--
The listener has a certificate of a type—RSA or ECC— that matches the ciphers sent by the client.	--	--	Yes	Yes

Table 11–2 (Cont.) Certificate-Selection Logic

Condition	Case A	Case B	Case C	Case D
Certificate selected:	Certificate of the matching virtual server	Certificate of the default virtual server	Certificate of the listener	Certificate of the listener

[1] A *matching* virtual server is a virtual server that is bound to the listener and has a host pattern that matches the `Host:` header sent by the client.

11.2.6 About Strict SNI Host Matching

When a client sends an HTTPS request to an SSL/TLS-enabled Oracle Traffic Director instance, the server needs to send a certificate to the client to initiate the SSL/TLS handshake. If the host name in the request does not match the server name (common name, CN) in the certificate provided by the server, a warning message is displayed by the client to the user. To continue with the SSL/TLS handshake process, the user must then explicitly choose to trust the certificate.

If an Oracle Traffic Director instance contains multiple, name-based virtual servers configured with a single IP address and port combination, to determine the appropriate certificate that should be sent to the client, the server needs to know the value of the `Host` header in the HTTP request, which it cannot read until after the SSL/TLS connection is established.

An extension to the SSL and TLS protocols, called Server Name Indication (SNI), addresses this issue, by allowing clients to provide the requested host name during the SSL/TLS handshake in the SNI host extension. Oracle Traffic Director uses the host name in the SNI host extension to determine the virtual server certificate that it should send to the client. For information about associating certificates with virtual servers, see Section 11.2.3, "Associating Certificates with Virtual Servers."

Support for SNI is enabled by default for SSL/TLS-enabled HTTP listeners in Oracle Traffic Director. For stricter control, like if you want to prevent non-SNI clients from accessing name-based virtual servers, you should enable the Strict SNI Host Matching parameter.

When Strict SNI Host Matching is enabled for an HTTP listener, and if for that listener at least one of the virtual servers has certificates, then Oracle Traffic Director returns a `403-Forbidden` error to the client, if any of the following conditions is true:

- The client did not send the SNI host extension during the SSL/TLS handshake.
- The request does not have the `Host:` header.
- The host name sent by the client in the SNI host extension during the SSL/TLS handshake does not match the `Host:` header in the request.

11.2.7 SSL/TLS Concepts

This section provides basic information about security-related concepts. It contains the following topics:

- About SSL
- About Ciphers
- About Keys
- About Certificates

About SSL

Secure Socket Layer (SSL) is a protocol for securing Internet communications and transactions. It enables secure, confidential communication between a server and clients through the use of digital certificates. Oracle Traffic Director supports SSL v3 and Transport Layer Security (TLS) v1.

In a 2-way HTTP over SSL (HTTPS) connection, each party—say a browser or a web server—first verifies the identity of the other. This phase is called the SSL/TLS handshake. After the identities are verified, the connection is established and data is exchanged in an encrypted format. The following are the steps in the SSL/TLS handshake between an SSL-enabled browser and an SSL-enabled server:

1. The browser attempts to connect to the server by sending a URL that begins with `https://` instead of `http://`.
2. The server sends its digital certificate (see "About Certificates") and public key to the client.
3. The client checks whether the server's certificate is current (that is, it has not expired) and is issued by a certificate authority (CA) that the client trusts.
4. If the certificate is valid, the client generates a one-time, unique session key and encrypts it with the server's public key, and then sends the encrypted session key to the server.
5. The server decrypts the message from the client by using its private key and retrieves the session key.

At this point, the client has verified the identity of the server; and only the client and the server have a copy of the client-generated, unique session key. Till the session is terminated, the client and the server use the session key to encrypt all communication between them.

About Ciphers

A cipher is an algorithm, a mathematical function, used for encrypting and decrypting data. Some ciphers are stronger and more secure than others. Usually, the more bits a cipher uses, the harder it is to decrypt the data encrypted using that cipher.

SSL v3 and TLS v1 support a variety of ciphers. Clients and servers may support different *cipher suites* (sets of ciphers), depending on factors such as the protocol they support, the organizational policies on encryption strength, and government restrictions on export of encrypted software.

In any 2-way encryption process, the client and the server must use the same cipher suite. During the SSL/TLS handshake process, the server and client negotiate the cipher suite—typically, the strongest one—that they will use to communicate.

About Keys

Encryption using ciphers, by itself, does not ensure data security. A key must be used with the encrypting cipher to produce the actual encrypted result, or to decrypt previously encrypted information. The encryption process uses two keys—a public key and a private key. The keys are mathematically related; so information that is encrypted using a public key can be decrypted only using the associated private key, and vice versa. The public key is published by the owner as part of a certificate (see "About Certificates"); only the associated private key is safeguarded.

About Certificates

A certificate is a collection of data that uniquely identifies a person, company, or other entity on the Internet. It enables secure, confidential communication between two

entities. Personal certificates are used by individuals; server certificates are used to establish secure sessions between the server and clients over SSL.

Certificates can be self-signed (by the server), signed by a trusted third party called Certification Authority (CA) or one that you created. The holder of a certificate can present the certificate as proof of identity to establish encrypted, confidential communication. The CA could be a third-party vendor or an internal department responsible for issuing certificates for an organization's servers.

Certificates are based on public-key cryptography, which uses a pair of *keys* (very long numbers) to encrypt information so that it can be read only by its intended recipient. The recipient then decrypts the information using one of the keys.

A certificate binds the owner's public key to the owner's identity. In addition to the public key, a certificate typically includes information such as the following:

- The name of the holder and other identification, such as the URL of the server using the certificate
- The name of the CA that issued the certificate
- The digital signature of the issuing CA
- The validity period of the certificate

11.3 Configuring SSL/TLS Between Oracle Traffic Director and Origin Servers

This section describes how to use SSL/TLS to secure connections between Oracle Traffic Director instances and origin servers that are Oracle WebLogic Server and Oracle HTTP Server instances. It contains the following topics:

- Section 11.3.1, "About One-Way and Two-Way SSL/TLS"
- Section 11.3.2, "Configuring One-Way SSL/TLS Between Oracle Traffic Director and Origin Servers"
- Section 11.3.3, "Configuring Two-Way SSL/TLS Between Oracle Traffic Director and Origin Servers"

11.3.1 About One-Way and Two-Way SSL/TLS

The connections between Oracle Traffic Director and origin servers in the back end can be secured using one-way or two-way SSL/TLS.

- **One-way SSL/TLS**: The SSL/TLS-enabled origin server presents its certificate to the Oracle Traffic Director instance. The Oracle Traffic Director instance is not configured to present any certificate to the origin server during the SSL/TLS handshake.
- **Two-way SSL/TLS**: The SSL/TLS-enabled origin server presents its certificate to the Oracle Traffic Director instance. The Oracle Traffic Director instance too presents its own certificate to the origin server. The origin server verifies the identity of the Oracle Traffic Director instance before establishing the SSL/TLS connection. Additionally, either end of the SSL/TLS connection—Oracle Traffic Director and/or origin servers—can be configured to verify the host name while exchanging certificates.

11.3.2 Configuring One-Way SSL/TLS Between Oracle Traffic Director and Origin Servers

To configure one-way SSL/TLS between Oracle Traffic Director and origin servers, you must export the origin servers' certificates in PKCS#12 format, install them in the certificate database of Oracle Traffic Director, and, optionally, configure Oracle Traffic Director to trust the certificates.

Note:

- The procedure described in this section is for a scenario where all of the servers in the origin-server pool use certificates issued by the same CA. In such a scenario, you can configure one-way SSL/TLS by importing just the root certificate of the CA that signed the certificates for the origin servers into the certificates database of Oracle Traffic Director.
- If the origin servers use self-signed certificates or certificates issued by different CAs, you should individually export and import each of the server certificates or the individual root certificates of the CAs that signed the server certificates.
- If the `WebLogic Server Plug-In Enabled` attribute, which can be accessed using the Weblogic Server administration console, is set to `true` and when Oracle Traffic Director terminates an SSL connection, Oracle Traffic Director communicates the certificate information to the applications deployed on the WebLogic Server. An application can then validate for specific information in the certificate, such as key size or cipher, before allowing the clients to access the application.

1. Export the root certificate of the CA that issued certificates to the origin servers into the PKCS#12 format.
   - **For Oracle WebLogic Server origin servers**:

     Use the `keytool` command available in Java SE 6.

     Syntax:

     ```
     > $JAVA_HOME/bin/keytool -exportcert -alias alias -file file -keystore
     keystore -storepass storepass -rfc
     ```

     `alias` is the nickname of the certificate to be exported, `file` is the name for the exported certificate, `keystore` is the name of the custom Oracle WebLogic Server identity store file, and `storepass` is the password for the specified keystore.

     Example:

     ```
     > $JAVA_HOME/bin/keytool -exportcert -alias wlsos1 -file wls_os_cert
      -keystore $DOMAIN_HOME/soa_domain/soa_keystore.jks -storepass stpass -rfc
     ```

     For more information about `keytool`, see the documentation at:

     http://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html

   - **For Oracle HTTP Server origin servers**:

     Use the `exportWalletObject` WebLogic Scripting Tool (WLST) command.

Syntax:

```
exportWalletObject(instName, compName, compType, walletName, password,
type, path, DN)
```

Example:

```
> exportWalletObject('inst1', 'ohs1', 'ohs','wallet1', 'password',
'Certificate', '/tmp','cn=soa.example.com')
```

This command exports the certificate with the DN `cn=soa.example.com` from the wallet `wallet1`, for Oracle HTTP Server instance `ohs1`. The trusted certificate is exported to the directory `/tmp`.

For more information about the `exportWalletObject` command, see the documentation at http://docs.oracle.com/cd/E21764_01/web.1111/e13813/custom_infra_security.htm#CDDFDHDH.

2. Install the root certificate, which you just exported, in the certificates database of Oracle Traffic Director by using the `install-cert` CLI command.

Note: For information about installing a certificate using the Administration Console, see Section , "Installing a Self-signed or CA-signed Certificate Using the Administration Console."

Syntax:

```
tadm> install-cert --config=config_name --token=name --cert-type=ca
--nickname=nickname cert_file
```

Example:

```
tadm> install-cert --config=soa --token=internal --cert-type=ca
--nickname=Server-Cert os_cert
```

Note: If the origin servers use self-signed certificates or certificates issued by different CAs, do the following instead of steps 1 and 2:

1. Export each server certificate, or each root certificate of the CAs that signed the server certificates, individually, by using the same commands used in step 1.
2. Install each certificate, which you exported in the previous step, in the certificates database of Oracle Traffic Director, by using the `install-cert` CLI command, as described in step 2 but with `--cert-type=server`.
3. Configure Oracle Traffic Director to trust each of the origin servers' certificates, as described in Section 11.4.7, "Configuring Oracle Traffic Director to Trust Certificates."

---

3. If required, configure Oracle Traffic Director to verify the host name in the origin server's certificate by using the `set-route-prop` CLI command.

Syntax:

```
tadm> set-route-prop --config=config_name  --vs=vs_name --route=route_name
validate-server-cert=true
```

Example:

```
tadm> set-route-prop --config=soa  --vs=vs1 --route=route1
```

```
validate-server-cert=true
```

To view a list of the virtual servers in a configuration and the routes defined for a virtual server, use the `list-virtual-servers` and `list-routes` CLI commands, respectively.

Note: If you choose to configure Oracle Traffic Director to validate the host name in the origin server's certificate during the SSL/TLS handshake, then you must do the following:

- Ensure that the server name (CN) in the origin server's certificate matches the origin server's host name as specified in the origin-server pool of the Oracle Traffic Director configuration. For more information about configuring origin-server pools, see Chapter 6, "Managing Origin-Server Pools."
- Ensure that dynamic discovery is disabled (default setting). For more information about dynamic discovery, see Section 6.5, "Configuring an Oracle WebLogic Server Cluster as an Origin-Server Pool."

Otherwise, when the origin server presents its certificate, Oracle Traffic Director cannot validate the host name in the certificate, and so the SSL/TLS handshake will fail.

---

4. Deploy the updated configuration to the Oracle Traffic Director instances by using the deploy-config CLI command.

```
tadm> deploy-config config_name
```

Note: For more information, about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.3.3 Configuring Two-Way SSL/TLS Between Oracle Traffic Director and Origin Servers

To configure two-way SSL/TLS between Oracle Traffic Director and origin servers, do the following:

Note: For more information, about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

---

1. Perform the procedure for configuring one-way SSL/TLS, as described in Section 11.3.2, "Configuring One-Way SSL/TLS Between Oracle Traffic Director and Origin Servers."
2. Obtain a CA-issued server certificate for Oracle Traffic Director, as described in Section 11.4.2, "Obtaining a CA-Signed Certificate."
3. Install the CA-issued server certificate in the Oracle Traffic Director configuration, as described in Section 11.4.3, "Installing a Certificate.".

4. Configure the required Oracle Traffic Director route with the certificate that Oracle Traffic Director should present to the origin server, by using the `enable-route-auth` CLI command.

   Syntax:

   ```
   tadm> enable-route-auth --config=config_name --vs=vs_name --route=route_name
   --client-cert-nickname=cert_nickname
   ```

   Example:

   ```
   tadm> enable-route-auth --config=soa --vs=vs1 --route=route1
   --client-cert-nickname=soa_cert
   ```

   To view a list of the virtual servers in a configuration and the routes defined for a virtual server, use the `list-virtual-servers` and `list-routes` CLI commands, respectively.

5. Deploy the updated configuration to the Oracle Traffic Director instances by using the `deploy-config` CLI command.

   ```
   tadm> deploy-config config_name
   ```

6. Export the root certificate of the CA that signed the certificate for the Oracle Traffic Director instance, from the Oracle Traffic Director certificates database to the `.pem` format.

   Syntax:

   ```
   > $ORACLE_HOME/bin/certutil -L -d certdir -n nickname -a -o output_cert_file
   ```

   `certdir` is the full path to the `config` directory of the Oracle Traffic Director instance from which you want to export the root CA certificate, `nickname` is the nickname of the certificate that you want to export, and `output_cert_file` is the name of the `.pem` file to which the certificate should be exported.

   Example:

   ```
   > $ORACLE_HOME/bin/certutil -L -d ../net-test/config/ -n rootca -a -o
   /tmp/rootca1.pem
   ```

   For more information about the `certutil` command, run the following command:

   ```
   > $ORACLE_HOME/bin/certutil -H
   ```

7. Import the root certificate that you exported in the previous step into the trust keystore for the Oracle WebLogic Server origin servers (and the Oracle wallet for Oracle HTTP Server origin servers).

   - **For Oracle WebLogic Server origin servers**:

     Use the `keytool` command available in Java SE 6.

     Syntax:

     ```
     > $JAVA_HOME/bin/keytool -importcert -v -trustcacerts -alias alias
      -file cert_file -keystore keystore_file -storepass keystore_password
      -noprompt
     ```

     `alias` is the nickname of the CA-issued root CA exported in the previous step, `file` is the name of the exported `.pem` certificate file, `keystore` is the name of the custom Oracle WebLogic Server identity store file, and `storepass` is the password for the specified keystore.

Example:

```
> $JAVA_HOME/bin/keytool -importcert -v -trustcacerts -alias rootca1
 -file /tmp/rootca1.pem -keystore $DOMAIN_HOME/soa_domain/soa_keystore.jks
 -storepass stpass -noprompt
```

For more information about `keytool`, see the documentation at:

http://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html

- **For Oracle HTTP Server origin servers**:

  Use the `importWalletObject` WLST command.

  Syntax:

  ```
  importWalletObject(instName, compName, compType, walletName, password,
  type, filePath)
  ```

  Example:

  ```
  > importWalletObject('inst1', 'ohs1', 'ohs','wallet1', 'password',
  'TrustedCertificate','/tmp/rootca1.pem')
  ```

  For more information about the `importWalletObject` command, see the documentation at http://docs.oracle.com/cd/E21764_01/web.1111/e13813/custom_infra_security.htm#CDDGIBEJ.

8. Configure the origin servers to require Oracle Traffic Director to present its client certificate during the SSL/TLS handshake.

   - **For Oracle WebLogic Server origin servers**:

     Perform the procedure described in "Configure two-way SSL" in the *Oracle WebLogic Server Administration Console Online Help* at http://docs.oracle.com/cd/E21764_01/apirefs.1111/e13952/taskhelp/security/ConfigureTwowaySSL.html.

     Note: By default, host name verification is enabled in Oracle WebLogic Server. For information about disabling host name verification, see "Disable host name verification" in the *Oracle WebLogic Server Administration Console Online Help* at http://docs.oracle.com/cd/E21764_01/apirefs.1111/e13952/taskhelp/security/DisableHostNameVerification.html.

   - **For Oracle HTTP Server origin servers**:

     Add the following directive in the `httpd.conf` file.

     ```
     SSLVerifyClient require
     ```

11.4 Managing Certificates

This section contains the following topics:

- Section 11.4.1, "Creating a Self-Signed Certificate"
- Section 11.4.2, "Obtaining a CA-Signed Certificate"
- Section 11.4.3, "Installing a Certificate"

- Section 11.4.4, "Viewing a List of Certificates"
- Section 11.4.5, "Renewing a Server Certificate"
- Section 11.4.6, "Deleting a Certificate"
- Section 11.4.7, "Configuring Oracle Traffic Director to Trust Certificates"

Note:

- The information in this section is aimed at readers who are familiar with the concepts of SSL, certificates, ciphers, and keys. For basic information about those concepts, see Section 11.2.7, "SSL/TLS Concepts."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

11.4.1 Creating a Self-Signed Certificate

You can create a self-signed certificate if you do not need your certificate to be signed by a CA, or if you want to test the SSL/TLS implementation while the CA is in the process of signing your certificate.

Note that if you use a self-signed certificate to enable SSL/TLS for an Oracle Traffic Director virtual server, when a client accesses the `https://` URL of the virtual server, an error message is displayed indicating that the signing CA is unknown and not trusted. To proceed with the connection, the client can choose to trust the self-signed certificate.

You can create a self-signed certificate by using either the administration console or the CLI.

Before You Begin

Before you begin creating a self-signed certificate or a certificate-signing request, decide the following:

- The fully qualified host name used in DNS lookups (for example, `www.example.com`).

  If the host name in the client request does not match the name on the certificate, the client is notified that the certificate server name does not match the host name.

  Note: In a high availability scenario, ensure that the server name (CN) in the server's certificate matches the host name of the VIP that the OTD instance is configured to listen on.

- The nickname of the certificate (required only for creating a self-signed certificate).
- The certificate's validity period, in months (required only for creating a self-signed certificate).
- The key type—RSA or ECC.

  Oracle Traffic Director supports generation of the traditional RSA-type keys and the more advanced Elliptic Curve Cryptography (ECC) keys. ECC offers equivalent security with smaller key sizes, which results in faster computations, lower power consumption, and memory and bandwidth savings.

- The key size (for RSA) or curve (for ECC).

  For RSA keys, you can specify 1024, 2048, 3072, or 4096 bits. Long keys provide better encryption, but Oracle Traffic Director would need more time to generate them.

  For ECC keys, you should specify the curve for generating the key pair. Oracle Traffic Director supports the following curves: prime256v1, secp256r1, nistp256, secp256k1, secp384r1, nistp384, secp521r1, nistp521, sect163k1, nistk163, sect163r1, sect163r2, nistb163, sect193r1, sect193r2, sect233k1, nistk233, sect233r1, nistb233, sect239k1, sect283k1, nistk283, sect283r1, nistb283, sect409k1, nistb409, sect571k1, nistk571, sect571r1, nistb571, secp160k1, secp160r1, secp160r2, secp192k1, secp192r1, nistp192, secp224k1, secp224r1, nistp224, prime192v1.

Creating a Self-Signed Certificate Using the Administration Console

To create a self-signed certificate by using the administration console, do the following:

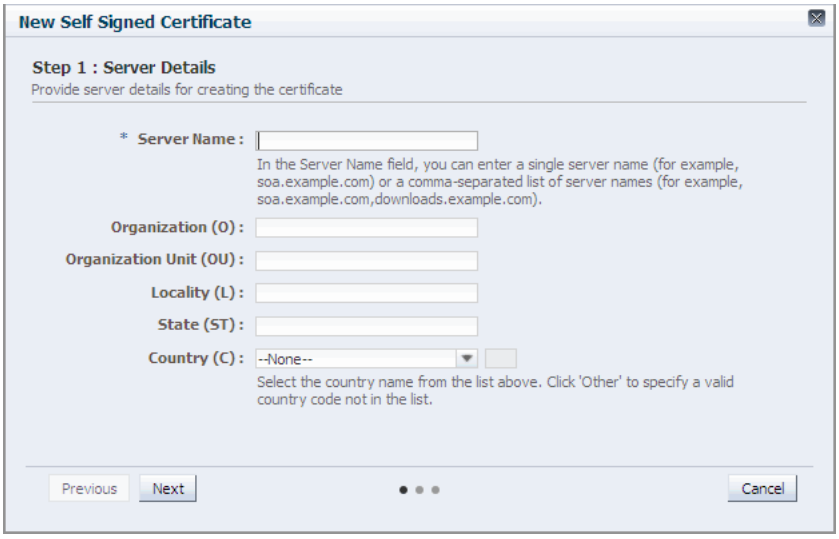
1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create an self-signed certificate.
4. In the navigation pane, expand **SSL** and select **Server Certificates**.

   The Server Certificates page is displayed.
5. Click the **New Self-Signed Certificate** button.

   The New Self-Signed Certificate wizard starts.

Figure 11–1 New Self-Signed Certificate Wizard



New Self-Signed Certificate wizard

****

Note: If the PKCS#11 token, in which the certificates and keys for the configuration are stored, is protected by a pin, the first screen of the wizard displays a prompt to select the token and enter the pin.

1. Select the appropriate token.

   If the keys are stored in the local key database maintained by Oracle Traffic Director, select the **internal** token.

   If the keys are stored in a Smart Card, or in an external device or engine, select the name of that external token.

2. Enter the pin for the selected token.

To avoid having to enter token pins repeatedly during an administration-console session, you can cache the pins as described in "Caching the Token Pins for an Administration Console Session".

6. Follow the on-screen prompts to complete creation of the certificate by using the details—server name, certificate nickname, validity, key type, and so on—that you decided earlier.

   After the certificate is created, the Results screen of the New Self-Signed Certificate wizard displays a message confirming successful creation of the certificate.

7. Click **Close**.

   - A message is displayed in the Console Message pane confirming that the certificate was created.
   - The certificate that you created is displayed on the Server Certificates page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Creating a Self-Signed Certificate Using the CLI

To create a self-signed certificate, run the `create-selfsigned-cert` command, as shown in the following example:

```
tadm> create-selfsigned-cert --config=soa --server-name=soa.example.com
 --nickname=cert-soa
OTD-70201 Command 'create-selfsigned-cert' ran successfully.
```

This command creates a self-signed certificate that is valid for a default period of 12 months with the nickname `cert-soa` for the server `soa.example.com` in the configuration `soa`. The key type and other parameters were not specified; so the command creates the certificate with RSA-type (default) keys that are 2048 bits (default) long.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information, about `create-selfsigned-cert`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

11.4.2 Obtaining a CA-Signed Certificate

To obtain a certificate signed by a Certificate Authority (CA), you should submit a *Certificate Signing Request* (*CSR*) to the CA, pay the prescribed fee if required, and wait for the CA to approve the request and grant the certificate.

The CSR is a digital file—a block of encrypted text in Base-64 encoded PEM format—containing information such as your server name, organization name, and country. It also contains the public key that will be included in the certificate.

You can create a CSR by using either the administration console or the CLI of Oracle Traffic Director.

Before You Begin

Before you begin creating a CSR, decide the server name; key type; and key size (for RSA) or curve (for ECC), as described in Section 11.4.1, "Creating a Self-Signed Certificate.".

> **Note:** In a high availability scenario, ensure that the server name (CN) in the server's certificate matches the host name of the VIP that the OTD instance is configured to listen on.

Creating a CSR Using the Administration Console

To create a CSR by using the administration console, do the following:

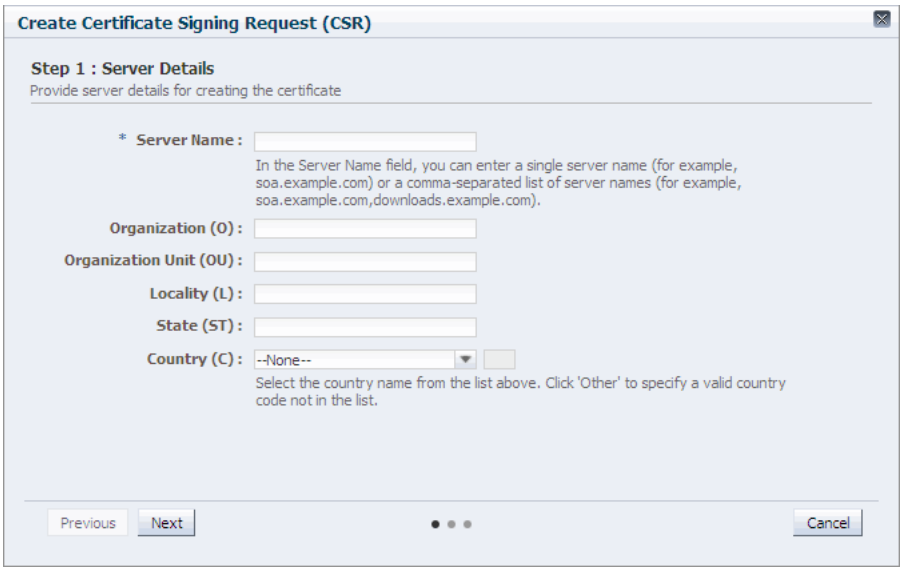
1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create a CSR.
4. In the navigation pane, expand **SSL** and select **Server Certificates**.

   The Server Certificates page is displayed.
5. Click the **Create Certificate Request** button.

   The Create Certificate Signing Request wizard starts.

Figure 11–2 Create Certificate Signing Request Wizard



Create Certificate Signing Request wizard

****

Note: If the PKCS#11 token in which the certificates and keys for the configuration are stored is protected by a pin, the first screen of the wizard displays a prompt to select the token and enter the pin.

1. Select the appropriate token.

   If the keys are stored in the local key database maintained by Oracle Traffic Director, select the **internal** token.

   If the keys are stored in a Smart Card, or in an external device or engine, select the name of that external token.

2. Enter the pin for the selected token.

To avoid having to enter token pins repeatedly during an administration console session, you can cache the pins as described in "Caching the Token Pins for an Administration Console Session".

6. Follow the on-screen prompts to complete creation of the CSR by using the details—server name, key type, and so on—that you decided earlier.

   After the CSR is created, the Results screen of the Create Certificate Signing Request wizard displays the encrypted text of the CSR as shown in the following example:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
 MIICmDCCAYACAQAwDDEKMAgGA1UEAxMBeTCCASIwDQYJKoZIhvcNAQEBBQADggEP
 ADCCAQoCggEBAMBzgU1mQJrQYQOiedKVpQVedJplQT1gh943RfNfCsl6VbD1Kid8
 ...
 lines deleted
 ...
 v6PWA9azqAfnJ8IriK6xTMQ54oQNzSALEKvIGb+jBUUzo2S+UiEr+VXvfPAdHnPX
 2ZBCA4qvPr477lETgPphfxDjjvvH+EKrZMClM4JkJ4g3p+X0X+5vz53w964=
 -----END NEW CERTIFICATE REQUEST-----
```

7. Copy and store the CSR text, *including* the header line `BEGIN NEW CERTIFICATE REQUEST` and the footer line `END NEW CERTIFICATE REQUEST`, and click **Close**.

The CSR includes the public key and other information that the CA needs to verify the identity of the Oracle Traffic Director server for which you want to enable SSL. The private key is stored in encrypted form in the `INSTANCE_HOME/net-config_name/config/key4.db` file.

You can now send the CSR with the required certificate-signing fee to a CA of your choice.

Creating a CSR Using the CLI

To create a CSR, run the `create-cert-request` command, as shown in the following example:

```
tadm> create-cert-request--config=soa --server-name=soa.example.com
 --token=internal
OTD-70201 Command 'create-selfsigned-cert' ran successfully.
```

This command creates a CSR and displays the encrypted text of the CSR as shown in "Creating a Self-Signed Certificate Using the Administration Console".

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information, about `create-cert-request`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

After obtaining the CA-signed certificate in response to your CSR, you should install the certificate in the appropriate configuration, as described in Section 11.4.3, "Installing a Certificate."

11.4.3 Installing a Certificate

You can install a self-signed or CA-signed certificate by using the administration console or the CLI. In addition, you can install an existing certificate by using the pk12util utility.

This section contains the following topics:

- Section , "Installing a Self-signed or CA-signed Certificate Using the Administration Console"
- Section , "Installing a Self-signed or CA-signed Certificate Using the CLI"
- Section , "Installing an Existing Certificate Using pk12util"

Installing a Self-signed or CA-signed Certificate Using the Administration Console

To install a self-signed or CA-signed certificate by using the administration console, do the following:

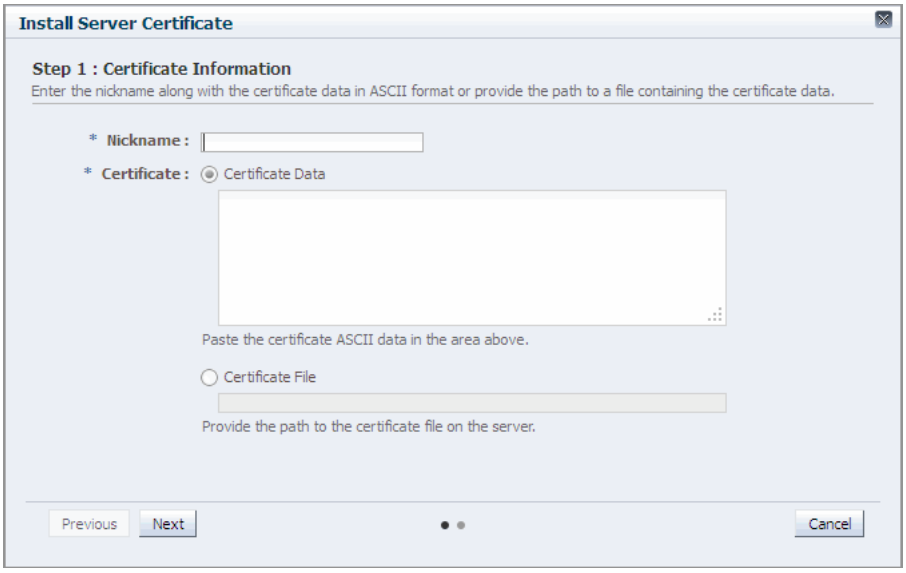
1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to install a certificate.

4. In the navigation pane, expand **SSL**, and select **Server Certificates** or **Certificate Authorities**.
   - To install self-signed or CA-signed certificates, select **Server Certificates**.
   - To install root certificates or certificate chains, select **Certificate Authorities**.
5. Click the **Install Certificate** button.

   The Install Certificate wizard or the Install Server Certificate wizard (Figure 11–3) starts, depending on whether you were on Server Certificates page or the Certificate Authorities page when you clicked the **Install Certificate** button.

Figure 11–3 Install Server Certificate Wizard



Install Server Certificate wizard

****

Note: If the PKCS#11 token in which the certificates and keys for the configuration are stored is protected by a pin, the first screen of the wizard displays a prompt to select the token and enter the pin.

1. Select the appropriate token.

   If the keys are stored in the local key database maintained by Oracle Traffic Director, select the **internal** token.

   If the keys are stored in a Smart Card, or in an external device or engine, select the name of that external token.
2. Enter the pin for the selected token.

To avoid having to enter token pins repeatedly during an administration console session, you can cache the pins as described in "Caching the Token Pins for an Administration Console Session".

6. Paste the certificate text from a `.pem` file or specify the path name of the certificate file.

If you opt to paste the certificate text, be sure to include the headers `BEGIN CERTIFICATE` and `END CERTIFICATE`, including the beginning and ending hyphens, as shown in the following example:

```
-----BEGIN CERTIFICATE-----
MIIEuTCCA6GgAwIBAgIQQBrEZCGzEyEDDrvkEhrFHTANBgkqhkiG9w0BAQsFADCB
vTELMAkGA1UEBhMCVVMxFzAVBgNVBAoTDlZlcmlTaWduLCBJbmMuMR8wHQYDVQQL
...
lines deleted
...
lRQOfc2VNNnSj3BzgXucfr2YYdhFh5iQxeuGMMY1v/D/w1WIg0vvBZIGcfK4mJO3
7M2CYfE45k+XmCpajQ==
-----END CERTIFICATE-----
```

If you opt to specify the path name, ensure that the file resides on the admin server.

7. Follow the on-screen prompts to complete installation of the certificate.

Installing a Self-signed or CA-signed Certificate Using the CLI

To install a self-signed or CA-signed certificate, run the `install-cert` command, as shown in the following example:

```
tadm> install-cert --config=soa --token=internal --cert-type=server
--nickname=soa-cert /home/admin/certs/verisign-cert.cer
```

The `--cert-type` option specifies the certificate type—server or CA. This command install the server certificate with the nickname `soa-cert` in the configuration `soa`. To install a CA certificate, specify `ca` for the `--cert-type` option. Note that the `--nickname` option is not mandatory for installing `ca` and `chain` certificate types.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `install-cert`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Installing an Existing Certificate Using pk12util

The command-line utility `pk12util` can be used to import an existing certificate and private key into an internal or external PKCS#11 module. By default, `pk12util` uses certificate and private key databases named `cert7.db` and `key3.db`.

Perform the following steps to install an existing certificate:

1. Add `ORACLE_HOME/lib` to your path.
2. Run the `pk12util` command as shown below:

```
pk12util -i importfile [-d certdir] [-P dbprefix] [-h tokenname] [-k slotpwfile
| -K slotpw] [-w p12filepwfile | -W p12filepw] [-v]
```

Note:

- Option `-P` must follow the `-h` option, and it must be the last argument.
- Enter the exact token name including capital letters and spaces between quote marks.

For example, the following command imports a PKCS12-formatted certificate into an NSS certificate database:

```
pk12util -i certandkey.p12 [-d certdir][-h "nCipher"][-P
https-jones.redplanet.com-jones-
]
```

3. Enter the database and/or token password. For more information about PKCS#11 tokens, see Section 11.5, "Managing PKCS#11 Tokens."
4. Associate the certificate that you installed with one more listeners. For more information, see Section 11.2.2, "Configuring SSL/TLS for a Listener."

11.4.4 Viewing a List of Certificates

You can view a list of the certificates installed in a configuration by using either the administration console or the CLI.

Viewing a List of Certificates Using the Administration Console

To view a list of the certificates installed in a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view certificates.
4. In the navigation pane, expand **SSL**, and select **Server Certificates** or **Certificate Authorities**.
   - To view self-signed or CA-signed certificates installed in the configuration, select **Server Certificates**.
   - To view root certificates or certificate chains, select **Certificate Authorities**.

   The resulting page displays the installed certificates.

Note: If the pin is enabled for a token in the selected configuration, the installed certificates are not displayed. Instead, a message to enter the token pins is displayed on the page.

1. Click **Cache Token Pin**.
2. In the resulting dialog box, enter the pins for the tokens, and click **OK**.

Viewing a List of Certificates Using the CLI

- To view a list of the certificates installed in a configuration, run the `list-certs` command, as shown in the following examples.
  - The following command displays a list of the server certificates in the configuration `soa`.

```
tadm> list-certs --config=soa --verbose --all
nickname        issuer-name     expiry-date
----------------------------------------
cert-adf        adf             "Aug 17, 2012 5:32:40 AM"
cert-soa        soa             "Aug 17, 2012 5:32:26 AM"
```

- The following command displays a partial list of the CA certificates that are installed in the configuration `soa`.

```
tadm> list-certs --config=soa --server-type=ca --verbose --all
nickname        issuer-name     expiry-date
----------------------------------------
"Builtin Object Token:GlobalSignRootCA" "GlobalSign" "Jan 28, 2028 4:00:00
AM"
"Builtin Object Token:GlobalSignRootCA-R2" "GlobalSign" "Dec 15, 2021
12:00:00 AM"
```

- To view the properties of a certificate, run the `get-cert-prop` command, as shown in the following example.

```
tadm> get-cert-prop --config=soa --nickname=cert-soa
nickname=cert-soa
subject="CN=soa.example.com"
server-name=soa.example.com
issuer="CN=soa.example.com"
serial-number=00:95:9C:34:04
fingerprint=34:E7:52:5E:3F:0A:EE:30:ED:BF:96:81:DD:1E:A3:02
key-type=rsa
key-size=2048
issue-date=Sep 14, 2011 12:22:41 AM
expiry-date=Sep 14, 2012 12:22:41 AM
is-expired=false
is-read-only=false
is-self-signed=true
is-user-cert=true
is-ca-cert=false
has-crl=false
```

Note: If the pin is enabled for a token in the specified configuration, a prompt to enter the token pin is displayed when you run the `list-certs` and `get-cert-prop` commands.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

11.4.5 Renewing a Server Certificate

To renew a certificate, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to renew certificates.
4. In the navigation pane, expand **SSL** and select **Server Certificates**.

   The resulting page displays the installed server certificates.

Note: If the pin is enabled for a token in the selected configuration, the installed certificates are not displayed. Instead, a message to enter the token pins is displayed on the page.

1. Click **Cache Token Pin**.
2. In the resulting dialog box, enter the pins for the tokens, and click **OK**.

5. Click the **Renew** button for the certificate that you want to renew.

   The Renew Server Certificate dialog box is displayed.
6. Specify the new validity period and click **Next**.
7. Click **Renew Certificate**.
8. Click **Close**.
   - A message is displayed in the Console Messages pane, confirming that the certificate has been renewed for the specified period.
   - The new expiry date for the certificate is displayed on the Server Certificates page.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

11.4.6 Deleting a Certificate

You can delete certificates in a configuration by using either the administration console or the CLI.

Deleting a Certificate Using the Administration Console

To delete a certificate in a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to delete certificates.
4. In the navigation pane, expand **SSL** and select **Server Certificates** or **Certificate Authorities**.
   - To delete self-signed or CA-signed certificates, select **Server Certificates**.
   - To delete root certificates or certificate chains, select **Certificate Authorities**.

   The resulting page displays the installed certificates.

Note: If the pin is enabled for a token in the selected configuration, the installed certificates are not displayed. Instead, a message to enter the token pins is displayed on the page.

1. Click **Cache Token Pin**.
2. In the resulting dialog box, enter the pins for the tokens, and click **OK**.

5. Click the **Delete** button for the certificate that you want to delete.
   - If one or more listeners are associated with the certificate that you are deleting, a message is displayed indicating that the certificate cannot be deleted.
   - If the certificate that you are deleting is not associated with any listener, a prompt to confirm deletion of the certificate is displayed.

     Click **OK** to proceed.

   A message is displayed in the Console Messages pane, confirming that the certificate has been deleted.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Deleting a Certificate Using the CLI

To delete a certificate, run the `delete-cert` command.

For example, the following command deletes the certificate with the nickname `rsa-cert-1` from the configuration `soa`.

```
tadm> delete-cert --token=internal --config=soa rsa-1
```

If the certificate that you are deleting is associated with one or more listeners, the following message is displayed.

```
OTD-64309 Certificate 'rsa-1' is being referred by listeners: listener1,listenerN
```

You can delete the certificate forcibly by including the `--force` option.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `delete-cert`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

11.4.7 Configuring Oracle Traffic Director to Trust Certificates

The built-in certificates database in Oracle Traffic Director includes several pre-installed root certificates, including those from popular commercial CAs like VeriSign. You can also use the administration console and the CLI to configure Oracle Traffic Director to trust certificates signed by specific CAs.

Configuring Certificate Trust Flags Using the Administration Console

To specify whether Oracle Traffic Director should trust certificates signed by a specific CA by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."

2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration for which you want to change certificate trust flags.

4. In the navigation pane, expand **SSL** and select **Certificate Authorities**.

   The resulting page displays the installed certificates.

   Note: If the pin is enabled for a token in the selected configuration, the installed certificates are not displayed. Instead, a message to enter the token pins is displayed on the page.

   1. Click **Cache Token Pin**.
   2. In the resulting dialog box, enter the pins for the tokens, and click **OK**.

5. Click the nickname of the certificate for which you want to change the trust flags.

   The Edit Certificate Authority dialog box is displayed.

6. Select the **Trusted to Sign Client Certificates** or **Trusted to Sign Server Certificates** check box, as required.

7. Click **Save**.

   A message is displayed in the Console Messages pane, confirming that the trust flags for the selected certificate have been updated.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Certificate Trust Flags Using the CLI

To specify whether Oracle Traffic Director should trust certificates signed by a specific CA, run the `set-cert-trust-prop` command.

For example, the following command configures the certificate with the nickname `Visa eCommerce Root` in the configuration `soa` to be trusted to sign client and server certificates.

```
tadm> set-cert-trust-prop --config=soa --nickname="Visa eCommerce Root"
 is-client-ca=true is-server-ca=true
OTD-70201 Command 'set-cert-trust-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `set-cert-trust-prop`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

11.5 Managing PKCS#11 Tokens

A PKCS#11 token is a software or hardware interface to a Public-Key Cryptography Standards (PKCS) 11-compliant security database in which digital certificates and keys can be stored. Oracle Traffic Director includes a token named `internal` that provides the interface to the built-in Network Security Services (NSS) certificate database. If any other PKCS#11-compliant databases are available on the administration server host,

Oracle Traffic Director recognizes them automatically and exposes the corresponding tokens, including those implemented through physical devices like hardware accelerators and smart cards.

> **Note:** The information in this section is aimed at readers who are familiar with the concepts of SSL, certificates, ciphers, and keys. For basic information about those concepts, see Section 11.2.7, "SSL/TLS Concepts."

You can enable and disable initialization of PKCS#11 tokens for a Oracle Traffic Director configuration and enable the pins for the tokens.

- If initialization of PKCS#11 tokens is enabled for a configuration *and* if the pin is enabled for any of the tokens, when you attempt to start instances of the configuration, a prompt to enter the pins for pin-enabled tokens is displayed.

  To avoid having to enter the token pin every time you start instances, while specifying the pin, you can opt to save it in the configuration file, as described later in this section.

- If the pin is enabled for a token in a configuration, when you access the certificates database represented by that token for any purpose (for example, to view installed certificates or to install a certificate), a prompt to select the token and enter the token pin is displayed. To avoid having to enter the token pin repeatedly, you can cache it as described in "Caching the Token Pins for an Administration Console Session".

You can configure PKCS#11 tokens by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Configuring PKCS#11 Settings Using the Administration Console

To configure a PKCS#11 token by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to configure tokens.
4. In the navigation pane, select **SSL**.

   The SSL Settings page is displayed. The Cryptographic Tokens section contains the parameters pertaining to PKCS#11 tokens.

   - To enable initialization of PKCS#11 tokens, select the **PKCS#11 Tokens** check box.
   - If you want Oracle Traffic Director to bypass processing of the PKCS#11 layer during SSL/TLS processing, select the **Allow PKCS#11 Bypass** check box. Bypassing the PKCS#11 layer improves performance.

- To enable or disable a token, and to set or change a token's pin, click on the token name.

  The Edit Token dialog box is displayed.

  – To enable the token, select the **Token State** check box.

  – To enable the token pin, select the **Set Token Pin** check box.

    Enter the new pin and confirm it.

  – To change the token pin, select the **Edit Token Pin** check box.

    Enter the current pin, and then enter the new pin and confirm it.

  – To disable the token pin, select the **Edit Token Pin** check box.

    Enter the current pin, and then select the **Unset Pin** check box.

Note:

- If you select **Save Pin**, the token pin is saved in the configuration file. Users are not prompted to enter the token pin when they attempt to start instances of the configuration.
- If you set or change the token pin, and choose not to save it in the configuration file, then to restart the server, you should stop it and then start it again. You cannot restart the server by using the `restart-admin` CLI command or through the administration console.

When you change the value in a field or tab out of a text field that you changed, the **Save** button is enabled.

5. After making the required changes, click **Save**.

   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring PKCS#11 Settings Using the CLI

- To enable or disable initialization of PKCS#11 tokens, run the `set-pkcs11-prop` command, as shown in the following example:

```
tadm> set-pkcs11-prop --config=soa enabled=true
OTD-70201 Command 'set-pkcs11-prop' ran successfully.
```

- To view the available PKCS#11 tokens in a configuration, run the `list-tokens` commands as shown in the following example:

```
tadm> list-tokens --config=soa --verbose --all
name            enabled         has-saved-pin
-----------------------------------------------
internal        false           false
```

- To enable or disable a token, run the `set-token-prop` command, as shown in the following example:

```
tadm> set-token-prop --config=soa --token=internal enabled=true
OTD-70201 Command 'set-token-prop' ran successfully.
```

- To set or change the pin for a token, run the `set-token-pin` command, as shown in the following example:

  ```
  tadm> set-token-pin --config=soa --token=internal
  ```

  If the token is already protected with a pin, a prompt to enter the current pin is displayed. Enter the current pin, and when prompted, enter the new pin and confirm it.

  Note: If you enable initialization of PKCS#11 tokens (`set-pkcs11-prop` ... `enabled=true`) and if the pin is enabled for any of the tokens, then when you attempt to start or restart the instances of the configuration, a prompt to enter the pins for the pin-enabled tokens is displayed. To suppress the pin prompt, you can save the pins in the configuration file by specifying the `--save-pin=true` option for the `set-token-pin` command.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

Caching the Token Pins for an Administration Console Session

If the pin is enabled for a token in a configuration, when you access the certificates database represented by that token for any purpose (for example, to view installed certificates or to install a certificate), a prompt to select the token and enter the token pin is displayed. When using the administration console for managing certificates, you can avoid having to enter the token pins repeatedly, by specifying them once and caching them for the duration of the session; that is until you log out or until the session times out.

To cache the token pins for an administration-console session, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to cache token pins.
4. In the navigation pane, expand **SSL** and select **Server Certificates** or **Certificate Authorities**.
5. Click **Cache Token Pin**.

   The Cache Token Pin dialog box is displayed.
6. Enter the pins for the tokens.
7. Click **OK**.

   A message is displayed in the Console Messages pane confirming that the token pins are cached for the session.

11.6 Managing Certificate Revocation Lists

A Certificate Revocation List (CRL) is a list that a CA publishes to inform users about certificates that the CA has decided to revoke before they expire. CRLs are updated periodically; the updated CRLs can be downloaded from the CA's website.

To ensure that Oracle Traffic Director servers do not trust server certificates that have been revoked by CA, you should download the latest CRLs from the CAs' websites regularly and install them in your Oracle Traffic Director configurations.

You can install CRLs manually. You can also configure Oracle Traffic Director to take the downloaded CRLs from a specified directory and install them automatically at specified intervals.

This section contains the following topics:

- Section 11.6.1, "Installing and Deleting CRLs Manually"
- Section 11.6.2, "Installing CRLs Automatically"

Note:

- The information in this section is aimed at readers who are familiar with the concepts of SSL, certificates, ciphers, and keys. For basic information about those concepts, see Section 11.2.7, "SSL/TLS Concepts."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

11.6.1 Installing and Deleting CRLs Manually

You can install and delete CRLs manually by using either the administration console or the CLI.

Installing CRLs Manually Using the Administration Console

To install a downloaded CRL by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to install a certificate.
4. In the navigation pane, expand **SSL**, and select **Certificate Authorities**.
5. Click the **Install CRL** button.

   The Install Certificate Revocation List dialog box is displayed.
6. Specify the location of the downloaded CRL file, and click **Install CRL**.
   - A message, confirming successful installation of the CRL, is displayed in the Console Messages pane.
   - The CRL that you just installed is displayed on the Certificate Authorities page.

- In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Installing and Deleting CRLs Manually Using the CLI

- To install a downloaded CRL, run the `install-crl` command, as shown in the following example:

```
tadm> install-crl --config=soa /home/admin/crls/ServerSign.crl
OTD-70201 Command 'install-crl' ran successfully.
```

- To view a list of the installed CRLs in a configuration, run the `list-crls` command, as shown in the following example:

```
tadm> list-crls --config=soa --verbose --all
crl-name        next-update
--------------------------
"Class 1 Public Primary Certification Authority" "Sat Apr 15 16:59:59 PDT 2000"
"VeriSign Class 3 Code Signing 2010 CA" "Mon Aug 29 14:00:03 PDT 2011"
"VeriSign Class 3 Organizational CA" "Sun May 18 13:48:16 PDT 2014"
```

- To delete a CRL, run the `delete-crl` command, as shown in the following example:

```
tadm> delete-crl --config=config1 "VeriSign Class 3 Organizational CA"
OTD-70201 Command 'delete-crl' ran successfully.
```

  When you delete a CRL, it is removed from the Oracle Traffic Director configuration *and* from the directory in which the downloaded CRL was stored.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.6.2 Installing CRLs Automatically

You can configure Oracle Traffic Director to periodically take downloaded CRL files from a specified directory and install them automatically by using either the administration console or the CLI.

At the specified interval, Oracle Traffic Director looks for updated CRL files in the specified directory.

- If Oracle Traffic Director detects new CRL files, it installs them in the configuration and logs a message in the server log.
- If existing CRL files have been changed, Oracle Traffic Director installs the updated CRL files in the configuration and logs a message in the server log.
- If Oracle Traffic Director detects that previously installed CRL files have been removed from the directory, it deletes the CRLs from the configuration and logs a message in the server log.
- If no changes are detected in the CRL directory, Oracle Traffic Director does not perform any update.

Configuring Oracle Traffic Director to Install CRLs Automatically Using the Administration Console

To configure Oracle Traffic Director to install CRLs automatically by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Click the name of the configuration that you want to set up to install CRLs automatically.
4. In the navigation pane, select **SSL**.

   The SSL Settings page is displayed.
5. Go to the **Advanced Settings** section of the page.
6. In the **Update CRL Path** field, enter the absolute path to the directory that contains the updated CRL files.
7. Click **New Event**.

   The New CRL Update Event dialog box is displayed.
8. Specify the interval or time of the day at which the CRLs should be updated, and then click **OK**.
   - A message, confirming creation of the event, is displayed in the Console Messages pane.
   - The new event is displayed in the CRL Update Events list.
     - New events are enabled by default. To change the status, select the **Enable/Disable** check box.
     - To delete an event, click the **Delete** button.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Oracle Traffic Director to Install CRLs Automatically Using the CLI

To configure Oracle Traffic Director to install CRLs automatically, do the following:

1. Specify the directory in which the downloaded CRLs are stored, by using the `set-pkcs11-prop` command.

   For example, the following command specifies `/home/admin/crls` as the directory in which downloaded CRLs are stored.

   ```
   tadm> set-pkcs11-prop --config=soa crl-path=/home/admin/crls
   OTD-70201 Command 'set-pkcs11-prop' ran successfully.
   ```

2. Schedule an event for Oracle Traffic Director to take the downloaded CRLs from the specified directory and install them automatically, by using the `create-event` command.

   For example, the following command specifies that the CRLs for the configuration `soa` should be updated after every 86400 seconds (1 day).

```
tadm> create-event --config=soa --command=update-crl --interval=604800
OTD-70201 Command 'create-event' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.7 Managing Web Application Firewalls

A web application firewall (WAF) is a filter or server plugin that applies a set of rules, called rule sets, to an HTTP request. Web application firewalls are useful for establishing an increased security layer in order to identify and prevent attacks. It acts as a firewall for applications hosted within the origin server. In addition, it enables administrators to inspect any part of an HTTP request, such as headers and body, and configure conditions to accept or reject the HTTP request based on the condition.

Several free and commercial versions of web application firewall modules are available for use. The web application firewall module for Oracle Traffic Director supports ModSecurity 2.6, which is an intrusion detection and prevention engine for web applications. The ModSecurity rule sets can be customized to shield applications from common attacks such as cross-site scripting (XSS) and SQL injection. Based on various criterion, such as HTTP headers, environment variables and CGI variables, ModSecurity filters and rejects incoming requests. For more information about ModSecurity, see https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Introduction.

Among the many providers who have published different versions of the rule sets for ModSecurity, Oracle Traffic Director has been tested with the Open Web Application Security Project (OWASP), which is an open-source application security project, and is one of the most commonly used rule set providers. For more information, see https://www.owasp.org/index.php/Category:OWASP_ModSecurity_Core_Rule_Set_Project.

This section contains the following topics:

- Section 11.7.1, "Overview of Web Application Firewalls"
- Section 11.7.2, "Configuring Web Application Firewalls"
- Section 11.7.3, "Listing the Rule Set Files"
- Section 11.7.4, "Removing Rule Set Files"
- Section 11.7.5, "Supported Web Application Firewall Directives, Variables, Operators, Actions, Functions, Persistent Storages and Phases"

11.7.1 Overview of Web Application Firewalls

With Oracle Traffic Director, web application firewalls can be enabled (or disabled) for each virtual server in your configuration. This in turn applies a set of rules, and acts as a firewall for the web applications deployed on the origin servers. For more information about origin servers and virtual servers, see Chapter 7, "Managing Origin Servers" and Chapter 8, "Managing Virtual Servers" respectively.

Oracle Traffic Director supports rule sets at both virtual server level and configuration level. Note that rules defined at the virtual server level will override rules defined at

the configuration level. When deployed, these rules and the configuration changes are pushed to the instances, reconfiguring the instances. For more information about the web application firewall works, see Appendix B, "Web Application Firewall Examples and Use Cases."

11.7.2 Configuring Web Application Firewalls

To configure web application firewalls, you can either download an open source web application firewall rule sets or create your own rule sets. For example, download the ModSecurity Core Rule Set (CRS) from the OWASP repository, and unzip the rule sets to any folder. Oracle Traffic Director supports rules in the following directories:

- `base_rules`
- `optional_rules`
- `slr_rules`

> **Note:** Web application firewall supports the ModSecurity 2.6 directives that are used by the configurations within the `base_rules`, `optional_rules` and `slr_rules` directories of OWASP ModSecurity Core Rule Set. However, it does not support Apache core config directives such as `<IfDefine...>` and `<Location...>`, and the ones supported by other Apache modules such as `RequestHeader`, `Header` and so on.

After unzipping the above directories, the files in these directories can be edited and uploaded to the administration server. These rule set files are then pushed to the Oracle Traffic Director instances when deployed. For more information, see Section 11.7.2.1, "Enabling and Installing Web Application Firewall Rule Sets."

Note:

- Though the server can be configured to pick up the rule set files from a directory outside the `config` directory, rule file management will not be supported. When Oracle Traffic Director is configured for high availability, it is recommended that the web application firewall rule sets are placed within the `config` directory.
- Using unsupported directives, variables, operators, actions, phases, functions and storages can cause server startup errors. For example, installing the rule set file `modsecurity_crs_42_tight_security.conf` without removing the unsupported action `ver` can cause Oracle Traffic Director to display the following error message when you start the server:

```
[ERROR:16] [OTD-20016] Syntax error on line 20 of
/scratch/rgoutham/instance1/net-config1/config/ruleset/config1/
modsecurity_crs_42_tight_security.conf:
[ERROR:16] [OTD-20016] Error parsing actions: Unknown action:
ver
[ERROR:32] [OTD-20008] Failed to parse VS
webapp-firewall-ruleset (ruleset/config1/*.conf)
[ERROR:32] [OTD-10422] Failed to set configuration
[ERROR:32] server initialization failed
```

  To avoid getting this error, modify the rule set file, and remove or comment out unsupported directives, variables, operators, actions, phases, functions and storages, and then start the server.

11.7.2.1 Enabling and Installing Web Application Firewall Rule Sets

You can enable and install web application firewall rule sets by using either the administration console or the CLI.

Note:

- When you enable and install a web application firewall rule set, you are, in effect, modifying a configuration. So for the new rule set to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Enabling and Installing Web Application Firewall Rule Sets Using the Administration Console

To configure web application firewall for a virtual server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

A list of the available configurations is displayed.

3. Select the configuration for which you want to configure web application firewall.
4. In the navigation pane, expand **Virtual Servers**, expand the name of the virtual server for which you want to configure web application firewall, and select **Web Application Firewall**.

   The Web Application Firewall page is displayed.

   a. On the Web Application Firewall page, click **Enabled** to enable web application firewall for a particular virtual server.

   b. Click **Install Rule Set Files**.

      In the Install Rule Set Files dialog box, either browse to the folder where you unzipped the rule set files and select the rule set file or enter the full path to the location of the rule set file. To install multiple rule set files, install them one at a time.

      After you install one or more rule set files, the following text is added to the Rule Set Pattern field:

      ```
      ruleset/<virtual-server-id>/*.conf
      ```

      Note:

      - When you install rule set files at the configuration level, the rule set pattern appears as follows:

        ```
        ruleset/*.conf
        ```

      - If required, you can add custom rule set patterns. However, rule sets outside the `ruleset/<virtual-server-id>` directory (if at the virtual server level) or the `ruleset` directory (if at the configuration level) cannot be viewed or deleted using the Oracle Traffic Director administration console or CLI. These rule sets will need to be managed manually.

      ---

   c. Click **Install Rule Set**.

      A message, confirming that the rule set files were installed, is displayed in the Console Messages pane.

      In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Enabling Web Application Firewall Using the CLI

To enable web application firewall using the CLI, run the `enable-webapp-firewall` command.

For example, the following command enables web application firewall for the virtual server `vs1` in the configuration `soa`.

```
tadm> enable-webapp-firewall --config=soa --vs=vs1
OTD-70201 Command 'enable-webapp-firewall' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `enable-webapp-firewall`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

Installing Web Application Firewall Rule Sets Using the CLI

To install web application firewall rule sets using the CLI, run the `install-webapp-firewall-ruleset` command.

For example, the following command installs the web application firewall rule set `modsecurity_crs_20_protocol_violations.conf` for the virtual server `vs1` in the configuration `soa`.

```
tadm> install-webapp-firewall-ruleset --config=soa --vs=vs1
/home/rulesets/modsecurity_crs_20_protocol_violations.conf
OTD-70201 Command 'install-webapp-firewall-ruleset' ran successfully.
```

To install web application firewall rule sets at the configuration level, run the above command without the `--vs` option. For example, the following command installs the web application firewall rule set `modsecurity_crs_50_outbound.conf` for the configuration `soa`.

```
tadm> install-webapp-firewall-ruleset --config=soa /home/rulesets/modsecurity_crs_
50_outbound.conf
OTD-70201 Command 'install-webapp-firewall-ruleset' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `install-webapp-firewall-ruleset`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

> **Note:** You can use the `set-config-prop` and `set-virtual-server-prop` commands to set the value of `webapp-firewall-ruleset` property at the configuration level and virtual server level respectively. For more information, see the *Oracle Traffic Director Command-Line Reference*.

11.7.3 Listing the Rule Set Files

You can view the list of rule set files by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing the List of Rule Set Files Using the Administration Console

To view the list of rule set files by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the Configurations button that is situated at the upper left corner of the page.

A list of the available configurations is displayed.

3. Select the configuration for which you want to view rule set files.
4. On the Web Application Firewall page, the Rule Set Files table lists the installed rule set files. To view the contents of these files either click and select individual rule files, or click the **Name** check box to select all the rules files.
5. Click **View**.

   The contents of each rule file is displayed in the Rule set file contents window.

Viewing the List of Rule Set Files Using the CLI

While it is not possible to view the contents of individual rule set files using CLI, you can view the list of installed rule set files. To view the list of rule set files, run the `list-webapp-firewall-rulesets` command.

For example, the following command lists the installed web application firewall rule set files for the virtual server `vs1` in the configuration `soa`:

```
tadm> list-webapp-firewall-rulesets --config=soa --vs=vs1 --verbose

ruleset-file-name
--------------------
modsecurity_crs_45_trojans.conf
modsecurity_crs_42_tight_security.conf
modsecurity_crs_46_slr_et_sqli_attacks.conf
```

To view the list of web application firewall rule sets that are installed at the configuration level, run the above command without the `--vs` option. For example, the following command lists the web application firewall rule sets that are installed at the configuration level for the configuration `soa`.

```
tadm> list-webapp-firewall-rulesets --config=soa --verbose

ruleset-file-name
--------------------
modsecurity_crs_40_generic_attacks.conf
modsecurity_crs_47_common_exceptions.conf
```

You can view the properties of a web application firewall by running the `get-webapp-firewall-prop` command.

For more information about the `list-webapp-firewall-rulesets` and `get-webapp-firewall-prop` commands, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

11.7.4 Removing Rule Set Files

You can remove rule set files by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Removing Rule Set Files Using the Administration Console

To remove rule set files for a particular virtual server:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the Configurations button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to delete rule set files.
4. On the Web Application Firewall page, either click and select individual rule files or click the **Name** check box to select all the rule files.
5. Click the **Delete** button. At the confirmation prompt, click **OK**.

   A message is displayed in the Console Message pane confirming that the rule set files were deleted.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes, as described in Section 4.3, "Deploying a Configuration."

Removing Rule Set Files Using the CLI

To remove rule set files for a particular virtual server, run the `delete-webapp-firewall-ruleset` command, as shown in the following example:

```
tadm> delete-webapp-firewall-ruleset --config=soa --vs=vs1 modsecurity_crs_20_
protocol_violations.conf
OTD-70201 Command 'delete-webapp-firewall-ruleset' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `delete-webapp-firewall-ruleset`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

> **Note:** The `disable-webapp-firewall` command can be used to disable a web application firewall. For more information, see the *Oracle Traffic Director Command-Line Reference*.

11.7.5 Supported Web Application Firewall Directives, Variables, Operators, Actions, Functions, Persistent Storages and Phases

Oracle Traffic Director supports various ModSecurity 2.6 directives, variables, operators, actions, functions, persistent Storages and phases.

Supported Web Application Firewall Directives

Oracle Traffic Director supports the following ModSecurity 2.6 directives. For more information and to see the full list of ModSecurity directives, including unsupported directives, see https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Configuration_Directives.

```
SecAction
SecArgumentSeparator
SecAuditEngine
SecAuditLog
SecAuditLog2
```

```
SecAuditLogDirMode
SecAuditLogFileMode
SecAuditLogParts
SecAuditLogRelevantStatus
SecAuditLogStorageDir
SecAuditLogType
SecComponentSignature
SecContentInjection
SecCookieFormat
SecDataDir (see note below)
SecDebugLog
SecDefaultAction
SecDebugLogLevel
SecGeoLookupDb
SecInterceptOnError
SecMarker
SecPcreMatchLimit (see note below)
SecPcreMatchLimitRecursion (see note below)
SecRequestBodyAccess
SecRequestBodyInMemoryLimit (see note below)
SecRequestBodyNoFilesLimit (see note below)
SecRequestBodyLimitAction
SecResponseBodyAccess
SecResponseBodyLimit
SecResponseBodyLimitAction (see note below)
SecResponseBodyMimeType
SecResponseBodyMimeTypesClear
SecRule
SecRuleEngine (see note below)
SecRuleRemoveById
SecRuleRemoveByMsg
SecRuleRemoveByTag
SecRuleUpdateActionById
SecRuleUpdateTargetById
SecTmpDir
SecUnicodeMapFile (see note below)
SecUnicodeCodePage (see note below)
SecUploadDir
SecUploadFileLimit
SecUploadFileMode
SecUploadKeepFiles
SecWebAppId (see note below)
SecCollectionTimeout
```

Note:

- `SecWebAppId` can be specified within virtual server specific web application firewall configuration file to associate the application namespace to a particular virtual server.
- The directive `SecRequestBodyLimitAction` enables you to set action against requests that hit `SecRequestBodyNoFilesLimit`. However, the directive `SecRequestBodyLimit` is not supported by Oracle Traffic Director and hence, you cannot set action against this directive.
- Oracle Traffic Director does not support the directive `SecRequestBodyLimit`, which is used for configuring the maximum request body size that ModSecurity accepts for buffering. In place of this directive, the following options can be used:

  Option 1: Use the directives, `SecRequestBodyNoFilesLimit` and `SecRequestBodyLimitAction`. Example:

  ```
  SecRequestBodyNoFilesLimit 100
  SecRequestBodyLimitAction Reject
  ```

  Option 2: For Reject behavior, Oracle Traffic Director can be configured to check a request's `Content-Length` header in obj.conf. In addition, `max-unchunk-size` value can be set in server.xml.

  Similarly, for ProcessPartial behavior, `body-buffer-size` element in server.xml can be set to the desired limit. In this case, only the first part of the body that fits the limit is processed and the rest is passed through.
- If the directive `SecRuleEngine` is specified within the configuration file(s) specified by the `webapp-firewall-ruleset` element, then it will be ignored. However, this condition is not applicable if `SecRuleEngine` is set to `DetectionOnly` mode.
- The directive `SecRequestBodyInMemoryLimit` is ignored if the header `Content-Type` is set to `x-www-form-urlencoded`.
- The directives `SecDataDir`, `SecPcreMatchLimit`, `SecPcreMatchLimitRecursion`, `SecUnicodeCodePage`, and `SecUnicodeMapFile` can only be used at configuration level. The scope of these directives is considered to be Main. All the other directives can be used at both virtual server level and configuration level. The scope of these directives is considered to be Any. If a directive with Main scope is specified within the virtual server level configuration file, then an error will be logged and the server will fail to start.

Supported Web Application Firewall Variables

Oracle Traffic Director supports the following ModSecurity 2.6 variables. For more information and to see the full list of ModSecurity variables, including the unsupported variables, see

https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Variables.

```
ARGS
ARGS_COMBINED_SIZE
ARGS_GET
ARGS_GET_NAMES
ARGS_NAMES
ARGS_POST
ARGS_POST_NAMES
AUTH_TYPE
DURATION
ENV
FILES
FILES_COMBINED_SIZE
FILES_NAMES
FILES_SIZES
GEO
HIGHEST_SEVERITY
MATCHED_VAR
MATCHED_VARS
MATCHED_VAR_NAME
MATCHED_VARS_NAMES
MODSEC_BUILD
MULTIPART_BOUNDARY_QUOTED
MULTIPART_BOUNDARY_WHITESPACE
MULTIPART_DATA_AFTER
MULTIPART_DATA_BEFORE
MULTIPART_FILE_LIMIT_EXCEEDED
MULTIPART_HEADER_FOLDING
MULTIPART_INVALID_QUOTING
MULTIPART_INVALID_HEADER_FOLDING
MULTIPART_LF_LINE
MULTIPART_MISSING_SEMICOLON
MULTIPART_CRLF_LF_LINES
MULTIPART_STRICT_ERROR
MULTIPART_UNMATCHED_BOUNDARY
PERF_COMBINED
PERF_GC
PERF_LOGGING
PERF_PHASE1
PERF_PHASE2
PERF_PHASE3
PERF_PHASE4
PERF_PHASE5
PERF_SREAD
PERF_SWRITE
QUERY_STRING
REMOTE_ADDR
REMOTE_PORT
REMOTE_USER
REQBODY_ERROR
REQBODY_ERROR_MSG
REQBODY_PROCESSOR
REQBODY_PROCESSOR_ERROR
REQUEST_BASENAME
REQUEST_BODY (see note below)
REQUEST_BODY_LENGTH
REQUEST_COOKIES
REQUEST_COOKIES_NAMES
REQUEST_FILENAME
```

```
REQUEST_HEADERS (see note below)
REQUEST_HEADERS_NAMES
REQUEST_LINE
REQUEST_METHOD
REQUEST_PROTOCOL
REQUEST_URI
REQUEST_URI_RAW
RESPONSE_BODY
RESPONSE_CONTENT_LENGTH
RESPONSE_CONTENT_TYPE
RESPONSE_HEADERS
RESPONSE_HEADERS_NAMES
RESPONSE_PROTOCOL
RESPONSE_STATUS
RULE
SERVER_ADDR
SERVER_NAME
SERVER_PORT
SESSIONID
TIME
TIME_DAY
TIME_EPOCH
TIME_HOUR
TIME_MIN
TIME_MON
TIME_SEC
TIME_WDAY
TIME_YEAR
TX
UNIQUE_ID
URL_ENCODED_ERROR
USERID
WEBAPPID
WEBSERVER_ERROR_LOG (see note below)
XML
```

Note:

- The `REQUEST_BODY` variable, which holds raw request body, will contain the body content that is available after it passes through other filters.
- In open source ModSecurity, apache error log for each request/response can be collected and stored in the `WEBSERVER_ERROR_LOG` variable, and printed in the `auditlog` action. However, Oracle Traffic Director does not support this feature.
- As request headers with the same name are concatenated into a single one, the header count is always 1. Hence, `&REQUEST_HEADERS:<any header name>` will always return 1 in spite of how many same request headers were sent.

Supported Web Application Firewall Operators

Oracle Traffic Director supports the following ModSecurity 2.6 operators. For more information and to see the full list of ModSecurity operators, including unsupported operators https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Operators.

```
beginsWith
contains
containsWord
endsWith
eq
ge
geoLookup
gt
inspectFile
ipMatch
le
lt
pm
pmf
pmFromFile
rbl (see note below)
rx
streq
strmatch
validateByteRange
validateDTD
validateSchema
validateUrlEncoding
validateUtf8Encoding
verifyCC
verifyCPF
verifySSN
within
```

Note: ModSecurity 2.6 does not support the directive `SecHttpBlKey`. Hence use of Project Honey Pot (dnsbl.httpbl.cong) as RBL, which requires `SecHttpBlKey`, is not supported.

Supported Web Application Firewall Actions

Oracle Traffic Director supports the following ModSecurity 2.6 actions. For more information and to see the full list of ModSecurity actions, including the unsupported actions, see https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Actions.

```
allow
append
auditlog (see note below)
block
capture
chain
ctl
deny (see note below)
deprecatevar
drop (see note below)
exec
expirevar
id
initcol
log
logdata
msg
multiMatch
```

```
noauditlog
nolog
pass
pause
phase
prepend
redirect
rev
sanitiseArg
sanitiseMatched
sanitiseMatchedBytes
sanitiseRequestHeader
sanitiseResponseHeader
severity
setuid
setsid
setenv
setvar
skip
skipAfter
status
t
tag
xmlns
```

Note:

- In open source ModSecurity, apache error log for each request/response can be collected and stored in the `WEBSERVER_ERROR_LOG` variable, and printed in the `auditlog` action. However, Oracle Traffic Director does not support this feature.
- Actions that change HTTP response status, such as deny, will not successfully change the response status when it is invoked in phase 4. In such a scenario, the following error message is logged in the server log:

  ```
  " ModSecurity: Access denied with code 403 (phase 4)."
  ```

- When `drop` action is invoked in phase 4, Oracle Traffic Director will send out HTTP headers to the client and then drop the connection.
- When `deny` action is invoked in phase 4, Oracle Traffic Director strips the response body, instead of sending 403 response status. This might cause the following warning to appear in the server log:

  ```
  Response content length mismatch (0 bytes with a content length
  of <original content length>)
  ```

Supported Web Application Firewall Transformation Functions

Oracle Traffic Director supports the following ModSecurity 2.6 transformation functions. For more information and to see the full list of ModSecurity transformation functions, see https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Transformation_functions.

```
base64Decode
```

```
sqlHexDecode
base64DecodeExt
base64Encode
cmdLine
compressWhitespace
cssDecode
escapeSeqDecode
hexDecode
hexEncode
htmlEntityDecode
jsDecode
length
lowercase
md5
none
normalisePath
normalisePathWin
parityEven7bit
parityOdd7bit
parityZero7bit
removeNulls
removeWhitespace
replaceComments
removeCommentsChar
removeComments
replaceNulls
urlDecode
urlDecodeUni
urlEncode
sha1
trimLeft
trimRight
trim
```

Supported Web Application Firewall Persistent Storages

Oracle Traffic Director supports the following ModSecurity 2.6 persistent storages. For more information and to see the full list of ModSecurity persistent storages, see https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Persistant_Storage.

```
GLOBAL
IP
RESOURCE
SESSION
USER
```

Supported Web Application Firewall Phases

Oracle Traffic Director supports the following ModSecurity 2.6 phases. For more information and to see the full list of ModSecurity phases, see https://github.com/SpiderLabs/ModSecurity/wiki/Reference-Manual#wiki-Processing_Phases.

```
Phase:1 - Request headers stage
Phase:2 - Request body stage
Phase:3 - Response headers stage
Phase:4 - Response body stage
Phase:5 - Logging
```

Note:

- Actions that change HTTP response status, such as deny, will not successfully change the response status when it is invoked in phase 4.
- When `drop` action is invoked in phase 4, Oracle Traffic Director will send out HTTP headers to the client and then drop the connection.
- When `deny` action is invoked in phase 4, Oracle Traffic Director strips the response body, instead of sending 403 response status. This might cause the following warning to appear in the server log:

  ```
  Response content length mismatch (0 bytes with a content length
  of <original content length>)
  ```

11.8 Configuring Client Authentication

Client authentication is the verification of a client by the Oracle Traffic Director virtual server or TCP proxy, based on the certificate that the client provides.

Client authentication is not enabled by default. You can configure the Oracle Traffic Director listeners to require clients to provide a certificate, by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Configuring Client Authentication Using the Administration Console

To enable client authentication for a listener by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to enable client authentication for listeners.
4. In the navigation pane, expand **Listeners**, and select the listener for which you want to configure client authentication.

   The Listener Settings page is displayed.
5. Go to the **Advanced Settings** section of the page and scroll down to the **SSL/TLS** subsection.
6. Select the required **Client Authentication** mode.
   - **Required**: The server *requests* the client for a certificate; if the client does not provide a certificate, the connection is closed.

- **Optional**: The server *requests* the client for a certificate, but does not *require* it. The connection is established even if the client does not provide a certificate.
- **False** (default): Client authentication is disabled.

7. Specify the **Authentication Timeout** and **Maximum Authentication Data** parameters.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

8. After making the required changes, click **Save**.
   - A message, confirming that the updated listener was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Client Authentication Using the CLI

To enable client authentication for an HTTP or TCP listener, run the `set-ssl-prop` command.

For example, the following command makes client authentication mandatory for the listener `http-listener-1`, with 60 seconds as the authentication time-out duration and 262144 bytes as the maximum length of authentication data that can be buffered.

```
tadm> set-ssl-prop --config=soa --http-listener=http-listener-1
 client-auth=required max-client-auth-data=262144 client-auth-timeout=60
OTD-70201 Command 'set-ssl-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

11.9 Preventing Denial-of-Service Attacks

A denial-of-device (DoS) attack is an attempt by a malicious user to prevent legitimate users from accessing a service, by sending continuous requests to the server.

To prevent DoS attacks, you can configure Oracle Traffic Director virtual servers to reject requests when the frequency of requests or the number of concurrent connections exceeds a specified limit. For more granular control over requests, you can define several request limits and configure each limit to be applied to requests that match specified URL patterns and query string patterns, request headers that match specified values, and so on.

This section contains the following subsections:

- Section 11.9.1, "Request Limiting Parameters"
- Section 11.9.2, "Configuring Request Limits for a Virtual Server"

11.9.1 Request Limiting Parameters

You can specify multiple request limits for a virtual server. For each request limit, you can configure several parameters:

- You can make each request limit applicable to requests fulfilling a specified condition that you specify using expressions such as the following:

  ```
  $path = "*.jsp"
  $url = "/images/*"
  $ip =~ "^130\.35\.46\..*"
  ```

  You can use any variable or a combinations of variables to specify the condition for a limit. For more information about building expressions for request limit conditions, see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

- In each request limit, you can specify the number of concurrent requests (`max-connections`) and the average number of requests per second (`max-rps`).

  For example, if you specify a limit (say, `max-rps`=20), Oracle Traffic Director continuously tracks the request rate by recalculating it at a compute interval that you specify (default: 30 seconds), based on the number of requests received during that interval. When the specified request limit is reached, Oracle Traffic Director rejects all subsequent requests.

- You can also specify an optional attribute that Oracle Traffic Director should monitor when applying request limits. Oracle Traffic Director uses separate counters to track the request statistics for each monitored attribute.

  For example, to specify that Oracle Traffic Director should track the request rate separately for each client IP, you can specify the variable `$ip` as the monitor attribute. When the request rate exceeds the specified limit for any client, subsequent requests from *that* client are rejected, but requests from other clients continue to be served.

  You can also combine variables when specifying the attribute to be monitored. For example, to limit requests from clients that request the same URIs too frequently, you can specify `$ip:$uri` as the attribute to be monitored. When the request rate from any client for any single URI exceeds the limit, further requests to the same URI from that client are rejected, but requests from that client to other URIs, as well as requests from other clients to any URI continue to be served.

- For requests that Oracle Traffic Director rejects, it returns the HTTP response code that you specify. The default status code is `503 (service unavailable)`.

- After a specified limit—`max-connections` or `max-rps`—is reached, Oracle Traffic Director continues to reject all subsequent requests until a specified *continue condition* is satisfied. You can specify one of the following continue conditions:

  – **Threshold** (default): Service resumes when the request rate falls below the specified limit.

  – **Silence**: Service resumes when the incoming request falls to zero for an entire interval.

11.9.2 Configuring Request Limits for a Virtual Server

You can configure request limits for a virtual server by using either the administration console or the CLI.

Note:

- When you modify a virtual server, you are, in effect, modifying a configuration. So for the new virtual-server settings to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Configuring Request Limits Using the Administration Console

To configure request limits by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to configure request limits.
4. In the navigation pane, expand **Virtual Servers**, expand the name of the virtual server for which you want to configure request limits, and select **Request Limits**.

   The Request Limits page is displayed. It lists the request limits that are currently defined for the virtual server.

   Creating a Request Limit

   a. Click **New Request Limit**.

      The New Request Limit dialog box is displayed.

      In the **Name** field, enter a name for the new request limit.

      In the **Connections** field, specify the maximum number of concurrent connections to the virtual server.

      In the **Requests Per Second** field, specify the maximum number of requests that the virtual server can accept per second.

      Note: You must specify at least one of the limits—maximum number of connections or maximum number of requests per second.

      In the **Monitor Attribute** field, specify the attribute in the request header, which the virtual server should monitor for applying the request limit. If you do not specify this parameter, the request limit is applied to all requests.

   b. Click **Next**.

      If this is the first request limit for the virtual server, the New Caching Rule dialog box gives you the option to choose whether the limit should be applied to all requests. Select **All Requests**.

      If you wish to apply the limit to only those requests that satisfy a condition, create a new condition by selecting **Create a new condition**. In the New

Expression pane, select a Variable/Function and an Operator from the respective drop-down lists and provide a value in the **Value** field.

Select the `and`/or `operator` from the drop-down list when configuring multiple expressions. Similarly, use the `Not` operator when you want the route to be applied only when the given expression is not true.

To enter a condition manually, click **Cancel** and then click **Edit Manually**. In the **Condition** field, specify the condition under which the request limit should be applied. For information about building condition expressions, click the help button near the Condition field or see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

c. Click **Next** and then click **Create Request Limit**.

The request limit that you just created is displayed on the Request Limits page.

In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Editing a Request Limit

To change the settings of a request limit, do the following:

a. Click the **Name** of the request limit.

The Editing Request Limit page is displayed.

Note: To access the condition builder to edit conditions, select **Requests satisfying the condition** and click **Edit**. The condition builder enables you to delete old expressions and add new ones.

---

b. Specify the parameters that you want to change.

While editing a request limit, in addition to changing the parameters that you specified while creating the request limit, you can set and change the `requests-per-second` compute interval, and the HTTP status code that the virtual server should return for requests that it rejects when the specified limits are reached. In addition, you can edit the condition that you have set by clicking **Edit**, which allows you to edit the condition either manually or using the condition builder. You can also delete old expressions and add new ones.

On-screen help and prompts are provided for all of the parameters.

When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

At any time, you can discard the changes by clicking the **Reset** button.

c. After making the required changes, click **Save**.

A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.

In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by

clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Deleting a Request Limit

To delete a request limit, click the **Delete** button. At the confirmation prompt, click **OK**.

Configuring Request Limits Using the CLI

- To create a request limit, run the `create-request-limit` command.

  Examples:

  – The following command creates a request limit named `limit_ip` in the virtual server `soa.example.com` of the configuration `soa`, to limit the number of concurrent requests from any single client to 5.

  ```
  tadm> create-request-limit --config=soa --vs=soa.example.com
  --max-connections=5 limit_ip
  OTD-70201 Command 'create-request-limit' ran successfully.
  ```

  – The following command creates a request limit named `limit_subnet` in the virtual server `soa.example.com` of the configuration `soa`, to limit the number of requests per second from the client IP addresses in the subnet `111.23.30.*` to 20.

  ```
  tadm> create-request-limit --config=soa --vs=soa.example.com
  --condition="$ip='111.12.30.*'" --max-rps=20 limit_subnet
  OTD-70201 Command 'create-request-limit' ran successfully.
  ```

  Note that the value of the `--condition` option should be a regular expression. For information about building condition expressions, see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

- To view a list of the request limits defined for a virtual server, run the `list-request-limits` command, as shown in the following example:

  ```
  tadm> list-request-limits --config=soa --vs=soa.example.com
  request-limit   condition
  ------------------------
  limit_ip        -
  limit_subnet    "$ = '111.23.30.*'"
  ```

- To view the properties of a request limit, run the `get-request-limit-prop` command, as shown in the following example:

  ```
  tadm> get-request-limit-prop --config=soa --vs=soa.example.com
  --request-limit=limit_ip
  continue-condition=silence
  condition="$ip = '111.23.30.*'"
  error-code=503
  max-connections=50
  rps-compute-interval=30
  max-rps=20
  request-limit=limit_ip
  ```

- To change the properties of a request limit, run the `set-request-limit-prop` command.

For example, the following command changes the request-per-second compute interval of the request limit `limit_ip` in the virtual server `soa.example.com` of the configuration `soa` to 60 seconds.

```
tadm> set-request-limit-prop --config=soa --vs=soa.example.com --rule=loan-rule
rps-compute-interval=60
```

- To delete a request limit, run the `delete-request-limit` command, as shown in the following example:

```
tadm> delete-request-limit --config=soa --vs=soa.example.com limit_ip
OTD-70201 Command 'delete-request-limit' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

12 Managing Logs

Oracle Traffic Director records data about server events such as configuration changes, instances being started and stopped, errors while processing requests, and so on in log files. You can use the logs to diagnose server problems, evaluate server usage patterns, and tune the system for improved performance.

This chapter contains the following sections:

12.1 About the Oracle Traffic Director Logs

Each Oracle Traffic Director instance, including the administration server, has two logs—an access log and a server log. The instance logs are enabled by default and initialized when the instance is started for the first time. In addition to the instance logs, you can enable access and server logs for each virtual server in the instance.

- The default location of the access log and server log for an Oracle Traffic Director instance is the *INSTANCE_HOME*/net-*config_name*/logs directory.
- For the administration server, the default location of the log files is *INSTANCE_HOME*/admin-server/logs directory.

This section provides an overview of the access and server logs. For information about changing log settings, including the name and location of log files, see Section 12.3, "Configuring Log Preferences."

12.1.1 Access Log

The access log contains information about requests to, and responses from, the server. The default name of the access log file is access.log.

The following example shows the first three lines in a typical access log:

```
format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
"%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%
%Req->vars.ecid%
10.177.243.207 - - [28/Aug/2011:23:28:30 -0700] "GET / HTTP/1.1" 200 4826 -
10.177.243.207 - - [28/Aug/2011:23:28:31 -0700] "GET / HTTP/1.1" 200 916 -
```

The first line indicates the access log format. The second and third lines are the actual entries.

You can change the access log format, file name, and location. You can also disable the access log. For more information, see Section 12.3, "Configuring Log Preferences."

12.1.2 Server Log

The server log contains data about lifecycle events—server start-up, shut down, and restart; configuration updates; and so on. It also contains errors and warnings that the server encountered. The default name of the server log file is `server.log`.

The following line is an example of an entry in a server log.

```
[2011-10-03T02:04:59.000-07:00] [net-soa] [NOTIFICATION] [OTD-10358] []
 [pid: 11722] http-listener-1: http://example.com:1904 ready to accept requests
```

The default server-log level is `NOTIFICATION:1`, at which only major lifecycle events, warnings, and errors are logged.

You can change the log level, the log file name, and the log file location. For more information, see Section 12.3, "Configuring Log Preferences."

Table 12–1 lists the log levels that you can specify for the server log.

Table 12–1 Server Log Levels

Log Level	Description
`INCIDENT_ERROR:1`	A serious problem caused by unknown reasons. You should contact Oracle for support.

`ERROR:1`
`ERROR:16`
`ERROR:32`
	A serious problem that requires your immediate attention.

`WARNING:1`	A potential problem that you should review.
`NOTIFICATION:1` (default)	A major lifecycle event, such as a server being started or restarted.

`TRACE:1`
`TRACE:16`
`TRACE:32`
	Trace or debug information to help you or Oracle Support diagnose problems with a particular subsystem.

The number following each log level indicates the severity of the logged event on the scale 1–32. An `ERROR:1` message is of higher severity than an `ERROR:16` message.

`TRACE:32` is the most verbose log level and `INCIDENT_ERROR:1` is the least verbose. Enabling the `TRACE` log levels might affect performance, because of the high volume of messages that are logged. Therefore, avoid enabling verbose log levels in production systems, except in situations when you need more detailed information to debug issues.

12.2 Viewing Logs

You can view the access and server logs of Oracle Traffic Director instances and virtual servers by using either the administration console or the CLI.

Note:

- Besides using the CLI and administration console, you can also use the standard operating-system commands such as `ls` and `more` to list and view the log files.
- The Log Viewer in the administration console and the `get-access-log` CLI command display only the log entries that currently exist in the access log file on the disk. They do not display items from the access-log buffer (see Section 15.7, "Configuring Access-Log Buffer Settings")).
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Viewing Logs Using the Administration Console

To view log data for a node, an instance, or virtual server within an instance by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view logs.
4. In the Common Tasks pane, click **View Logs**.

   The Oracle Traffic Director Log Viewer window is displayed.
5. Click the **Select Node, Instance, or Virtual Server** button, and select the node, instance, or virtual server for which you want to view log data.
   - To view the server log, select the **Server Log** tab.
   - To view the access log, select the **Access Log** tab.

   To search for specific records, click the **Search Options** button near the upper right corner of the window, and specify the appropriate filters.

   You can refresh the display by clicking the refresh button near the Search Options button. Note that when you refresh the Log Viewer by clicking the refresh button, the search options and sort order are reset to the default settings.

   If you want the log viewer to be refreshed automatically after every 15 seconds, select the **Auto Refresh** check box.

Viewing Logs Using the CLI

- To view the access log for an instance or a virtual server, run the `get-access-log` command.

  For example, the following command displays the access-log records with status=304 for the instance of the configuration `soa` running on the node `example.com`.

```
tadm> get-access-log --status-code=304 --config=soa example.com
format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
"%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status%
```

```
%Req->srvhdrs.content-length%
10.177.243.207 - - [25/Aug/2011:04:41:34 -0700] "GET / HTTP/1.1" 304 0
10.177.243.207 - - [25/Aug/2011:04:41:35 -0700] "GET / HTTP/1.1" 304 0
```

The first line shows the format that is currently defined for the access log.

To view the access log for a particular virtual server within the instance, specify the virtual server name by using the `--vs` option.

- To view the server log for an instance or a virtual server, run the `get-log` command.

  For example, the following command displays the server-log records with log level `warning:1` or higher for the instance of the configuration `soa` running on the node `example.com`.

  ```
  tadm> get-log --log-level=warning:1 --config=soa example.com
  ```

  To view the server log for a particular virtual server within the instance, specify the virtual server name by using the `--vs` option.

For more information about `get-access-log` and `get-log`, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

12.3 Configuring Log Preferences

When you create a configuration, the server and access logs are enabled with certain default settings. You can change the server log level, file name, and location. You can change the access log format, file name, and location. You can also disable the access log. If you change the location of the server log, you should restart the instance for the change to take effect.

The log preferences defined in a configuration are applicable to all the virtual servers in the configuration. At the virtual-server level, you can define the access-log location and format, and the server-log location.

You can configure log preferences for Oracle Traffic Director instances by using either the administration console or the CLI.

Note:

- To change the log preferences for the administration server, you should change the properties of the administration server as described in Section 2.6, "Changing Administration Server Settings."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Configuring Log Preferences Using the Administration Console

To configure log preferences for a configuration or a virtual server by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

A list of the available configurations is displayed.

3. Select the configuration for which you want to configure log preferences.
   - To change log preferences for the entire configuration, select **Logging** in the navigation pane.
   - To set or change the log preferences for a specific virtual server in the configuration, expand **Virtual Servers** in the navigation pane, select the virtual server for which you want to define log preferences, and then select **Logging**.

   The Log Preferences page is displayed.
4. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   For information about specifying a custom access-log format, see "Using the Custom Access-Log Format" in the *Oracle Traffic Director Configuration Files Reference*.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
5. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Log Preferences Using the CLI

- To view the current access-log preferences for a configuration or a virtual server, run the `get-access-log-prop` command.

  For example, the following command displays the access-log preferences for the configuration `soa`.

```
tadm> get-access-log-prop --config=soa
enabled=true
file=../logs/access.log
format=%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
"%Req->reqpb.clf-request%" %Req->srvhdrs.clf-status%
%Req->srvhdrs.content-length% %Req->vars.ecid%
mode=text
```

  To view the access-log preferences for a particular virtual server in the configuration, run `get-access-log-prop` with the `--vs` option.
- To set or change access-log preferences for a configuration or a virtual server, run the `enable-access-log` command.

  For example, the following command changes the location of the access log for the configuration `soa` to `INSTANCE_HOME/net-`*`config_name`*`/logs/access`.

```
tadm> enable-access-log --config=soa --file=../logs/access
OTD-70201 Command 'enable-access-log' ran successfully.
```

> **Note:** If you specify a relative path for the log-file directory, the path is taken to be relative to the `config` directory of the instance.

To set or change the access-log location and format for a particular virtual server in the configuration, run `enable-access-log` with the `--vs` option.

For information about specifying a custom access-log format, see "Using the Custom Access-Log Format" in the *Oracle Traffic Director Configuration Files Reference*.

- To disable the access log for a configuration or a virtual server, run the `disable-access-log` command, as shown in the following example:

```
tadm> disable-access-log --config=soa
OTD-70201 Command 'disable-access-log' ran successfully.
```

To disable the access log for a particular virtual server in the configuration, run `disable-access-log` with the `--vs` option.

- To view the current server-log preferences for a configuration, run the `get-log-prop` command.

For example, the following command displays the server-log preferences for the configuration `soa`.

```
tadm> get-log-prop --config=soat
create-console=false
log-file=../logs/server.log
log-to-syslog=false
log-virtual-server-name=false
log-stdout=true
log-level=NOTIFICATION:1
log-to-console=true
log-stderr=true
```

- To view the server-log location for a particular virtual server in a configuration, run the `get-virtual-server-prop` command, as shown in the following example:

```
tadm> get-virtual-server-prop --config=soa --vs=vs1 log-file
```

- To set or change server-log preferences for a configuration, run the `set-log-prop` command. Note that if you change the location of the server log, you should restart the instance for the change to take effect.

For example, the following command changes the server-log level for the configuration `soa` to `WARNING:1`.

```
tadm> set-log-prop --config=soa log-level=WARNING:1
OTD-70201 Command 'set-log-prop' ran successfully.
```

- To set or change server-log preferences for a virtual server, run the `set-virtual-server-prop` command.

For example, the following command changes the server-log location for the virtual server `vs1` in the configuration `soa` to `INSTANCE_HOME/net-`*`config_name`*`/log/server`.

```
tadm> set-virtual-server-prop --config=soa --vs=vs1 log-file=../log/server
OTD-70201 Command 'set-virtual-server-prop' ran successfully.
```

Note: If you specify a relative path for the log-file directory, the path is taken to be relative to the `config` directory of the instance.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command. If you change the location of the server log, you should restart the instance for the change to take effect

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

12.4 About Log Rotation

You can configure Oracle Traffic Director to automatically *rotate* (archive) the logs at specified intervals. You can also rotate the logs manually whenever required.

When the logs are rotated, the old log files are renamed with a suffix indicating the rotation date (in the `yyyymmdd` format) and 24-hour time (in the `hhmm` format). For example, the file name of the server log archive created at 11 p.m. on August 25, 2011 would be `server-201108252300.log`.

After log rotation, the server and access logs are re-initialized.

For information about how to rotate logs, see Section 12.5, "Rotating Logs Manually" and Section 12.6, "Configuring Oracle Traffic Director to Rotate Logs Automatically."

Note: Rotate Access Log event will also rotate TCP access logs.

12.5 Rotating Logs Manually

You can rotate the server and access logs of Oracle Traffic Director instances manually by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Rotating Logs Manually Using the Administration Console

To rotate logs by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to rotate logs.

   To rotate logs for all the instances of the selected configuration, do the following:

   a. In the navigation pane, select **Logging**.

      The Log Preferences page is displayed.

b. Go to the **Log Rotation** section of the page.

c. If you want Oracle Traffic Director to run a specific command on the rotated log files, specify the absolute path to the required command in the **Archive Command** field.

For example, if you specify `/usr/bin/gzip` as the archive command, after rotating the logs, Oracle Traffic Director compresses the rotated log files by running the following commands:

```
$ /usr/bin/gzip access-yyyymmddhhmm.log
$ /usr/bin/gzip server-yyyymmddhhmm.log
```

d. Click **Rotate Logs Now**.

The server and access logs, including any virtual server-specific logs, for all the instances of the configuration are archived.

To rotate logs for a specific instance of the selected configuration, do the following:

a. In the navigation pane, select **Instances**.

The Instances page is displayed.

b. Click the **Rotate Logs** button for the required instance.

The server and access logs, including any virtual server-specific logs, for the selected instance are archived.

A message is displayed in the Console Messages pane confirming that the logs were rotated.

Rotating Logs Manually Using the CLI

To rotate logs for one or more instances of a configuration, run the `rotate-log` command.

For example, the following command rotates the access and server logs, including any virtual server-specific logs, for the instance of the configuration `soa` running on the nodes `soa1.example.com` and `soa2.example.com`.

```
tadm> rotate-log --config=soa soa1.example.com soa2.example.com
OTD-70201 Command 'rotate-log' ran successfully.
```

If you do not specify any node, the logs are rotated for all the instances of the configuration.

Note: If you want Oracle Traffic Director to run a specific command on the rotated log files, specify the absolute path to the required command by running the `set-log-prop` command as shown in the following example:

```
tadm> set-log-prop --config=soa archive-command=/usr/bin/gzip
OTD-70201 Command 'set-log-prop' ran successfully.
```

In this example, after rotating the logs, Oracle Traffic Director compresses the rotated log files by running the following commands:

```
$ /usr/bin/gzip access-yyyymmddhhmm.log
$ /usr/bin/gzip server-yyyymmddhhmm.log
```

For more information about `rotate-log` and `set-log-prop`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

12.6 Configuring Oracle Traffic Director to Rotate Logs Automatically

You can configure Oracle Traffic Director to rotate logs automatically at specified times or intervals by creating log-rotation events.

You can create log-rotation events by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Creating Log-Rotation Events Using the Administration Console

To create log-rotation events by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create log-rotation events.
4. In the navigation pane, select **Logging**.

   The Log Preferences page is displayed.
5. Go to the **Log Rotation** section of the page.
6. If you want Oracle Traffic Director to run a specific command on the rotated log files, specify the absolute path to the required command in the **Archive Command** field.

   For example, if you specify `/usr/bin/gzip` as the archive command, after rotating the logs, Oracle Traffic Director compresses the rotated log files by running the following commands:

   ```
   $ /usr/bin/gzip access-yyyymmddhhmm.log
   $ /usr/bin/gzip server-yyyymmddhhmm.log
   ```
7. Click **New Event**.

   The New Log Rotation Event dialog box is displayed.
8. Specify whether the event is for the server log or the access log.
9. Specify the interval or time of the day at which the log should be updated, and then click **OK**.
   - A message, confirming creation of the event, is displayed in the Console Messages pane.
   - The new event is displayed in the Log Rotation Events list.
     - New events are enabled by default. To change the status, select the **Enable/Disable** check box.
     - To delete an event, click the **Delete** button.

- In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Creating Log-Rotation Events Using the CLI

To create log-rotation events, run the `create-event` command.

For example, the following commands configure Oracle Traffic Director to rotate the access logs and server logs for all instances of the configuration `soa` after every 3600 seconds and 7200 seconds respectively.

```
tadm> create-event --config=soa --command=rotate-access-log --interval=3600
OTD-70201 Command 'create-event' ran successfully.

tadm> create-event --config=soa --command=rotate-log --interval=7200
OTD-70201 Command 'create-event' ran successfully.
```

Note: If you want Oracle Traffic Director to run a specific command on the rotated log files, specify the absolute path to the required command by running the `set-log-prop` command as shown in the following example:

```
tadm> set-log-prop --config=soa archive-command=/usr/bin/gzip
OTD-70201 Command 'set-log-prop' ran successfully.
```

In this example, after rotating the logs, Oracle Traffic Director compresses the rotated log files by running the following commands:

```
$ /usr/bin/gzip access-yyyymmddhhmm.log
$ /usr/bin/gzip server-yyyymmddhhmm.log
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about `create-event`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

13

Monitoring Oracle Traffic Director Instances

Oracle Traffic Director records statistics about server activity at different levels—instances, virtual servers, listeners, connections, and origin servers. For example, for each instance of a configuration, Oracle Traffic Director collects statistics about the duration for which the instance has been running, number of requests processed, volume of data received and sent, number of responses that the instance sent of each type, average load, and so on. Similarly for each virtual server in an instance, Oracle Traffic Director collects statistics about the number of requests processed, volume of data received and sent, and the number of responses of each type. For a full list of the metrics that Oracle Traffic Director collects, see Appendix A, "Metrics Tracked by Oracle Traffic Director."

This chapter describes the monitoring capabilities of Oracle Traffic Director. It contains the following sections:

13.1 Methods for Monitoring Oracle Traffic Director Instances

Table 13–1 summarizes the methods that you can use to view statistical data about an instance of a configuration and about individual virtual servers within an instance.

Table 13–1 Methods for Monitoring Oracle Traffic Director Instances

Monitoring Method	Requirements	Advantages
CLI ■ For one or all instances of a configuration: `get-config-stats` ■ For a specific instance: Summary in plain-text format: `get-perfdump` Detailed report in XML format: `get-stats-xml` ■ For a specific virtual server within one or all instances of a configuration: `get-virtual-server-stats` ■ For a specific origin-server pool: `get-origin-server-stats` See Section 13.4, "Viewing Statistics Using the CLI."	Administration server must be running.	Enabled by default. Accessible even when request-processing threads are hanging.
Browser ■ Detailed statistics for a specific virtual server in XML format ■ Summary report for a specific virtual server in plain-text format See Section 13.5, "Viewing stats-xml and perfdump Reports Through a Browser."	Must be enabled and configured explicitly. See Section 13.3, "Configuring URI Access to Statistics Reports."	The administration server need not be running. It is sufficient if the instance is running.
SNMP	Must be configured explicitly. See Section 13.6, "Monitoring Using SNMP."	Statistics available through network management systems.

13.2 Configuring Statistics-Collection Settings

When you create an Oracle Traffic Director configuration, statistics collection is enabled by default, with five seconds as the update interval. You can disable, enable, and configure statistics collection by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Configuring Statistics-Collection Settings Using the Administration Console

To configure statistics-collection settings by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."

2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration for which you want to configure statistics-collection settings.

4. In the navigation pane, expand Advanced Settings, and select **Monitoring**.

   The Monitoring Settings page is displayed.

5. Go to the **Statistics Collection** section of the page.

6. Specify the parameters that you want to change.

   Note: When deciding the statistics-collection interval, remember that frequent collection of statistics affects performance.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

7. After making the required changes, click **Save**.

   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Statistics-Collection Settings Using the CLI

- To view the current statistics-collection properties, run the `get-stats-prop` command, as shown in the following example:

```
tadm> get-stats-prop --config=soa
enabled=true
interval=15
profiling=true
```

- To configure statistics-collection properties, run the `set-stats-prop` command.

  For example, the following command changes the interval at which statistics are updated for the configuration `soa` to 10 seconds.

```
tadm> set-stats-prop --config=soa interval=10
OTD-70201 Command 'set-stats-prop' ran successfully.
```

  For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

  For more information about `get-stats-prop` and `set-stats-prop`, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

13.3 Configuring URI Access to Statistics Reports

As described in Section 13.1, "Methods for Monitoring Oracle Traffic Director Instances," in addition to viewing activity statistics by using the CLI, you can view the following reports through a URI.

- `stats-xml`: Detailed statistics in XML format. For a sample, see Section 13.7.
- `perfdump`: A summary report in plain-text format containing a subset of the data in the `stats-xml` report. For a sample, see Section 13.8. Note that though you enable the `perf-dump` report at the virtual-server level, the data in the report is aggregated at the instance level.

> **Note:** If you enable URI access to statistics reports, then you should be aware that this URL can be visible to end users by accessing `/.perfdump`. The administrator should block this access through an external hardware load balancer.

Relative Advantages of URI-Based and CLI Access to Statistics Reports

- The administration server need not be running for users to access the `stats-xml` and `perfdump` reports through URIs. When compared with accessing statistics by using the CLI, accessing URI-based reports involves lower processing overhead.
- Access to statistics by using the CLI is enabled by default, but to view statistics through the browser, you should explicitly enable URI-based reporting and specify the URIs at which users can access the reports.

You can configure URI-based reporting of statistics by using either the administration console or the CLI.

Configuring URI Access to Statistics Using the Administration Console

To configure URI-based reporting by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to configure URI-based reports.
4. In the navigation pane, expand **Virtual Servers**, and select the virtual server for which you want to configure URI-based reports.

   The Virtual Server Settings page is displayed.
5. Go to the **Monitoring** section of the page.
   - To enable URI-based reporting in XML format, select the **XML Report** check box and specify a valid URI.
   - To enable URI-based reporting in plain-text format, select the **Plain Text Report** check box and specify a valid URI for the report.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

At any time, you can discard the changes by clicking the **Reset** button.

6. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring URI Access to Statistics in XML Format Using the CLI

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

- To view the current XML reporting settings, run the `get-stats-xml-prop` command, as shown in the following example:

```
tadm> get-stats-xml-prop --config=soa --vs=vs1
enabled=false
uri=/stats-xml(|/*)
```

- To enable and configure URI-based XML reporting, run the `enable-stats-xml` command.

  For example, the following command enables URI-based statistics reporting in XML format for the virtual server `vs1` in the configuration `soa` and specifies that the report should be available at the URI `/stats`.

```
tadm> enable-stats-xml --config=soa --vs=vs1 --uri=/stats
OTD-70201 Command 'enable-stats-xml' ran successfully.
```

- To disable URI-based XML reporting, run the `disable-stats-xml` command, as shown in the following example:

```
tadm> disable-stats-xml --config=soa --vs=vs1
OTD-70201 Command 'disable-stats-xml' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

Configuring URI Access to Statistics in Plain-Text Format Using the CLI

- To view the plain-text reporting settings, run the `get-perfdump-prop` command, as shown in the following example:

```
tadm> get-perfdump-prop --config=soa --vs=vs1
enabled=true
uri=/.perf
```

- To enable and configure the plain-text reporting, run the `enable-perfdump` command.

  For example, the following command enables URI-based statistics reporting in plain-text format for the virtual server `vs1` in the configuration `soa` and specifies that the report should be available at the URI `/perf`.

```
tadm> enable-perfdump --config=soa --vs=vs1 --uri=/perf
```

```
OTD-70201 Command 'enable-perfdump' ran successfully.
```

- To disable URI-based plain-text reporting, run the `disable-perfdump` command, as shown in the following example:

```
tadm> disable-perfdump --config=soa --vs=vs1
OTD-70201 Command 'disable-perfdump' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

13.4 Viewing Statistics Using the CLI

By using the CLI, you can view statistics for one or all instances of a configuration, for a specific virtual server, for a specific origin-server pool, and for a specific TCP proxy.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

- To view statistics for one or all instances of a configuration, run the `get-config-stats` command, as shown in the following example:

```
tadm> get-config-stats --config=soa
countRequests=20
rpsLast1MinAvg=0.06666667
rpsLast5MinAvg=0.013377926
rpsLast15MinAvg=0.0022099447
countErrors=2
epsLast1MinAvg=0.0
epsLast5MinAvg=0.0
epsLast15MinAvg=0.0
maxResponseTime=23.001
rtLast1MinAvg=1.0
rtLast5MinAvg=1.0
rtLast15MinAvg=1.0
```

  To view statistics for a specific instance of a configuration, run the `get-config-stats` command with the `--node` option.

- To view aggregated virtual-server statistics for one or all instances of a configuration, run the `get-virtual-server-stats` command, as shown in the following example:

```
tadm> get-virtual-server-stats --config=soa --vs=vs1
count200=9
count2xx=9
count302=0
count304=6
count3xx=6
count400=0
count401=0
count403=0
count404=4
count4xx=4
```

```
count503=0
count5xx=2
countBytesReceived=42215
countBytesTransmitted=69298
countErrors=2
countOpenConnections=0
countOther=0
countRequests=21
rateBytesTransmitted=0
vsName=vs1
webapp-firewall.countRequestsAllowed=5
webapp-firewall.countRequestsDenied=2
webapp-firewall.countRequestsDenyDetected=0
webapp-firewall.countRequestsDropDetected=0
webapp-firewall.countRequestsDropped=1
webapp-firewall.countRequestsIntercepted=10
webapp-firewall.countRequestsRedirectDetected=0
webapp-firewall.countRequestsRedirected=2
websocket.countActiveConnections=1
websocket.countBytesReceived=500
websocket.countBytesTransmitted=200
websocket.countRequestsAborted=0
websocket.countRequestsTimedout=0
websocket.countUpgradeRequests=1
websocket.countUpgradeRequestsFailed=0
websocket.countUpgradeRequestsRejected=1
websocket.millisecondsConnectionActiveAverage=1000
```

To view virtual-server statistics for a specific instance of a configuration, run the `get-virtual-server-stats` command with the `--node` option.

- To view statistics for a specific origin-server pool, run the `get-origin-server-stats` command, as shown in the following example:

```
tadm> get-origin-server-stats --config=soa --node=soa.example.com
--origin-server-pool=wls1
origin-server.1.backup=0
origin-server.1.countActiveConnections=0
origin-server.1.countBytesReceived=11776
origin-server.1.countBytesTransmitted=15024
origin-server.1.countConnectAttempts=41
origin-server.1.countConnectFailures=0
origin-server.1.countIdleConnections=1
origin-server.1.countMarkedOffline=0
origin-server.1.countRequests=44
origin-server.1.countRequestsAborted=0
origin-server.1.countRequestsTimedout=0
origin-server.1.discovered=0
origin-server.1.name=soa-app.example.com:1900
origin-server.1.online=1
origin-server.1.rampedUp=1
origin-server.1.secondsOnline=20
origin-server.1.type=http
origin-server.1.websocket.countRequests=2
origin-server.1.websocket.countUpgradeRejectedRequests=1
origin-server.1.websocket.countFailedStrictRequests=1
origin-server.1.websocket.countUpgradedRequests=1
origin-server.1.websocket.countAbortedRequests=0
origin-server.1.websocket.countTimeoutRequests=0
origin-server.1.websocket.countBytesReceived=500
origin-server.1.websocket.countBytesTransmitted=200
```

```
origin-server.1.websocket.countActiveConnections=1
origin-server.1.websocket.millisecondsConnectionActiveAverage=1000
...and so on for each of the origin servers in the specified pool
```

- To view detailed statistics for an instance in XML format, run the `get-stats-xml` command, as shown in the following example:

  ```
  tadm> get-stats-xml --config=soa --node=soa.example.com
  ```

  For a sample of the report, see Section 13.7.

- To view a summary of the statistics for an instance in plain-text format, run the `get-perfdump` command, as shown in the following example:

  ```
  tadm> get-perfdump --config=soa --node=soa.example.com
  ```

  For a sample of the report, see Section 13.8.

- To view statistics for a TCP proxy, run the `get-tcp-proxy-stats` command, as shown in the following example:

  ```
  tadm> get-tcp-proxy-stats --config=soa --tcp-proxy=tcp_proxy1
  interfaces:*:9898
  countActiveConnections=3
  countRequests=10
  countRequestsAborted=2
  countRequestsTimeout=4
  countBytesReceived=400
  countBytesTransmitted=200
  milliSecondsConnectionActiveAverage=1600
  mode=1
  name=tcp-proxy-1
  ```

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

13.4.1 Automating Retrieval of Monitoring Statistics

Oracle Traffic Director allows you to continuously retrieve monitoring statistics in an automated manner using a `cron` job. To enable automated collection of statistics:

Create a file such as `.tadmrc` within the Oracle Traffic Director admin user home directory:

```
cat > ~/.tadmrc
set tadm_password  <
[ CTRL-D ]
```

The administrator can continuously collect Oracle Traffic Director monitoring statistics every 5 minutes through a `cron` job, for example:

```
<otd-install-root>/bin/tadm get-perfdump –user=.. –config=..   –node=..  >>
<output-file>
```

13.5 Viewing stats-xml and perfdump Reports Through a Browser

If you enable URI access to statistics as described in Section 13.3, you can access the `stats-xml` and `perfdump` reports through a browser by using the following URL:

`http://`*`host`*`:`*`port`*`/`*`uri`*

`host` and `port` are the IP address (or host name) and port number of the virtual server for which you enabled URI access to statistics. `uri` is the location that you specified while enabling URI access. Note that if a virtual server is associated with multiple listeners, you can use the address `host:port` of any of the listeners to access the URI-based reports.

- For example, if `/perfdump` is the configured URI for the plain-text report for the virtual server `soa.example.com:1904`, the URL that you should use to access the report would be the following:

  ```
  http://soa.example.com:1904/perfdump
  ```

  In the URL, you can also specify the interval, in seconds, after which the browser should refresh the `perfdump` report automatically, as shown in the following example:

  ```
  http://soa.example.com:1904/perfdump?refresh=5
  ```

- Similarly, if `/stats-xml` is the configured URI for the XML report for the virtual server `soa.example.com:1904`, the URL that you should use to access the XML report would be the following:

  ```
  http://soa.example.com:1904/stats-xml
  ```

  You can limit the data that the XML report provides by specifying a URL query string indicating the elements that should not be displayed. If you do not include a query string, all the elements in the XML report are displayed.

  For example, the query string specified in the following URL suppresses display of the `virtual-server` and `server-pool` elements in the XML report.

  ```
  http://soa.example.com:1904/stats-xml?virtual-server=0&server-pool=0
  ```

  The following list shows the hierarchy of elements in the statistics XML report. Note that when you opt to suppress an element in the report, the child elements of that element are also suppressed.

  ```
  server
    connection-queue
    thread-pool
    profile (if profiling is enabled)
    process
      connection-queue-bucket
      thread-pool-bucket
      dns-bucket
      keepalive-bucket
      compression-bucket
      decompression-bucket
      thread
        request-bucket
        profile-bucket
    virtual-server
      request-bucket
      profile-bucket
    server-pool
      origin-server-bucket
    cache-bucket
    cpu-info
  ```

13.6 Monitoring Using SNMP

Simple Network Management Protocol (SNMP) is a standard that enables management of devices in a network from a network management application running on a remote system. The network management application might, for example, show which servers in the network are running or stopped at any point in time, and the number and type of error messages received.

You can use SNMP to monitor the Oracle Traffic Director instances. To be able to do this, you should do the following:

- Configure the instances to support monitoring through SNMP.
- Configure the SNMP subagent on the nodes.
- Start the SNMP subagent on the nodes.

This section contains the following topics:

13.6.1 Configuring Oracle Traffic Director Instances for SNMP Support

When you create a configuration, support for monitoring the instances through SNMP is enabled by default. You can disable, enable, and configure support for SNMP monitoring by using either the administration console or the CLI.

Configuring SNMP Support Using the Administration Console

To enable SNMP support for a configuration by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to enable SNMP support.
4. In the navigation pane, expand **Advanced Settings** and select **Monitoring**.

   The Monitoring Settings page is displayed.
5. In the **SNMP** section of the page, select the **SNMP** check box. The other parameters in the section are optional.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
6. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.

- In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring SNMP Support Using the CLI

- To view the current SNMP settings for a configuration, run the `get-snmp-prop` command, as shown in the following example:

```
tadm> get-snmp-prop --config=soa
enabled=false
```

- To enable SNMP support, run the `set-snmp-prop` command, as shown in the following example:

```
tadm> set-snmp-prop --config=soa enabled=true
OTD-70201 Command 'set-snmp-prop' ran successfully.
```

  For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

13.6.2 Configuring the SNMP Subagent

When you create an Oracle Traffic Director node (see Section 3.1), an SNMP *subagent* is created automatically. The SNMP subagent collects information about the instances running on the node. The SNMP subagent runs as the admin-server user. Oracle Traffic Director reads the `admin-server/config/server.xml` to read the <user>, which is the run as user for the SNMP subagent.

The SNMP subagent's configuration settings, including the frequency at which the subagent updates statistics, the duration after which cached statistics are timed out, and the port through which the subagent process communicates, are stored in the following file:

INSTANCE_HOME`/admin-server/config/snmpagt.conf`

You can configure the SNMP subagent's settings by editing the `snmpagt.conf` file. Table 13–2 lists the key SNMP subagent parameters.

Table 13–2 SNMP Subagent Configuration Parameters

Parameter in smnpagt.conf	Description	Default Value
`agentAddress`	Ports at which the SNMP subagent receives requests	11161
`statInterval`	Statistics update frequency (seconds)	5
`cacheTimeOut`	Cache timeout period (seconds)	5

The syntax for entries in `snmpagt.conf` should be as described in the documentation for `snmpd.conf` at: `http://www.net-snmp.org/docs/man/snmpd.conf.html`.

After configuring the SNMP subagent on a node, you should start it. The subagent then begins collecting statistics about the Oracle Traffic Director instances on the node. You can manage the SNMP subagent life cycle as a service through Oracle Enterprise Linux.

13.6.3 Starting and Stopping the SNMP Subagent

You can start and stop the SNMP subagent on a node by using either the administration console or the CLI.

Starting and Stopping the SNMP Subagent Using the Administration Console

To start or stop the SNMP subagent on a node by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Nodes** button that is situated near the upper left corner of the page.

   A list of available nodes is displayed.
3. From the list of nodes, select the node for which you want to start or stop the SNMP subagent.

   The General Settings page is displayed.

   - To start the SNMP subagent, click **Start SNMP Subagent**. The status changes to **Running**.
   - To stop the subagent, click **Stop SNMP Subagent**. The status changes to **Running**.
4. Specify the parameters that you want to change, and then click **Save**.

   A message is displayed in the Console Messages pane indicating that the updated settings are saved.
5. Restart the administration server by clicking **Restart** in the Common Tasks pane.

Starting and Stopping the SNMP Subagent Using the CLI

- To start the SNMP subagent on one or more nodes, run the `start-snmp-subagent`, as shown in the following example:

```
tadm> start-snmp-subagent --user=admin --port=3002 node1.example.com
node2.example.com
OTD-70210 Successfully started the SNMP subagent.
```

Note: Alternatively, you can start the SNMP agent in **agentx** mode, by specifying the `--agentx` option when you run the `start-snmp-subagent` command.

In agentx mode, the SNMP agent needs to communicate with the operating-system master agent (`snmpd`). So you must configure `snmpd` to listen to the agentx protocol, by doing the following:

1. Enable agentx by adding the following token to the operating-system master agent (`snmpd`) located at (`/etc/snmp/snmpd.conf`). This token enables the master agent to connect to the agentx paths you specify.

   ```
   master agentx
   ```

2. Specify the socket path and socket path permissions in the *ORACLE_HOME*`/admin-server/config/snmpagt.conf` file, as shown in the following example:

   Before configuring for agentx

   ```
   agentuser admin123
   agentxsocket /tmp/snmpagt-e6d7cd20/snmpagt.socket
   ```

   After configuring for agentx

   ```
   agentxsocket /tmp/snmpagt-e6d7cd20/snmpagt.socket
   agentxperms 0755 0755 admin123 admin123
   ```

3. Start `snmpd` daemon manually.

- To stop the SNMP subagent on one or more nodes, run the `stop-snmp-subagent`, as shown in the following example:

  ```
  tadm> stop-snmp-subagent --user=admin --port=3002 node1.example.com
  OTD-70210 Successfully stopped the SNMP subagent.
  ```

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

13.6.4 Viewing Statistics Using snmpwalk

Note: The prerequisites for using `snmpwalk` are as follows:

- **For Linux**: Make sure the contents snmpwalk package `net-snmp-utils-5.3.2.2-9.0.1.el5_5.1` RPM or higher and standard MIBS package `net-snmp-5.3.2.2-9.0.1.el5_5.1` RPM or higher are installed.
- **For Solaris**: Make sure the package located at `system/management/snmp/net-snmp` is installed. This package contains contents snmpwalk and standards MIBS.

Note: Prior to using `snmpwalk`, if required, you can set most of the `snmpwalk` options in the `snmp.conf` file, located at `<user-home>/.snmp/snmp.conf`. The advantage of setting various options in `snmp.conf` is that after setting the options, you can run the `snmpwalk` command without specifying the options that are already set in `snmp.conf`. For example, `snmp.conf` enables you to set the following options:

```
defaultport  11161
defversion  2c
defcommunity  public
mibdirs  +/usr/local/share/snmp/mibs #
mibdirs  + <otd_install_root>/lib/snmp #
mibs +ORACLE-TRAFFICDIRECTOR-MIB
```

After setting the above options, `snmpwalk` can be run as follows:

```
snmpwalk  <hostname> ORACLE-TRAFFICDIRECTOR-MIB::instanceTable
```

For information about all the options that can be set using snmp.conf, see the man-pages for snmp.conf.

SNMP Version 2c

You can view statistics collected by the SNMP subagent, by using the `snmpwalk` command-line utility that is available in the Net-SNMP suite of applications (`http://www.net-snmp.org`).

The following is the syntax of the `snmpwalk` command:

```
> snmpwalk -c public -v 2c host:port oid
```

- `host` is the host name of the Oracle Traffic Director node that you want to monitor.
- `port` is the listen port of the SNMP subagent on the node. The default port specified in the `snmpagt.conf` file is `11161`.
- `oid` is the unique object identifier series for which you want to view statistics. The OID for the Oracle Traffic Director product is `1.3.6.1.4.1.111.19.190`.

Note: OIDs are assigned and maintained by the Internet Assigned Numbers Authority. In the OID for Oracle Traffic Director, the first six numbers, `1.3.6.1.4.1`, represent private enterprises, `111` is the unique identifier for Oracle and `19.190` represents the Oracle Traffic Director product. For more information about the structure of OIDs, see RFC 2578 (`http://tools.ietf.org/html/rfc2578`).

SNMP Version 3

To monitor statistics by using SNMP v3, do the following:

1. Create an SNMP v3 user by running the following command as the `root` user:

   ```
   $ sudo net-snmp-config --create-snmpv3-user -ro -a MD5 -A abcd1234 otdadmin
   ```

   This command does the following:

   - Adds the following entry in `/var/net-snmp/snmpd.conf`:

     ```
     createUser otdadmin MD5 "abcd1234" DES
     ```

- Adds the following entry in `/etc/net-snmp/snmp/snmpd.conf`:

  ```
  rouser otdadmin
  ```

2. Start and stop `snmpd`.

   ```
   $ sudo /etc/init.d/snmpd start
   Starting snmpd:                                            [  OK  ]

   $ sudo /etc/init.d/snmpd stop
   Stopping snmpd:                                            [  OK  ]
   ```

   As a result of starting and stopping `snmpd`, the `createUser` entry in the `/var/net-snmp/snmpd.conf` file changes as shown in the following example:

   ```
   usmUser 1 3 0x80001f8801819ee527 0x676164686100 0x676164686100 NULL
    .1.3.6.1.6.3.10.1.1.2
   0x8b6a9b458c0cb628aa5ba10ebbec48e7 .1.3.6.1.6.3.10.1.2.2
    0x8b6a9b458c0cb628aa5ba10ebbec48e7 ""
   ```

   In this example, `0x80001f8801819ee527` is the generated engine ID.

3. Run the SNMP agent in **agentx** mode.

   Run `snmpwalk` by using the following command. The default port for snmpd is 161

   ```
   snmpwalk -v3 -u otdadmin -l authNoPriv  -a MD5 -A abcd1234 localhost:161
   1.3.6.1.4.1
   ```

Enabling the snmpwalk Command to Show MIB Object Names Instead of Numeric OIDs

When you run the `snmpwalk` command, the output would be as follows:

```
SNMPv2-SMI::enterprises.111.19.190.1.20.1.2.0.0 = INTEGER: 645
SNMPv2-SMI::enterprises.111.19.190.1.20.1.3.0.0 = Gauge32: 4
SNMPv2-SMI::enterprises.111.19.190.1.20.1.4.0.0 = Gauge32: 4
SNMPv2-SMI::enterprises.111.19.190.1.20.1.10.0.0 = Gauge32: 0
SNMPv2-SMI::enterprises.111.19.190.1.20.1.11.0.0 = Gauge32: 3072
SNMPv2-SMI::enterprises.111.19.190.1.20.1.12.0.0 = Counter64: 0
SNMPv2-SMI::enterprises.111.19.190.1.20.1.13.0.0 = Counter64: 0
SNMPv2-SMI::enterprises.111.19.190.1.20.1.14.0.0 = STRING: "0.0000"
```

Each line in the output shows the value of a metric, but because the OID is shown in numeric format, it is difficult to identify the name of the specific metric. The `snmpwalk` utility can resolve numeric OIDs to textual names by using the management information base (MIB) definitions. For Oracle Traffic Director, the MIB definitions file is available in the following directory:

ORACLE_HOME`/lib/snmp/ORACLE-TRAFFICDIRECTOR-MIB.txt`

To enable the `snmpwalk` command to show MIB object names instead of numeric OIDs, do one of the following:

- Set the `MIBS` environment variable on the host to point to the Oracle Traffic Director MIB.

  ```
  > set env MIBS=+ORACLE-TRAFFICDIRECTOR-MIB
  ```

  Then, run the `snmpwalk` command and either `grep` the output for the required MIB object or explicitly specify the required MIB object name.

  For example, to view statistics for proxy cache parameters for an Oracle Traffic Director instance running on the node `app1`, run the following command:

```
> snmpwalk snmpwalk -c public -v 2c app1:11161
ORACLE-TRAFFICDIRECTOR-MIB::proxyCacheTable

ORACLE-TRAFFICDIRECTOR-MIB::proxyCacheEnabledFlag.0.0 = INTEGER: enabled(1)
ORACLE-TRAFFICDIRECTOR-MIB::proxyCacheCountEntries.0.0 = Counter64: 0
ORACLE-TRAFFICDIRECTOR-MIB::proxyCacheSizeHeap.0.0 = Counter64: 16498
ORACLE-TRAFFICDIRECTOR-MIB::proxyCacheCountContentHits.0.0 = Counter64: 0
ORACLE-TRAFFICDIRECTOR-MIB::proxyCacheCountContentMisses.0.0 = Counter64: 0
ORACLE-TRAFFICDIRECTOR-MIB::proxyCacheCountHits.0.0 = Counter64: 0
...
```

- Specify the Oracle Traffic Director MIB explicitly for the `snmpwalk` command by using the `-m` option.

  For example, to view the origin-server names for an Oracle Traffic Director instance running on the local host, run the following command:

```
> snmpwalk -c public -v 2c -m $ORACLE_
HOME/lib/snmp/ORACLE-TRAFFICDIRECTOR-MIB.txt localhost:11161
ORACLE-TRAFFICDIRECTOR-MIB::originServerName
```

For a list of the SNMP MIB object names that you can use to query for specific statistics, see Appendix A, "Metrics Tracked by Oracle Traffic Director."

For more information about `snmpwalk`, see the documentation at: http://www.net-snmp.org/docs/man/snmpwalk.html.

13.7 Sample XML (stats-xml) Report

This section contains a sample statistics report in XML format, which you can view by using the `get-stats-xml` command or through a URI. For more information, see Section 13.4, "Viewing Statistics Using the CLI" and Section 13.5, "Viewing stats-xml and perfdump Reports Through a Browser."

Note that the values shown in this sample report might not be meaningful. The sample report is provided here merely to indicate the metrics that the report includes and to give you a general idea about the format and structure of the report.

```
<stats versionMajor="1" versionMinor="3" flagEnabled="1">
    <server id="net-test" versionServer="Oracle Traffic Director 11.1.1.7.0
B01/31/2013 03:40 (Linux)"
            timeStarted="1362099811" secondsRunning="451" ticksPerSecond="1000"
maxProcs="1" maxThreads="10"
            flagProfilingEnabled="1" load1MinuteAverage="0.000000"
load5MinuteAverage="0.020000"
            load15MinuteAverage="0.040000" rateBytesTransmitted="173858"
rateBytesReceived="1056"
            requests1MinuteAverage="0.000000" requests5MinuteAverage="0.000000"
requests15MinuteAverage="0.000000"
            errors1MinuteAverage="0.000000" errors5MinuteAverage="0.000000"
errors15MinuteAverage="0.000000"
            responseTime1MinuteAverage="0.000000"
responseTime5MinuteAverage="0.000000"
            responseTime15MinuteAverage="0.000000">
        <connection-queue id="cq1"/>
        <thread-pool id="thread-pool-0" name="NativePool"/>
        <profile id="profile-0" name="all-requests" description="All requests"/>
        <profile id="profile-1" name="default-bucket" description="Default
bucket"/>
        <profile id="profile-2" name="cache-bucket" description="Cached
responses"/>
```

```
        <process pid="25929" mode="active" timeStarted="1362099811"
countConfigurations="1"
                 sizeVirtual="238272" sizeResident="36464"
fractionSystemMemoryUsage="0.0045">
            <connection-queue-bucket connectionQueueId="cq1"
countTotalConnections="2" countQueued="0"
                                     peakQueued="1" maxQueued="2048"
countOverflows="0" countTotalQueued="3"
                                     ticksTotalQueued="0"
countQueued1MinuteAverage="0.000000"
                                     countQueued5MinuteAverage="0.000000"
countQueued15MinuteAverage="0.000000"/>
            <thread-pool-bucket threadPoolId="thread-pool-0" countIdleThreads="1"
countThreads="1"
                                maxThreads="128" countQueued="0" peakQueued="0"
maxQueued="0"/>
            <dns-bucket flagCacheEnabled="1" countCacheEntries="0"
maxCacheEntries="1024" countCacheHits="0"
                        countCacheMisses="0" flagAsyncEnabled="0"
countAsyncNameLookups="0" countAsyncAddrLookups="0"
countAsyncLookupsInProgress="0"/>
            <keepalive-bucket countConnections="0" maxConnections="4096"
countHits="0" countFlushes="0" countRefusals="0"
                              countTimeouts="0" secondsTimeout="30"/>
            <compression-bucket countRequests="0" bytesInput="0" bytesOutput="0"
compressionRatio="0.000000"
                                pageCompressionAverage="0.000000"/>
            <decompression-bucket countRequests="0" bytesInput="0"
bytesOutput="0"/>
            <thread mode="idle" timeStarted="1362099811"
connectionQueueId="keep-alive">
                <request-bucket countRequests="0" countBytesReceived="0"
countBytesTransmitted="0" countOpenConnections="0" count2xx="0" count3xx="0"
count4xx="0" count5xx="0" countOther="0" count200="0" count302="0" count304="0"
count400="0" count401="0"count403="0" count404="0" count503="0"/>
                <profile-bucket profile="profile-0" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
                <profile-bucket profile="profile-1" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
                <profile-bucket profile="profile-2" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
            </thread>
            <thread mode="idle" timeStarted="1362099811"
connectionQueueId="keep-alive">
                <request-bucket countRequests="0" countBytesReceived="0"
countBytesTransmitted="0" countOpenConnections="0" count2xx="0" count3xx="0"
count4xx="0" count5xx="0" countOther="0" count200="0" count302="0" count304="0"
count400="0" count401="0"count403="0" count404="0" count503="0"/>
                <profile-bucket profile="profile-0" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
                <profile-bucket profile="profile-1" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
                <profile-bucket profile="profile-2" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
            </thread>
            <thread mode="idle" timeStarted="1362099811" connectionQueueId="cq1"
clientAddress="10.229.131.192">
                <request-bucket countRequests="2" countBytesReceived="336"
countBytesTransmitted="18174" countOpenConnections="0" count2xx="2" count3xx="0"
count4xx="0" count5xx="0" countOther="0" count200="2" count302="0" count304="0"
```

```
count400="0" count401="0" count403="0" count404="0" count503="0"/>
                <profile-bucket profile="profile-0" countCalls="18"
countRequests="2" ticksDispatch="0" ticksFunction="2"/>
                <profile-bucket profile="profile-1" countCalls="18"
countRequests="2" ticksDispatch="0" ticksFunction="2"/>
                <profile-bucket profile="profile-2" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
            </thread>
            <thread mode="response" timeStarted="1362099811" function="stats-xml"
connectionQueueId="cq1"
                    virtualServerId="test" clientAddress="10.159.75.10"
timeRequestStarted="1362100279014665">
                <request-bucket method="GET" uri="/stats-xml" countRequests="0"
countBytesReceived="0" countBytesTransmitted="0" countOpenConnections="0"
count2xx="0" count3xx="0" count4xx="0" count5xx="0" countOther="0" count200="0"
count302="0" count304="0" count400="0" count401="0" count403="0" count404="0"
count503="0"/>
                <profile-bucket profile="profile-0" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
                <profile-bucket profile="profile-1" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
                <profile-bucket profile="profile-2" countCalls="0"
countRequests="0" ticksDispatch="0" ticksFunction="0"/>
            </thread>
            <tcp-thread countPairConnections="0" countConnections="0"/>
            <tcp-thread countPairConnections="0" countConnections="0"/>
        </process>
        <virtual-server id="test" mode="active" interfaces="*:7011">
            <request-bucket method="GET" uri="/stats-xml" countRequests="2"
countBytesReceived="336" countBytesTransmitted="18174" countOpenConnections="0"
count2xx="2" count3xx="0" count4xx="0" count5xx="0" countOther="0" count200="2"
count302="0" count304="0" count400="0" count401="0" count403="0" count404="0"
count503="0"/>
            <profile-bucket profile="profile-0" countCalls="18" countRequests="2"
ticksDispatch="0" ticksFunction="2"/>
            <profile-bucket profile="profile-1" countCalls="18" countRequests="2"
ticksDispatch="0" ticksFunction="2"/>
            <profile-bucket profile="profile-2" countCalls="0" countRequests="0"
ticksDispatch="0" ticksFunction="0"/>
            <websocket-bucket countUpgradeRequests="0"
countUpgradeRequestsFailed="0" countUpgradeRequestsRejected="0"
                              countActiveConnections="0" countRequestsAborted="0"
countRequestsTimedout="0"
                              countBytesReceived="0" countBytesTransmitted="0"
millisecondsConnectionActiveAverage="0"/>
        </virtual-server>
        <server-pool name="origin-server-pool-1" type="http" countRetries="0">
            <origin-server-bucket name="http://adc2120844:4005" flagOnline="1"
flagDiscovered="0" flagRampedUp="1" type="generic"
                                  flagBackup="0" secondsOnline="465"
countDetectedOffline="0" countConnectAttempts="15"
                                  countConnectFailures="0"
countClosedConnections="14" countConnectionsClosedByOriginServer="0"
                                  countActiveConnections="0"
countIdleConnections="1" countActiveStickyConnections="0"
                                  secondsKeepAliveTimeout="0"
countRequestsAborted="0" countRequestsTimedout="0"
                                  countStickyRequests="0" countRequests="0"
countHealthCheckRequests="15" countBytesTransmitted="0"
                                  countBytesReceived="0"
```

```
weightResponseTime="1.00">
                <websocket-bucket countUpgradeRequests="0"
countUpgradeRequestsFailed="0" countUpgradeRequestsRejected="0"
                                  countActiveConnections="0"
countRequestsAborted="0" countRequestsTimedout="0"
                                  countBytesReceived="0" countBytesTransmitted="0"
millisecondsConnectionActiveAverage="0"/>
            </origin-server-bucket>
        </server-pool>
        <cache-bucket flagEnabled="1" countEntries="0" sizeHeapCache="16492"
countContentHits="0" countContentMisses="0" countHits="0"
                      countRevalidationRequests="0"
countRevalidationFailures="0"/>
        <cpu-info cpu="1" percentIdle="99.224155" percentUser="0.682178"
percentKernel="0.093667"/>
        <cpu-info cpu="2" percentIdle="99.323128" percentUser="0.601763"
percentKernel="0.075109"/>
    </server>
</stats>
```

13.8 Sample Plain-Text (perfdump) Report

This section contains a sample perfump statistics report that you can view by using the `get-perfdump` command or through a URI. For information about viewing the plain-text report, see Section 13.4, "Viewing Statistics Using the CLI" and Section 13.5, "Viewing stats-xml and perfdump Reports Through a Browser."

Note that the values shown in this sample report might not be meaningful. The sample report is provided here merely to indicate the metrics that the report includes and to give you a general idea about the format of the report.

```
Oracle Traffic Director 11.1.1.7.0 B01/14/2013 04:13 (Linux)

Server started Wed Feb 27 23:53:18 2013
Process 10909 started Wed Feb 27 23:53:18 2013

ConnectionQueue:
----------------------------------------
Current/Peak/Limit Queue Length            0/0/1536
Total Connections Queued                   0
Average Queue Length (1, 5, 15 minutes)    0.00, 0.00, 0.00
Average Queueing Delay                     0.00 milliseconds

ListenSocket http-listener-1:
-----------------------
Address                   0.0.0.0:8786
Acceptor Threads          2
Default Virtual Server    config1

KeepAliveInfo:
--------------------
KeepAliveCount        0/3072
KeepAliveHits         0
KeepAliveFlushes      0
KeepAliveRefusals     0
KeepAliveTimeouts     0
KeepAliveTimeout      30 seconds

SessionCreationInfo:
------------------------
```

```
Active Sessions           0
Keep-Alive Sessions       0
Total Sessions Created    6/258

Proxy Cache:
---------------------------
Proxy Cache Enabled              yes
Object Cache Entries             0
Cache lookup (hits/misses)       0/0
Requests served from Cache       0
Revalidation (successful/total)  0/0 (  0.00%)
Heap space used                  16501

TCP thread pool :
-----------------
Enabled               yes
CountConnectionPairs  0
CountDescriptors      0
CountHalfClosed       0

Native pools:
----------------------------
NativePool:
Idle/Peak/Limit               1/1/128
Work Queue Length/Peak/Limit  0/0/0

DNSCacheInfo:
-----------------
enabled             yes
CacheEntries        0/1024
HitRatio            0/0 (  0.00%)

Async DNS disabled

Performance Counters:
-----------------------------------------------
                           Average         Total      Percent

Total number of requests:                      0
Request processing time:    0.0000        0.0000

default-bucket (Default bucket)
Number of Requests:                            0    (  0.00%)
Number of Invocations:                         0    (  0.00%)
Latency:                    0.0000        0.0000    (  0.00%)
Function Processing Time:   0.0000        0.0000    (  0.00%)
Total Response Time:        0.0000        0.0000    (  0.00%)

Origin server statistics:
-------------------------------------------------------------------------------
---------------------------------------------------------------
Pool-name             Host:Port       Status   ActiveConn  IdleConn  StickyConn
Timeouts  Aborted  Sticky-Reqs  Total-Reqs  BytesTrans  BytesRecvd

origin-server-pool-1  http://test    Offline  0         0       0         0
0        0           0           0           0
origin-server-pool-2  http://test:84  Offline  0         0       0         0
0        0           0           0           0

Sessions:
```

```
---------------------------------------------------------------------
Process  Status  Client  Age  VS  Method  URI  Function  Origin-Server


TCP Proxy:
-----------------------------------
Active Connections                  0
Avg Duration                        0.00 seconds
Requests (timeout/aborted/total)    0/0/0
```

14

Configuring Oracle Traffic Director for High Availability

This chapter describes the high-availability capabilities of Oracle Traffic Director. These capabilities are applicable only on engineered systems platforms. This chapter contains the following sections:

- Overview of High-Availability Features
- Creating and Managing Failover Groups
- Configuring Health-Check Settings for Origin-Server Pools

14.1 Overview of High-Availability Features

In the context of Oracle Traffic Director instances, high availability includes the following capabilities:

- Receive and serve client requests without downtime caused by hardware failures, kernel crashes, and network issues.
  - You can set up a highly available traffic routing and load-balancing service for your enterprise applications and services by configuring two Oracle Traffic Director instances to provide active-active or active-passive failover. For more information, see Section 14.2, "Creating and Managing Failover Groups."
  - If an Oracle Traffic Director process crashes, it restarts automatically.

    Oracle Traffic Director provides two levels of availability, application level and node level. Application level availability is the default feature and does not require any additional configuration. Application level availability ensures that the load balancing service is monitored through the Oracle Traffic Director Watchdog daemon and is available even during application level failures such as process crash. This feature ensures that Oracle Traffic Director as a software load balancer can continue to front-end requests to back-end applications even if there is a software issue within the load balancing service. The node level availability ensures that Oracle Traffic Director continues to front-end requests to back-end applications even if the system/vServer crashes because of issues such as CPU failure or memory corruption. For node level availability, Oracle Traffic Director must be installed on two compute notes or vServers, and a failover group must be configured between them.

    To provide high availability to the Oracle Traffic Director instance itself, each load balancer server instance includes at least three processes, a watchdog process, a primordial process, and one or more load balancer processes. The watchdog process spawns the primordial, which then spawns the load

balancer processes. The watchdog process and the primordial process provide a limited level of high availability within the server processes. If the load balancer process or primordial process terminates abnormally for any reason, then Oracle Traffic Director watchdog is responsible for restarting these services, to ensure that Oracle Traffic Director as a software load balancer service continues to be available. An Oracle Traffic Director instance will have exactly one watchdog process, one primordial process and one or more load balancer processes.

  - Most configuration changes to Oracle Traffic Director instances can be deployed dynamically, without restarting the instances and without affecting requests that are being processed. For configuration changes that do require instances to be restarted, the administration interfaces—CLI and administration console—display a prompt to restart the instances.

- Distribute client requests reliably to origin servers in the back end.
  - If a server in the back end is no longer available or is fully loaded, Oracle Traffic Director detects this situation automatically through periodic health checks and stops sending client requests to that server. When the failed server becomes available again, Oracle Traffic Director detects this automatically and resumes sending requests to the server. For more information, see Section 14.3, "Configuring Health-Check Settings for Origin-Server Pools."
  - In each origin-server pool, you can designate a few servers as backup servers. Oracle Traffic Director sends requests to the backup servers only when none of the primary servers in the pool is available. For more information, see Section 6.3, "Modifying an Origin-Server Pool."
  - You can reduce the possibility of requests being rejected by origin servers due to a connection overload, by specifying the maximum number of concurrent connections that each origin server can handle.

    For each origin server, you can also specify the duration over which the rate of sending requests to the server is increased. This capability helps minimize the possibility of requests getting rejected when a server that was offline is in the process of restarting.

    For more information, see Section 7.3, "Modifying an Origin Server."

14.2 Creating and Managing Failover Groups

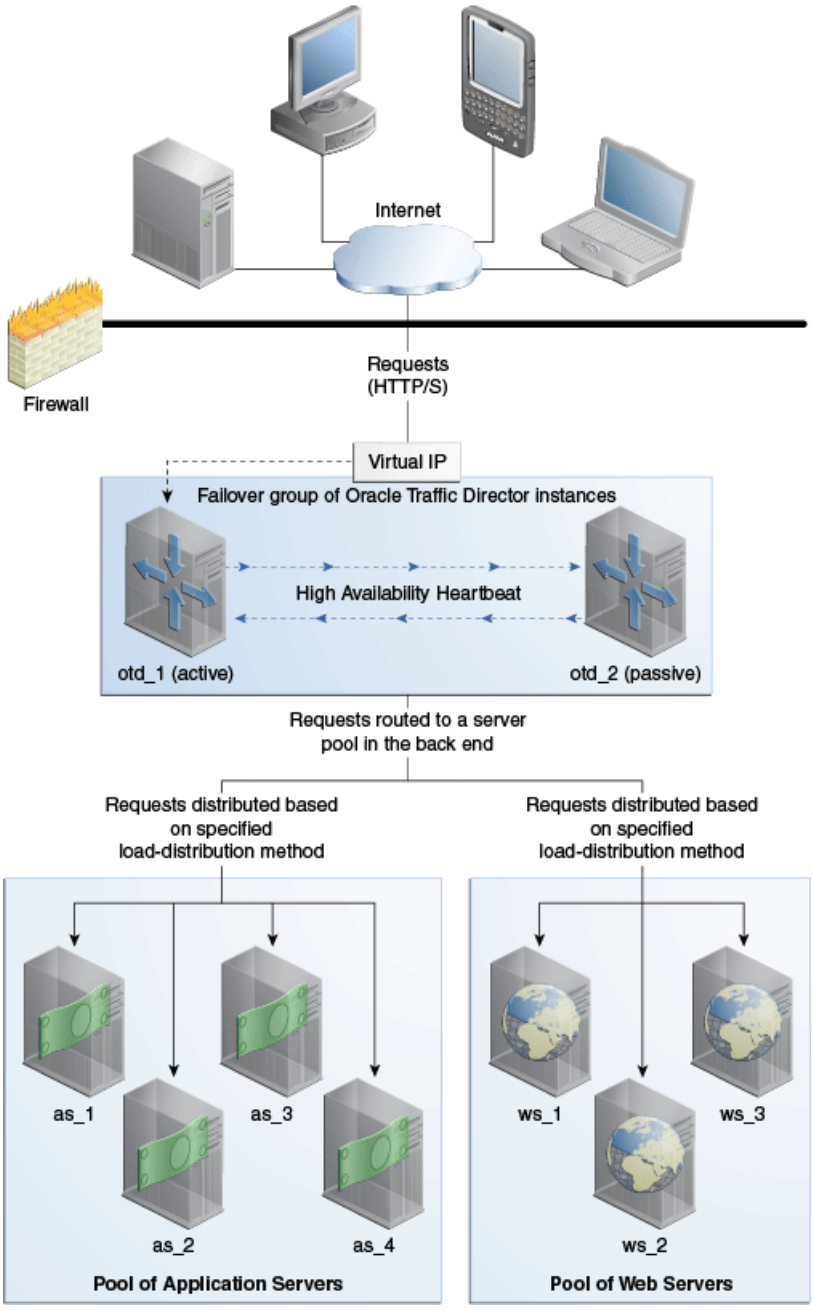
You can ensure high availability of Oracle Traffic Director instances by combining two Oracle Traffic Director instances in a *failover group* represented by one or two virtual IP (VIP) addresses. Both the hosts in a failover group must run the same operating system version, use identical patches and service packs, and run Oracle Traffic Director instances of the same configuration.

Note:

- You can create multiple failover groups for the same node, but with a distinct VIP address for each failover group.

Figure 14–1 shows Oracle Traffic Director deployed for a high-availability use case in an active-passive failover mode.

Figure 14–1 Oracle Traffic Director Network Topology: Active-Passive Failover Mode



Oracle Traffic Director network topology for an active-passive failover mode

****

The topology shown in Figure 14–1 consists of two Oracle Traffic Director instances—`otd_1` and `otd_2`—forming an active-passive failover pair and providing a single virtual IP address for client requests. When the active instance (`otd_1` in this example) receives a request, it determines the server pool to which the request should be sent and forwards the request to one of the servers in the pool based on the load distribution method defined for that pool.

Note that Figure 14–1 shows only two server pools in the back end, but you can configure Oracle Traffic Director to route requests to servers in multiple server pools.

In the active-passive setup described here, one node in the failover group is redundant at any point in time. To improve resource utilization, you can configure the two Oracle Traffic Director instances in active-active mode with two virtual IP addresses. Each instance caters to requests received on one virtual IP address *and* backs up the other instance.

This section contains the following topics:

14.2.1 How Failover Works

Oracle Traffic Director provides support for failover between the instances in a failover group by using an implementation of the Virtual Routing Redundancy Protocol (VRRP), such as keepalived for Linux and vrrpd (native) for Solaris.

Keepalived v1.2.2 is included in Oracle Linux in Exalogic environment. You need not install or configure it. Keepalived is licensed under the GNU General Public License. Keepalived provides other features such as load balancing and health check for origin servers, but Oracle Traffic Director uses only the VRRP subsystem. For more information about Keepalived, go to `http://www.keepalived.org`.

VRRP specifies how routers can failover a VIP address from one node to another if the first node becomes unavailable for any reason. The IP failover is implemented by a router process running on each of the nodes. In a two-node failover group, the router process on the node to which the VIP is currently addressed is called the master. The master continuously advertises its presence to the router process on the second node.

Caution: On a host that has an Oracle Traffic Director instance configured as a member of a failover group, Oracle Traffic Director should be the only consumer of Keepalived. Otherwise, when Oracle Traffic Director starts and stops the `keepalived` daemon for effecting failovers during instance downtime, other services using `keepalived` on the same host can be disrupted.

If the node on which the master router process is running fails, the router process on the second node waits for about three seconds before deciding that the master is down, and then assumes the role of the master by assigning the VIP to its node. When the first node is online again, the router process on that node takes over the master role. For more information about VRRP, see RFC 5798 at `http://datatracker.ietf.org/doc/rfc5798`.

14.2.2 Failover Modes

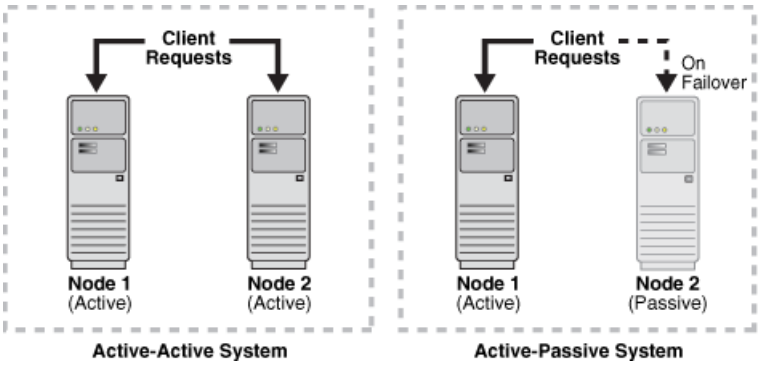
You can configure the Oracle Traffic Director instances in a failover group to work in the following modes:

- **Active-passive**: A single VIP address is used. One instance in the failover group is designated as the primary node. If the primary node fails, the requests are routed through the same VIP to the other instance.

- **Active-active**: This mode requires two VIP addresses. Each instance in the failover group is designated as the primary instance for one VIP address and the backup for the other VIP address. Both instances receive requests concurrently.

The following figure illustrates the active-active and active-passive failover modes.

Figure 14–2 Failover Modes



Failover modes

****

14.2.3 Creating Failover Groups

This section describes how to implement a highly available pair of Oracle Traffic Director instances by creating failover groups. For information about how failover works, see Section 14.2.1, "How Failover Works."

You can create a failover group by using either the administration console or the CLI.

> **Note:** The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Before You Begin

- Decide the unique **VIP address** that you want to assign to the failover group.
  - The VIP addresses must be accessible to clients.

> **Note:** To configure an active-active pair of Oracle Traffic Director instances, you would need to create two failover groups with the same instances, but with a distinct VIP address for each failover group, and with the primary and backup node roles reversed.

- Identify the **network prefix** of the interface on which the VIP should be managed. The network prefix is the subnet mask represented in the Classless Inter-Domain Routing (CIDR) format, as described in the following examples.
  - For an IPv4 VIP address in a subnet that contains 256 addresses (8 bits), the CIDR notation of the subnet mask `255.255.255.0` would be `24`, which is derived by deducting the number of addresses in the given subnet (8 bits) from the maximum number of IPv4 addresses possible (32 bits).

  - Similarly, for an IPv4 VIP address in a subnet that has 4096 addresses (12 bits), the CIDR notation of the subnet mask `255.255.240.0` would be `20` (=`32` minus `12`).
  - To calculate the CIDR notation of the subnet mask for an IPv6 subnet, you should deduct the bit-size of the subnet's address space from 128 bits, which is the maximum number of IPv6 addresses possible.

  The default network-prefix-length is `24` or `64` for an IPv4 VIP or IPv6 VIP, respectively. The default network-prefix-length is used, if not specified, for automatically choosing the NIC.

  While actually plumbing the VIP it is preferred to use the hostmask, 32 for IPv4 and 128 for IPv6, so that any outgoing traffic originating from that node does not use the VIP as the source address.

- Identify the Oracle Traffic Director **administration nodes** that you want to configure as primary and backup nodes in the failover group.

  Note that the administration nodes that you select should have Oracle Traffic Director instances present on them for the specified configuration.

- Identify the **network interface** for each node.

  If you do not specify the network interface, the administration server attempts to automatically discover a usable network interface for the specified VIP. For each network interface that is currently up on the host, the administration server compares the network part of the interface's IP address with the network part of the specified VIP. The first network interface that results in a match is used as the network interface for the VIP.

  For this comparison, depending on whether the VIP specified for the failover group is an IPv4 or IPv6 address, the administration server considers only those network interfaces on the host that are configured with an IPv4 or IPv6 address, respectively.

- You can bind to a VIP IP address within the HTTP listener by performing a system configuration that allows you to bind to a non-existing address, as a sort of forward binding. Perform one of the following system configurations:

  ```
  echo 1 > /proc/sys/net/ipv4/ip_nonlocal_bind
  ```

  or,

  `sysctl net.ipv4.ip_nonlocal_bind=1` (change in `/etc/sysctl.conf` to keep after a reboot)

  Make sure that the IP addresses of the listeners in the configuration for which you want to create a failover group are either an asterisk (`*`) or the same address as the VIP. Otherwise, requests sent to the VIP will not be routed to the virtual servers.

- Make sure that the router ID for each failover group is unique. If you do not specify the router ID, it is set to 255 for the first failover group. For every subsequent failover group that you create, the default router ID is decremented by one: 254, 253, and so on.

Creating Failover Groups Using the Administration Console

To create a failover group by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."

2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.

3. Select the configuration for which you want to create a failover group.

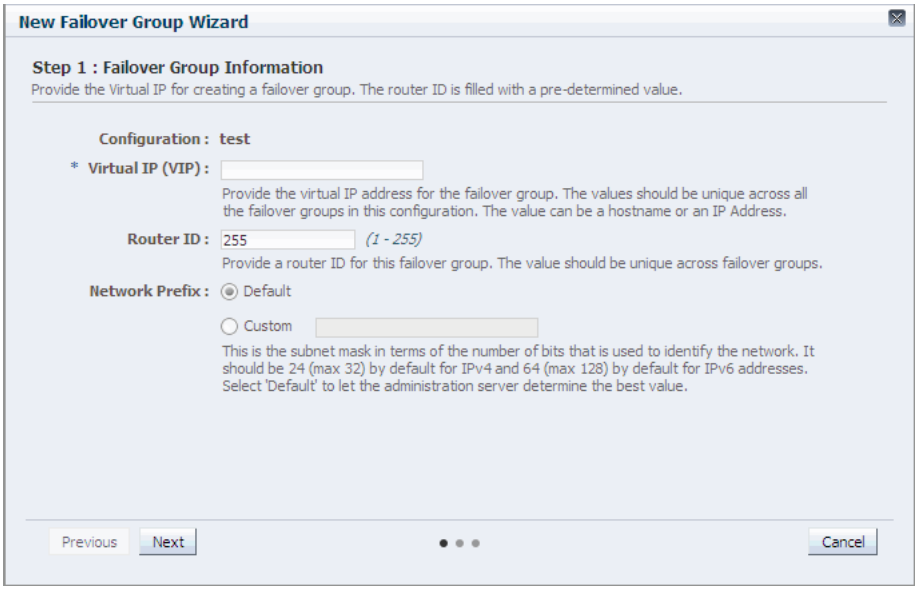
4. In the navigation pane, select **Failover Groups**.

   The Failover Groups page is displayed.

5. Click **New Failover Group**.

   The New Failover Group wizard is displayed.

Figure 14–3 New Failover Group Wizard



New Failover Group wizard

****

6. Follow the on-screen prompts to complete creation of the failover group by using the details—virtual IP address, network interface, host names of administration nodes, and so on—that you decided earlier.

   After the failover group is created, the Results screen of the New Failover Group wizard displays a message confirming successful creation of the failover group.

7. Click **Close** on the Results screen.

   The details of the failover group that you just created are displayed on the Failover Groups page.

> **Note:** At this point, the two nodes form an active-passive pair. To convert them into an active-active pair, create another failover group with the same two nodes, but with a different VIP and with the primary and backup roles reversed.

Creating Failover Groups Using the CLI

To create a failover group, run the `create-failover-group` command.

For example, the following command creates a failover group with the following details:

- Configuration: `soa`
- Primary node: `node1.example.com`
- Backup node: `node2.example.com`
- Virtual IP address: `10.229.227.80`
- Network prefix for the VIP: Not specified; so the command assumes the network prefix to be `24` (equivalent to subnet mask `255.255.255.0`)

```
> tadm create-failover-group --config=soa --virtual-ip=10.229.227.80
--primary-node=node1.example.com --backup-node=node2.example.com
OTD-70201 Command 'create-failover-group' ran successfully.
```

Note: When creating a failover group, if the administration node process is running as non-`root` on the node where the instances are located, then you must run `start-failover` on those nodes as a `root` user. This is to manually start the failover. If this command is not executed, failover will not start and there will be no high availability. For more information about `start-failover`, see the *Oracle Traffic Director Command-Line Reference*.

To enable active-active failover, create another failover group with the same two nodes, but with the primary and backup roles reversed.

For more information about `create-failover-group`, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

14.2.4 Managing Failover Groups

Oracle Traffic Director starts the `keepalived` daemon automatically when you start instances that are part of a failover group, and stops the daemon when you stop the instances. The configuration parameters for the `keepalived` daemon are stored in a file named `keepalived.conf` in the `config` directory of each instance that is part of the failover group. If the administration node process is running as non-`root` on the node where the instances are located, then you must run the `start-failover` command on those nodes as a `root` user. This is to manually start the failover. If this command is not executed, failover will not start and there will be no high availability. For more information about `start-failover`, see the *Oracle Traffic Director Command-Line Reference*.

Note: For the `keepalived` daemon to be started and stopped automatically, you must run the commands to start and stop the Oracle Traffic Director instances as the `root` user.

After creating failover groups, you can list them, view their settings, change the primary node for a failover group, switch the primary and backup nodes, and delete them. Note that to change the VIP or any property of a failover group, you should delete the failover group and create it afresh.

You can view, modify, and delete failover groups by using either the administration console or the CLI.

Note: The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

Managing Failover Groups Using the Administration Console

To view, modify, and delete failover groups by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to manage failover groups.
4. In the navigation pane, select **Failover Groups**.

   The Failover Groups page is displayed. It shows the list of available failover groups, and indicates the primary and backup nodes for each failover group.

   - To view the properties of a failover group, click its virtual IP.
   - To switch the hosts for the primary and backup nodes, click the **Toggle Primary** button. In the resulting dialog box, click **OK**.
   - To delete a failover group, click the **Delete** button. In the resulting dialog box, click **OK**.

Managing Failover Groups Using the CLI

- To view a list of the failover groups for a configuration, run the `list-failover-groups` command, as shown in the following example:

```
tadm> list-failover-groups --config=soa --verbose --all
virtual-ip      primary-node    backup-node
----------------------------------------
10.229.231.254  node1.example.com      node2.example.com
10.229.231.253  node2.example.com      node1.example.com
```

- To view the current settings of a failover group, run the `get-failover-group-prop` command, as shown in the following example:

```
tadm> get-failover-group-prop --config=soa --virtual-ip=10.229.231.254
virtual-ip=10.229.231.254
backup-node=node2.example.com
network-prefix-length=21
router-id=255
primary-node=node1.example.com
primary-nic=eth0
backup-nic=eth0
```

- To switch the primary and backup nodes in a failover group, run the `set-failover-group-primary` command.

  For example, the following command changes the primary node in the failover group represented by the VIP address `10.228.12.250` in the configuration `soa` to `app2.example.com`.

```
tadm> set-failover-group-primary --config=soa  --virtual-ip=10.228.12.250
```

```
--primary-node=app2.example.com
OTD-70201 Command 'set-failover-group-primary' ran successfully.
```

Note: If the administration node process is running as non-`root` on the node where the instances are located, then you must run `start-failover` on those nodes as a `root` user. This is to manually toggle the nodes. If this command is not executed, the nodes will not be toggled. And, when you execute `get-failover-group-prop`, the result will include the configured primary and the backup nodes, which will not be the same as the runtime primary and backup nodes.

- To delete a failover group, run the `delete-failover-group` command, as shown in the following example:

```
tadm> delete-failover-group --config=soa  --virtual-ip=10.228.12.250
OTD-70201 Command 'delete-failover-group' ran successfully.
```

Note: When deleting a failover group, if the administration node process is running as non-`root` on the node where the instances are located and if at least one failover group is still available, then you must run `start-failover` on those nodes as a `root` user. On the other hand, after deleting a failover group, if no other failover groups are available for the corresponding instances, then `stop-failover` must be executed to stop the failover. If you do not execute either `start-failover` or `stop-failover`, then the VIP associated with the deleted failover group will continue to be available. For more information about these commands, see the *Oracle Traffic Director Command-Line Reference.*

For more information about the commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

Note:

- If you want to assign a different node as the primary or backup node in a failover group, you should create the failover group afresh.
- There can be a maximum of 255 failover groups *across configurations*.

14.3 Configuring Health-Check Settings for Origin-Server Pools

To ensure that requests are distributed to only those origin servers that are available and can receive requests, Oracle Traffic Director monitors the availability and health of origin servers by sending health-check requests to all of the origin servers in a pool.

You can configure health-check parameters for an origin-server pool by using either the administration console or the CLI.

Note:

- When you configure health-check settings for an origin-server pool, you are, in effect, modifying a configuration. So for the updated configuration to take effect in the Oracle Traffic Director instances, you should redeploy the configuration as described in Section 4.3, "Deploying a Configuration."
- The CLI examples in this section are shown in shell mode (`tadm>`). For information about invoking the CLI shell, see Section 2.3.1, "Accessing the Command-Line Interface."

When Does Oracle Traffic Director Send Health-Check Requests?

When an Oracle Traffic Director instance starts, it performs an initial health check for all the origin servers in all of the configured origin-server pools.

If the initial health check indicates that an origin server is healthy, Oracle Traffic Director sends further health-check requests to an origin server only in the following situations:

- The server has not served any request successfully for the entire duration of the previous health-check interval.
- Dynamic discovery is enabled for this origin server pool. For more information, see Section 6.5, "Configuring an Oracle WebLogic Server Cluster as an Origin-Server Pool."

If a health check—either initial or subsequent—indicates that an origin server is not available, Oracle Traffic Director repeats the health check at the specified health-check interval.

Configurable Health-Check Settings

Table 14–1 lists the health-check settings that you can configure for each origin-server pool in a configuration.

Table 14–1 Health-Check Parameters

Parameter	Default Value
The type of connection—HTTP, TCP, or COMMAND—that Oracle Traffic Director should attempt with the origin server to determine its health. - TCP connection: Oracle Traffic Director attempts to open a TCP connection to each origin server. - HTTP request: Oracle Traffic Director sends an HTTP GET or OPTIONS request to each origin server in the pool, and checks the response to determine the availability *and* health of the origin server. **Note:** If you want to enable dynamic discovery of Oracle WebLogic Server managed servers in a cluster, then the health-check connection type must be set to HTTP. - COMMAND: Oracle Traffic Director uses an external executable created by the customer to monitor the health of specific origin servers. This mechanism is useful when you want to have a protocol-level health check monitor for the origin servers, which provide different services.	HTTP
The frequency at which health-check requests should be sent.	30 seconds
The duration after which a health-check request should be timed out if no response is received from the origin server.	5 seconds

Table 14–1 (Cont.) Health-Check Parameters

Parameter	Default Value
The number of times that Oracle Traffic Director should attempt to connect to an origin server in the pool, before marking it as unavailable.	5
The HTTP request method—GET or OPTIONS—that should be sent.	OPTIONS
The URI that should be sent for HTTP requests.	/
For external connections, the path to the external executable to run.	
The HTTP response codes that Oracle Traffic Director can accept as indicators of a healthy origin server. By default, Oracle Traffic Director accepts response codes from `1xx` to `4xx` as indicators of a healthy origin server.	
For HTTP GET health-check requests, a regular expression for the response body that Oracle Traffic Director can accept as the indicator of a healthy origin server	
For HTTP GET health-check requests, the maximum number of bytes in the response body that Oracle Traffic Director should consider when comparing the response body with the specified acceptable response body.	2048

When Is an Origin Server Considered Available and Healthy?

If the configured health-check connection type is TCP, an origin server is considered available if the connection is successfully established, indicating that the server is actively listening on its service port.

If the configured health-check connection type is HTTP, an origin server is considered available and health when all of the following conditions are fulfilled:

- There is no error while sending the HTTP request.
- The response is received before timeout period is reached.
- The status code in the response matches any of the acceptable response codes, if specified.

  By default, Oracle Traffic Director accepts response codes from `1xx` to `4xx` as indicators of a healthy origin server.
- The response body matches the acceptable response body, if specified.

Configuring Health-Check Settings for Origin Servers Using the Administration Console

To view and change health-check settings origin servers in a pool by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to view or change origin-server health-check settings.
4. In the navigation pane, expand **Origin-Server Pools**, and select the origin-server pool for which you want to view or change health-check settings.

The Origin-Server Pools page is displayed. It shows a list of the origin-server pools that are defined for the configuration.

5. Click the name of the origin-server pool that you want to modify.

   The Server Pool Settings page is displayed.

6. Go to the **Advanced Settings** section of the page.

7. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

8. After making the required changes, click **Save**.

   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Health-Check Settings for Origin Servers Using the CLI

- To view the current health-check settings for an origin-server pool in a configuration, run the `get-health-check-prop` command, as shown in the following example:

```
tadm> get-health-check-prop --config=soa --origin-server-pool=osp1
response-body-match-size=2048
protocol=HTTP
interval=30
request-method=OPTIONS
failover-threshold=3
request-uri=/
dynamic-server-discovery=false
timeout=5
```

- To change the health-check settings for an origin-server pool in a configuration, run the `set-health-check-prop` command.

  For example, the following command changes the health-check interval to 60 seconds and the health-check timeout period to 10 seconds for the origin-server pool `osp1` in the configuration `soa`.

```
tadm> set-health-check-prop --config=soa --origin-server-pool=osp1 interval=60
timeout=10
OTD-70201 Command 'set-health-check-prop' ran successfully.
```

  For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

14.3.1 Using an External Health-Check Executable to Check the Health of a Server

Oracle Traffic Director supports a generic health check hook-up mechanism, so that you can write your own health check programs/scripts to monitor the health of specific origin servers. An external executable is especially useful for a protocol-level health check monitor for the origin servers.

If you configure Oracle Traffic Director to use an external executable to check the health of a server, Oracle Traffic Director periodically invokes the executable and passes certain parameters to it as arguments and environment variables. If the executable successfully returns a status code 0 before a timeout, Oracle Traffic Director sets the server's status to online. If the executable returns a value other than zero or a timeout occurs before the execution ends, Oracle Traffic Director immediately sets the server status to offline without retrying, and terminates the execution in the timeout case. There are different reasons why the executable could return a non-zero status code, including a core dump, signal termination, or the logic of external executable itself. Oracle Traffic Director marks the server offline whenever the return status is non-zero.

Also, Oracle Traffic Director captures the standard output and standard error from the executable and logs the messages into the event log (server log).

The external executable handles the actual health check jobs, including establishing connection to the origin server, sending/receiving request/response, dealing with SSL (if applicable), retry logic (if required), and so on. The executable is expected to exit with a status 0 after it finishes the health check operation and wants to set the server status to online. If the executable wants to have some messages logged in the event log, it should print those messages to standard output.

14.3.1.1 Configuring Health-Check Settings to Use an External Executable

To configure the health-check settings to use an external executable for an origin-server pool in a configuration, run the `set-health-check-prop` command.

For example, the following command sets the health-check method to COMMAND, and specifies a path of `/path/myhcscript` for the external health-check executable. The interval, and timeout properties are also specified.

```
tadm> set-health-check-prop --user=admin --host=admin.example.com
--password-file=./admin.passwd --port=8989 --no-prompt --config=www.example.org
--origin-server-pool=test-pool protocol=COMMAND interval=25 timeout=4
command=/path/myhcscript
OTD-70201 Command 'set-health-check-prop' ran successfully.
```

Note: In case of an HTTP type of origin server pool, the `COMMAND` health check protocol is not considered if:

- the origin server type is `UNDETECTED` or,
- the origin server type is `WLS` and dynamic discovery is set.

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

14.3.1.2 Passing Parameters to the External Health Check Executable

Oracle Traffic Director passes parameters to the external health check executable in two ways. In particular, Oracle Traffic Director passes the origin server host, origin

server port, and timeout value via arguments, and passes all the existing environment variables as well as ORACLE_HOME, INSTANCE_HOME, INSTANCE_NAME, DOMAIN_HOME, and OTD_LOG_LEVEL as environment variables. The argument parameters are passed in the format of command line options, as shown in the following example command:

```
/path/myhcscript -h server1.myserver.com -p 389 -t 10
```

Where, -h, -p, and -t stand for host, port, and timeout respectively.

Table 14–2 Argument Parameters

Option	Meaning
-h	Origin server host.
-p	Origin server port.
-t	Health-check timeout.

You can pass other parameters to the external executable by specifying additional option arguments in the parameter command:

```
/path/myhcscript --secure -d /dbpath
```

Correspondingly, Oracle Traffic Director passes those additional arguments to the external executable:

```
/path/myhcscript --secure -d /dbpath -h server1.myserver.com -p 389 -t 10
```

Oracle Traffic Director does not automatically pass the origin server port type (for example, LDAP over SSL) to the executable. If the type information is needed in the executable, you can specify the type information in the command string as an additional argument (as shown in the example above) or have the type hard-coded or obtained from other resource (for example, its own configuration file or environment variable) in their health check program/script.

Furthermore, it is recommended that the external executable takes the timeout value into account and tries to complete execution and return status before timeout. If timeout occurs but execution is not complete, Oracle Traffic Director terminates the process and set the server status to offline.

14.3.1.3 Logging

Oracle Traffic Director passes the configured logging level to the external program via the environment variable OTD_LOG_LEVEL, and the value of the environment variable is an integer. In the external executable, you can customize the amount of logging messages based on the logging level. The following table defines the mapping between the Oracle Traffic Director logging levels and the argument values.

Table 14–3 Mapping Oracle Traffic Director Logging Levels and Argument Values

Value	Oracle Traffic Director Logging Level
0	NOTIFICATION:1 or higher
1	TRACE:1
2	TRACE:16
3	TRACE:32

Oracle Traffic Director logs contents in both standard output and the standard error of the external executable in a single log entry in the server log. If the exit status of the

command health check script is 0, the messages are logged at TRACE:1 level. Otherwise, standard output is logged at NOTIFICATION:1 level and the standard error is logged at WARNING:1 level.

15

Tuning Oracle Traffic Director for Performance

This chapter describes how you can use statistical data about Oracle Traffic Director instances and virtual servers to identify potential performance bottlenecks. It also describes configuration changes that you can make to improve Oracle Traffic Director performance.

This chapter contains the following sections:

15.1 General Tuning Guidelines

The outcome of the tuning suggestions provided in this chapter might vary depending on your specific environment. When deciding the tuning parameters that are suitable for your needs, keep the following guidelines in mind:

- **Adjust one parameter at a time**

  To the extent possible, make one adjustment at a time. Measure the performance before and after each change, and revert any change that does not result in measurable improvement.

- **Establish test cases that you can use to create a performance benchmark**

  Before changing any parameter, set up test cases, and automate them if possible, to test the effect of the changes on performance.

- **Tune gradually**

  When adjusting a quantitative parameter, make changes in small increments. This approach is most likely to help you identify the optimal setting quickly.

- **Start afresh after a hardware or software change**

At each major system change, a hardware or software upgrade, for example, verify whether the previous tuning changes still apply.

15.2 Tuning Connection Handling Settings

This section contains the following topics:

- Section 15.2.1, "Tuning the Thread Pool and Connection Queue"
- Section 15.2.2, "Tuning HTTP Listener Settings"
- Section 15.2.3, "Tuning Keep-Alive Settings"

15.2.1 Tuning the Thread Pool and Connection Queue

This section contains the following topics:

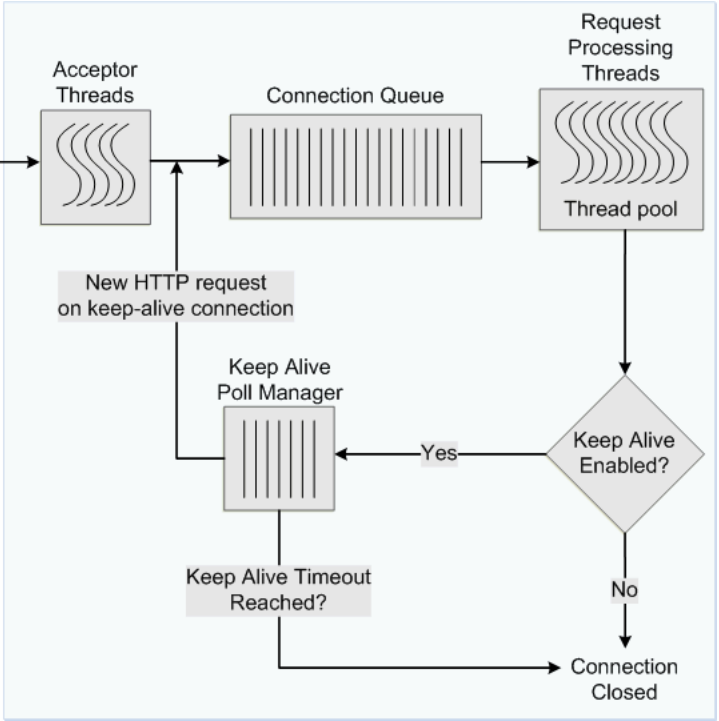
- Section 15.2.1.1, "Connection Handling Overview"
- Section 15.2.1.2, "Reviewing Connection Queue Metrics for an Instance"
- Section 15.2.1.3, "Reviewing Thread Pool Metrics for an Instance"
- Section 15.2.1.4, "Tuning the Thread Pool and Connection Queue Settings"

15.2.1.1 Connection Handling Overview

When a client sends a request to an HTTP listener in an Oracle Traffic Director instance, the connection is first accepted by an *acceptor thread* that is associated with the HTTP listener. The acceptor thread puts the connection in a *connection queue* and then waits for the next client request. A *request processing thread* from a *thread pool* takes the connection from the connection queue and processes the request. Note that if the thread pool is disabled, acceptor threads themselves process every request. The connection queue and request-processing threads do not exist.

Figure 15–1 depicts the connection handling process.

Figure 15–1 Connection Handling in Oracle Traffic Director



Connection handling in Oracle Traffic Director with keep-alive enabled, showing how a request is transmitted to a request processing thread and how connections are kept alive.

****

When an Oracle Traffic Director instance starts, it creates the specified number of acceptor threads for each listener and a thread pool that contains a specified, minimum number of request-processing threads.

- If the number of acceptor threads for a listener is not specified, Oracle Traffic Director creates one acceptor thread per CPU on the host.
- If the minimum size of the thread pool is not specified, Oracle Traffic Director creates one request-processing thread per processor on the host on which the instance is running.

As the request load increases, Oracle Traffic Director compares the number of requests in the connection queue with the number of request-processing threads. If the number of requests in the queue is more than the number of request-processing threads, Oracle Traffic Director creates additional threads, up to the specified maximum size for the thread pool.

The default value of the maximum number of request-processing threads will never be more than quarter of the maximum number of file descriptors available to the process. If there are 1, 2 CPUs, then the default is 256 and if there are 3, 4 CPUs, the default is 512. If there are more than 4 CPUs, the default is 1024.

The maximum number of threads is a hard limit for the number of sessions that can run simultaneously. Note that the maximum threads limit applies across all the virtual servers in the instance.

15.2.1.2 Reviewing Connection Queue Metrics for an Instance

If the maximum size of the connection queue is not large enough, client requests might be rejected during peak load periods. You can detect this situation by examining the connection queue section in the `perfdump` plain-text report, as shown in the following example.

```
ConnectionQueue:
----------------------------------------
Current/Peak/Limit Queue Length            0/1853/160032
Total Connections Queued                   11222922
Average Queue Length (1, 5, 15 minutes)    90.35, 89.64, 54.02
Average Queueing Delay                     4.80 milliseconds
```

- The `Current/Peak/Limit Queue Length` line indicates the following:
  - `Current`: The number of connections currently in the queue.
  - `Peak`: The largest number of connections that have been in the queue simultaneously.

    If the peak queue length is close to the limit, it is an indication that the connection queue might not be large enough for the given load.
  - `Limit`: The maximum size of the connection queue, which is equal to the size of the thread-pool queue + maximum threads + the size of the keep-alive queue.
- `Total Connections Queued` is the total number of times a connection has been queued. This number includes newly-accepted connections and connections from the keep-alive system.
- `Average Queue Length` is the average number of connections in the queue during the most recent 1-minute, 5-minute, and 15-minute intervals.
- `Average Queueing Delay` is the average amount of time a connection spends in the connection queue. It represents the delay between when a request is accepted by the server and when a request-processing thread begins processing the request. If the average queueing delay is relatively high in proportion to the the average response time, consider increasing the number of threads in the thread pool.

15.2.1.3 Reviewing Thread Pool Metrics for an Instance

You can review the thread-pool information for an instance in the `SessionCreationInfo` section of the plain-text `perfdump` report, as shown in the following example.

```
SessionCreationInfo:
------------------------
Active Sessions 2187
Keep-Alive Sessions 0
Total Sessions Created 4016/4016
```

- `Active Sessions` is the number of request-processing threads that are currently servicing requests.
- `Keep-Alive Sessions` shows the number of HTTP request processing threads serving keep-alive sessions.
- `Total Sessions Created`
  - The first number is the number of request-processing threads created.

- The second number is the maximum threads allowed in the thread pool; that is, the sum of the maximum threads configured in the thread-pool and the number of keep alive threads.

If you observe that the total number of request-processing threads created is consistently near the maximum number of threads, consider increasing the thread limit. Otherwise, requests might have to wait longer in the connection queue; and, if the connection queue becomes full, further requests are not accepted. If the average queueing delay (see Section 15.2.1.2) is significantly high in proportion to the average response time, that too is an indication that the thread limit needs to be increased.

15.2.1.4 Tuning the Thread Pool and Connection Queue Settings

You can change the thread pool and connection queue settings by using either the administration console or the CLI.

Changing the Thread Pool and Connection Queue Settings Using the Administration Console

To change the thread-pool settings by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration that you want to modify.
4. In the navigation pane, expand **Advanced Settings** and select **HTTP**.

   The HTTP Settings page is displayed.
5. Go to the **Thread Pool** section on the page.
6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Changing the Thread Pool and Connection Queue Settings Using the CLI

- To view the current thread-pool settings, run the `get-thread-pool-prop` command, as shown in the following example:

```
tadm> get-thread-pool-prop --config=soa
enabled=true
stack-size=262145
```

```
max-threads=20480
queue-size=2000
min-threads=20480
```

- To change the thread-pool settings, run the `set-thread-pool-prop` command.

  For example, to change the connection queue size, run the following command:

  ```
  tadm> set-thread-pool-prop --config=soa queue-size=2000
  OTD-70201 Command 'set-thread-pool-prop' ran successfully.
  ```

  For the updated configuration to take effect, deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

15.2.2 Tuning HTTP Listener Settings

The following are the key HTTP listener parameters that affect performance:

- **Listener address**

  The listener address consists of an IP address and a port number. The host on which an Oracle Traffic Director instance is running can have multiple network interfaces and multiple IP addresses.

  A listener that is configured to listen for client requests on all network interfaces on the host machine would have `0.0.0.0` as its IP address. While specifying `0.0.0.0` as the IP address for a listener is convenient, it results in one additional system call for each connection. For better performance, consider specifying an actual IP address for the listener.

- **Number of acceptor threads**

  Acceptor threads receive client requests and put them in the connection queue. When an Oracle Traffic Director instance starts, it creates the specified number of acceptor threads for each listener. If the number of acceptor threads for a listener is not specified, Oracle Traffic Director creates one acceptor thread per CPU on the host

  Too many idle acceptor threads place an unnecessary burden on the system, while having too few acceptor threads might result in client requests not being accepted. One acceptor thread per CPU, which is the default setting, is an acceptable trade-off in most situations.

  For HTTP 1.0 workloads, which necessitate opening and closing a relatively large number of connections, the default number of acceptor threads—1 per listener—would be suboptimal. Consider increasing the number of acceptor threads.

- **Listen queue size**

  As explained earlier, acceptor threads receive client requests and put them in the connection queue. If the operating system has not yet scheduled the acceptor thread, the operating system kernel maintains TCP connections on behalf of Oracle Traffic Director process. The kernel can accept connections up to the limit specified by the listen queue size.

  HTTP 1.0-style workloads can have many connections established and terminated. So if clients experience connection timeouts when an Oracle Traffic Director

instance is heavily loaded, you can increase the size of the HTTP listener backlog queue by setting the listen queue size to a larger value.

The plain-text `perfdump` report shows the IP address and the number of acceptor threads for each HTTP listener in the configuration, as shown in the following example:

```
ListenSocket ls1:
------------------------
Address                  https://0.0.0.0:1904
Acceptor Threads         1
Default Virtual Server   net-soa
```

You can change the HTTP listener settings by using either the administration console or the CLI, as described in Section 10.3, "Modifying a Listener."

15.2.3 Tuning Keep-Alive Settings

This section contains the following topics:

- Section 15.2.3.1, "About Keep-Alive Connections"
- Section 15.2.3.2, "Reviewing Keep-Alive Connection Settings and Metrics"
- Section 15.2.3.3, "Tuning Keep-Alive Settings"

15.2.3.1 About Keep-Alive Connections

HTTP 1.0 and HTTP 1.1 support sending multiple requests over a single HTTP connection. This capability, which was called *keep alive* in HTTP 1.0, is called *persistent connections* in HTTP 1.1 and is enabled by default in Oracle Traffic Director.

Keeping a connection active even after processing the original request helps reduce the time and overhead associated with creating and closing TCP connections for future similar requests. However, keep-alive connections over which few or no requests are received are an unnecessary burden on the system.

To avoid this problem, you can specify the maximum number of waiting keep-alive connections. When a keep-alive request is received, if there are more open connections waiting for requests than the specified maximum number, the oldest connection is closed. In addition, you can specify the period after which inactive keep-alive connections should be closed.

15.2.3.2 Reviewing Keep-Alive Connection Settings and Metrics

The plain-text `perfdump` report shows the current keep-alive settings and metrics, as shown in the following example:

```
KeepAliveInfo:
--------------------
KeepAliveCount 26/60000
KeepAliveHits 154574634
KeepAliveFlushes 0
KeepAliveRefusals 0
KeepAliveTimeouts 5921
KeepAliveTimeout 120 seconds
```

The `KeepAliveInfo` section of the `perdump` report shows the following:

- `KeepAliveCount`:
  - The first number is the number of connections in keep-alive mode.

  - The second number is the maximum number of keep-alive connections allowed.

- `KeepAliveHits` is the number of times a request was successfully received over a connection that was kept alive.

  If `KeepAliveHits` is high when compared with `KeepAliveFlushes`, it indicates that the keep-alive connections are being utilized well.

  If `KeepAliveHits` is low, it indicates that a large number of keep-alive connections remain idle, unnecessarily consuming system resources. To address this situation, you can do the following:

  - Decrease the maximum number of keep-alive connections so that fewer connections are kept alive.

    Note that the number of connections specified by the maximum connections setting is divided equally among the keep-alive threads. If the maximum connections setting is not equally divisible by the keep-alive threads setting, the server might allow slightly more than the maximum number of keep-alive connections.

  - Decrease the `KeepAliveTimeout` so that keep-alive connections do not remain idle for long. Note that if the `KeepAliveTimeout` is very low, the overhead of setting up new TCP connections increases.

- `KeepAliveFlushes` is the number of times the server closed connections that the client requested to be kept alive.

  To reduce keep-alive flushes, increase the keep-alive maximum connections.

  Caution: On UNIX/Linux systems, if the keep-alive maximum connections setting is too high, the server can run out of open file descriptors. Typically, 1024 is the limit for open files on UNIX/Linux; so increasing the keep-alive maximum connections above 500 is not recommended. Alternatively, you can increase the file descriptor limit, as described in Section 15.3, "Tuning the File Descriptor Limit."

- `KeepAliveRefusals` is the number of times the server could not hand off a connection to a keep-alive thread, possibly because the `KeepAliveCount` exceeded the keep-alive maximum connections. If this value is high, consider increasing the maximum number of keep-alive connections.

- `KeepAliveTimeouts` is the number of times idle keep-alive connections were closed because no requests were received over them during the last `KeepAliveTimeout` period.

- `KeepAliveTimeout` is the duration, in seconds, after which idle keep-alive connections are closed.

  Another parameter that is configurable and affects performance, but is not shown in the `perfdump` report is the keep-alive poll interval, which, together with `KeepAliveTimeout`, controls latency and throughput. Decreasing the poll interval and the timeout period reduces latency on lightly loaded systems. Increasing the values of these settings raises the aggregate throughput on heavily loaded systems. However, if there is too much latency and too few clients, the aggregate throughput suffers, because the server remains idle unnecessarily. Therefore, at a given load, if there is idle CPU time, decrease the poll interval; if there is no idle CPU time, increase the poll interval.

15.2.3.3 Tuning Keep-Alive Settings

You can tune the keep-alive settings by using either the administration console or the CLI.

Changing Keep-Alive Settings Using the Administration Console

To change the keep-alive settings by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration that you want to modify.
4. In the navigation pane, expand **Advanced Settings** and select **HTTP**.

   The HTTP Settings page is displayed.
5. Go to the **Keep Alive** section on the page.
6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Changing Keep-Alive Settings Using the CLI

- To view the current the keep-alive settings, run the `get-keep-alive-prop` command, as shown in the following example:

```
tadm> get-keep-alive-prop --config=soa
enabled=true
threads=20
max-connections=2000
poll-interval=0.002
timeout=31
```

- To change the keep-alive settings, run the `set-keep-alive-prop` command.

  For example to change the maximum number of keep-alive connections, run the following command:

```
tadm> set-keep-alive-prop --config=soa max-connections=2000
OTD-70201 Command 'set-keep-alive-prop' ran successfully.
```

  For the updated configuration to take effect, deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

15.3 Tuning the File Descriptor Limit

The operating system uses file descriptors to handle file-system files as well as pseudo files, such as connections and listener sockets.

When an Oracle Traffic Director instance starts, the following parameters are taken into consideration when auto-configuring values related to file descriptors:

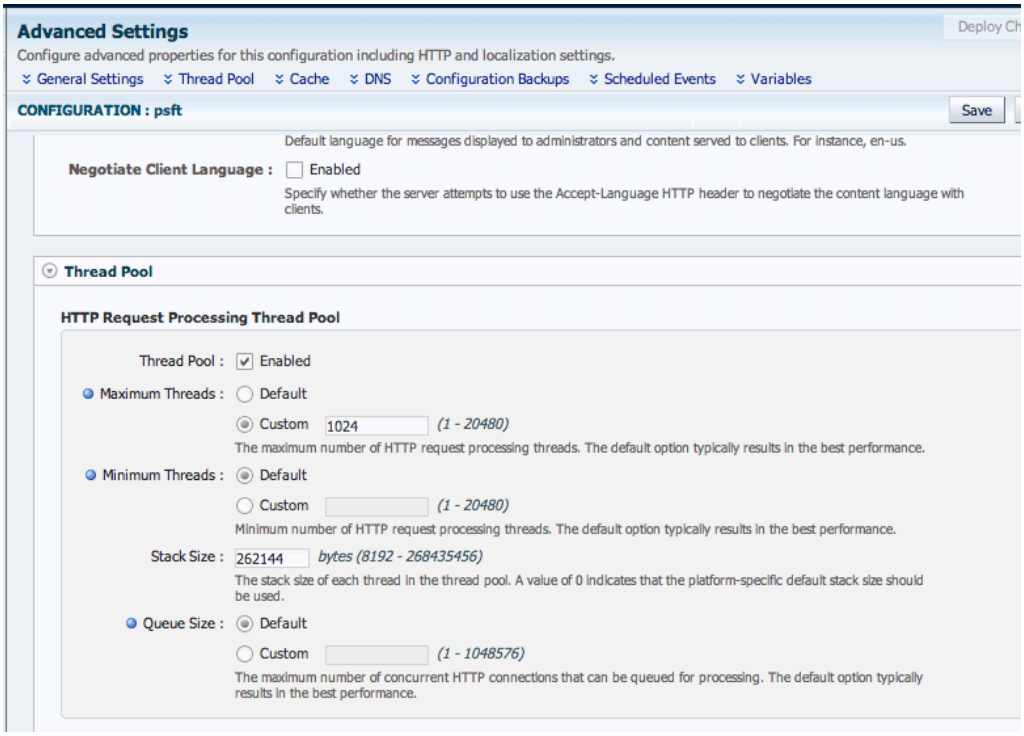
- HTTP processing threads (`<thread-pool>`)
- Access log counts for all virtual servers (`<access-log>`)
- Listeners (`<http-listener>, <tcp-listener>`)
- Keep-alive connections (`<keep-alive>`)
- Number of origin server pools (`<origin-server-pool>`)
- Number of origin servers (`<origin-server>`)
- Origin server connections (`<origin-server>/<max-connections>`)
- TCP processing threads (`<tcp-thread-pool>`)

The key Oracle Traffic Director objects that require file descriptors are keep-alive connections, queued connections, and connections to origin servers. If you do not explicitly specify limits for these objects, then when the Oracle Traffic Director instance starts, it configures the limits—maximum keep-alive connections, connection queue size, and maximum connections for each origin server—automatically based on the total number of available file descriptors in the system.

For instance, the maximum number of HTTP processing threads `* 4` should ideally be less than the maximum number of file descriptors available to the process. If you are going to increase the Maximum number of HTTP processing threads, then you should also correspondingly increase the total number of File Descriptor available to the Traffic Director. (For example, if the file descriptor limit is set to 65536, then setting the maximum number of HTTP processing threads to 20000 will cause sub-optimal tuning as 80000 (20000*4=80000) will exhaust/reserve file descriptors for the worker threads, which does not leave much for other subsystems). Hence you should set a high value for the maximum number of HTTP processing threads only after some experimentation.

Figure 15–1 depicts increasing the maximum number of HTTP processing threads in the advanced settings for the configuration.

Figure 15–2 Maximum Number of HTTP Processing Threads



Increasing the maximum number of HTTP processing threads in advanced settings.

****

The number of allocated file descriptors cannot exceed the limit that the system can support. To find out the current system limit for file descriptors, run the following command:

```
$ cat /proc/sys/fs/file-max
2048
```

To find out how many of the available file descriptors are being currently used, run the following command:

```
$ cat /proc/sys/fs/file-nr
```

The command returns an output that resembles the following:

```
625 52 2048
```

In this example, `625` is the number of allocated file descriptors, `52` is the number of free allocated file descriptors, and `2048` is the maximum number of file descriptors that the system supports.

Note: In Solaris, system wide file descriptors in use can be found by using the following command:

```
# echo ::kmastat | mdb -k | grep file_cache
```

This command returns an output that resembles the following:

```
file_cache        56    1154   1305      73728B   659529     0
```

In this example, `1154` is the number of file descriptors in use and `1305` the number of allocated file descriptors. Note that in Solaris, there is no maximum open file descriptors setting. They are allocated on demand as long as there is free RAM available.

When the number of allocated file descriptors reaches the limit for the system, the following error message is displayed in the system console when you try to open a file:

```
Too many open files in system.
```

The following message is written to the server log:

```
[ERROR:16] [OTD-10546] Insufficient file descriptors for optimum configuration.
```

This is a serious problem, indicating that the system is unable to open any more files. To avoid this problem, consider increasing the file descriptor limit to a reasonable number.

To change the number of file descriptors in Linux, do the following as the `root` user:

1. Edit the following line in the `/etc/sysctl.conf` file:

   ```
   fs.file-max = value
   ```

   `value` is the new file descriptor limit that you want to set.

2. Apply the change by running the following command:

   ```
   # /sbin/sysctl -p
   ```

Note: In Solaris, change the value of `rlim_fd_max` in the `/etc/system` file to specify the "hard" limit on file descriptors that a single process might have open. Overriding this limit requires superuser privilege. Similarly, `rlim_fd_cur` defines the "soft" limit on file descriptors that a single process can have open. A process might adjust its file descriptor limit to any value up to the "hard" limit defined by `rlim_fd_max` by using the `setrlimit()` call or by issuing the limit command in whatever shell it is running. You do not require superuser privilege to adjust the limit to any value less than or equal to the hard limit.

For example, to increase the hard limit, add the following command to /etc/system and reboot it once:

```
set rlim_fd_max = 65536
```

For more information about Solaris file descriptor settings, see Section 15.10.1, "Files Open in a Single Process (File Descriptor Limits)".

Most Operating Systems allow system administrators to configure a limit on the maximum number of file descriptors available to a process (such as Oracle Traffic Director). On Linux machines, this limit is typically set via `ulimit -n` – either at the time of logging into the system or simply while starting an application. System administrators will need to ensure that this limit (typically via `ulimit -n`) is set to least above 32276 file descriptors (or 65536 for high throughput systems) before starting an Oracle Traffic Director Server instance.

As a rough rule of thumb, the thread-pool element, max-threads * 4 should be less than the maximum number of file descriptors available to the process. That is, max-threads should be less than 1/5th of the maximum number of file descriptors.

For example, if the file descriptor limit is set to 65536, then setting max-threads to 20000 will cause sub-optimal tuning as 20000*4=80000 will exhaust/reserve file descriptors for the worker threads, leaving little else for other subsystems.

High values of max-threads should be used only after experimentation. Having tens of thousands of threads in a process may hurt performance.

15.4 Tuning HTTP Request and Response Limits

To optimize the time that an Oracle Traffic Director instance spends in processing requests and responses, you can configure parameters such as the size of request and response headers, the number of allowed header fields in a request, and the time that Oracle Traffic Director waits to receive an HTTP request body and header.

You can view the change the HTTP request and response limits by using either the administration console or the CLI.

Viewing and Changing HTTP Request/Response Limits Using the Administration Console

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

A list of the available configurations is displayed.

3. Select the configuration that you want to modify.
4. In the navigation pane, expand **Advanced Settings** and select **HTTP**.

   The HTTP Settings page is displayed.
5. Go to the **HTTP** section on the page.
6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Viewing and Changing HTTP Request/Response Limits Using the CLI

- To view the current settings, run the `get-keep-alive-prop` command, as shown in the following example:

```
tadm> get-http-prop --config=soa
request-header-timeout=30
request-body-timeout=-1
etag=true
io-timeout=30
max-request-headers=64
strict-request-headers=false
version=HTTP/1.1
discard-misquoted-cookies=true
ecid=true
favicon=true
unchunk-timeout=60
max-unchunk-size=8192
output-buffer-size=8192
request-header-buffer-size=8192
```

- To change the request and response limits, run the `set-http-prop` command.

  For example to change the response buffer size, run the following command:

```
tadm> set-http-prop --config=soa output-buffer-size=16384
OTD-70201 Command 'set-http-prop' ran successfully.
```

  For the updated configuration to take effect, deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

15.5 Tuning DNS Caching Settings

DNS caching helps reduce the number of DNS lookups that Oracle Traffic Director needs to perform to resolve client host names to IP addresses. The DNS cache is enabled by default in Oracle Traffic Director and stores IP address-to-DNS name mappings. Each entry in the DNS cache represents a single IP address or DNS name lookup. The DNS cache is used only when DNS lookup is enabled and when Oracle Traffic Director performs operations that require DNS lookup, such as recording client IP addresses and host names in the access log.

For the DNS cache hit rate to be high, the cache should be large enough to store the IP address-to-DNS name mappings for the maximum number of clients that you expect to access Oracle Traffic Director concurrently. You can tune the maximum number of entries allowed in the DNS cache and the cache expiry time. Note that setting the cache size too high might result in wasted memory.

This section contains the following topics:

- Section 15.5.1, "Viewing DNS Cache Settings and Metrics"
- Section 15.5.2, "Configuring DNS Cache Settings"

15.5.1 Viewing DNS Cache Settings and Metrics

Viewing DNS Cache Settings

To view the current DNS cache settings for a configuration, run the `get-dns-cache-prop` command, as shown in the following example:

```
tadm> get-dnscache-prop --config=soa
enabled=true
max-entries=1024
max-age=120
```

Viewing DNS Cache Metrics

You can view the current DNS cache utilization and hit rate in the plain-text `perfdump` report, as shown in the following example:

```
DNSCacheInfo:
------------------
enabled             yes
CacheEntries        0/1024
HitRatio            0/0 ( 0.00%)

Async DNS disabled
```

- The first line indicates whether the DNS cache is enabled.
- `CacheEntries` shows the number of entries currently in the DNS cache and the maximum number of entries allowed.
- `HitRatio` is the number of cache hits compared to the number of DNS cache lookups.
- The last line indicates whether asynchronous DNS lookup is enabled.

  You can configure Oracle Traffic Director to perform DNS lookups by using either its own asynchronous resolver or the operating system's synchronous resolver. DNS lookups performed by using the operating system's resolver are faster.

15.5.2 Configuring DNS Cache Settings

You configure the DNS cache settings for a configuration by using either the administration console or the CLI.

Configuring DNS Cache Settings Using the Administration Console

To configure DNS cache settings by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration that you want to modify.
4. In the navigation pane, select **Advanced Settings**.

   The Advanced Settings page is displayed.
5. Go to the **DNS** section on the page.
6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring DNS Cache Settings Using the CLI

To change the DNS cache settings for a configuration, run the `set-dns-cache-prop` command.

For example, the following command changes the maximum number of entries allowed in the DNS cache to 2048:

```
tadm> set-dns-cache-prop --config=soa max-entries=2048
OTD-70201 Command 'set-dns-cache-prop' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

15.6 Tuning SSL/TLS-Related Settings

This section contains the following topics:

- Section 15.6.1, "SSL/TLS Session Caching"
- Section 15.6.2, "Ciphers and Certificate Keys"

15.6.1 SSL/TLS Session Caching

During the initial SSL/TLS handshake process for an HTTPS connection, the client and server negotiate the cipher suites to be used, and the encryption/decryption and MAC keys (see "About SSL"). This activity requires significant CPU time, depending on whether RSA or ECC private keys are used, and the size of the keys.

The initial SSL/TLS handshake results in the generation of a unique SSL/TLS session ID. If the SSL/TLS session ID is cached, then the next time that same HTTPS client opens a new socket connection, the server can reduce the time taken to establish the connection by retrieving the SSL/TLS session ID from the cache and performing an abbreviated SSL/TLS handshake, which is less CPU-intensive than the initial handshake.

SSL/TLS session caching is enabled by default in Oracle Traffic Director. When a new connection is established on an SSL/TLS-enabled listener, Oracle Traffic Director checks whether the SSL/TLS session cache contains a session ID for the client. If the session ID for the client exists in the cache and is valid, Oracle Traffic Director allows the client to reuse the session.

You can configure the maximum number of entries in the SSL/TLS session cache and the duration for which SSL/TLS session IDs should be stored in the cache.

You can configure the SSL/TLS session cache settings for a configuration by using either the administration console or the CLI.

Configuring SSL/TLS Session Cache Settings Using the Administration Console

To configure SSL/TLS session cache settings by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration that you want to modify.
4. In the navigation pane, select **SSL**.

   The SSL Settings page is displayed.
5. Go to the **SSL & TLS** section on the page.
6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.

- In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring SSL/TLS Session Caching Settings Using the CLI

- To view the current SSL/TLS caching settings for a configuration, run the `get-ssl-session-cache-prop` command, as shown in the following example:

```
tadm> get-ssl-session-cache-prop --config=test
max-ssl3-tls-session-age=86400
enabled=true
max-entries=10000
```

- To change the SSL/TLS session caching settings, run the `set-ssl-session-cache-prop` command.

  For example, the following command changes the maximum number of entries allowed in the SSL/TLS session cache to 20000.

```
tadm> set-ssl-session-cache-prop --config=soa max-entries=20000
OTD-70201 Command 'set-ssl-session-cache-prop' ran successfully.
```

  For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

15.6.2 Ciphers and Certificate Keys

Strong ciphers and large private keys provide better security for SSL/TLS connections, but they affect performance.

- In SSL/TLS connections, certain ciphers—such as AES and RC4—require less computing resources for the data transfer than stronger ciphers such as 3DES. Consider this factor when you select SSL/TLS ciphers for listeners for which Strict SNI Host Matching is enabled.

  For information about configuring ciphers for listeners, see Section 11.2.4, "Configuring SSL/TLS Ciphers for a Listener."

  For information about SNI host matching, see Section 11.2.6, "About Strict SNI Host Matching."

- The initial SSL/TLS handshake process takes less time for RSA certificates with small key sizes—1024 and 2048 bits—than for certificates with large key sizes—3072 and 4096 bits.

  For information about creating self-signed certificates and certificate-signing requests, see Section 11.4, "Managing Certificates."

15.7 Configuring Access-Log Buffer Settings

The access log contains information about client requests to, and responses from, the server. When the rate at which an Oracle Traffic Director instance receives client requests is very high, which is usually the case in a production environment, the frequency of writing entries to the log file on the disk increases. Writing frequently to the disk is an I/O-intensive activity that can affect the performance of the server.

To reduce the frequency at which Oracle Traffic Director writes entries to the access log on the disk, access log updates can be buffered. Access-log buffering is enabled by default in Oracle Traffic Director.

You can specify limits for the access-log buffer size, the number of access-log buffers per server, and the maximum duration for which entries should be held in the buffer. When the buffer size, the number of buffers, or the age of an entry in the buffer reaches the specified limit, Oracle Traffic Director writes the buffered data to the access log on the disk.

You can configure the access-log buffer settings by using either the administration console or the CLI.

Configuring Access-Log Buffer Settings Using the Administration Console

To configure access-log buffer settings by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to configure access-log buffer preferences.
4. In the navigation pane, select **Logging**.

   The Log Preferences page is displayed.
5. Go to the Advanced Settings section on the page, and scroll down to the Access Log Buffer subsection.
6. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Access-Log Buffer Settings Using the CLI

- To view the current access-log buffer properties, run the `get-access-log-buffer-prop` command, as shown in the following example:

```
tadm> get-access-log-buffer-prop --config=soa
direct-io=false
enabled=true
max-buffers-per-file=default
buffer-size=8192
max-buffers=1000
```

```
max-age=1
```

To change the access-log buffer properties, run the `set-access-log-buffer-prop` command, as shown in the following example:

- To change the access-log buffer properties, run the `set-access-log-buffer-prop` command, as shown in the following example:

```
tadm> set-access-log-buffer-prop --config=soa
direct-io=false
enabled=true
max-buffers-per-file=default
buffer-size=8192
max-buffers=1000
max-age=1
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

For information about viewing logs, configuring log preferences, rotating logs, and so on, see Chapter 12, "Managing Logs."

15.8 Enabling and Configuring Content Compression

Compressed objects are delivered faster to clients, with fewer round-trips, reducing the overall latency without increasing the investment in expensive hardware.

You can create one or more compression rules specific to each Oracle Traffic Director virtual server, and configure the rules to be applicable either to all requests or to only those requests that match a specified condition.

Note: Certain files—such as GIF, JPEG, and PNG images; and zipped files—are either already compressed or cannot be compressed any further. Requiring Oracle Traffic Director to compress such files causes additional overhead without providing any compression benefit. Therefore, when creating compression rules for a virtual server, exclude such files.

For each compression rule, you can also specify the following parameters:

- Compression level, on the scale 1–9. At level 1, the compression time is the least; at level 9, the compression ratio is the best.

  At the higher compression levels, more CPU resources are consumed during the compression process, but relatively less network bandwidth is required to transmit the compressed content. On the other hand, compression at the lower levels is relatively less CPU-intensive, but more bandwidth is required to transmit the resulting content. So when choosing the compression level, consider which resource is more expensive in your environment—CPU resources or network bandwidth.

  – If CPU usage is more expensive, select a lower compression level.

  – If network bandwidth is the primary constraint, select a higher compression level.

- Number of bytes (fragment size) that should be compressed at a time.
- Whether the `Vary: Accept-Encoding` header should be included the response.

  The `Vary: Accept-Encoding` header instructs proxies situated between the client and Oracle Traffic Director that the compressed content should not be served to clients that cannot decompress the content.

Configuring Compression Rules Using the Administration Console

To create compression rules by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

   A list of the available configurations is displayed.
3. Select the configuration for which you want to create compression rules.
4. In the navigation pane, expand **Virtual Servers**, expand the name of the virtual server for which you want to create compression rules, and select **Compression**.

   The Compression Rules page is displayed. It lists the compression rules that are currently defined for the virtual server, and indicates whether the rules are enabled.

   Creating a Compression Rule

   a. Click **New Compression Rule**.

      The New Compression Rule dialog box is displayed.

      In the **Name** field, enter a name for the new compression rule.

      Select a compression level from the **Compression Level** drop-down list.

   b. Click **Next**.

      If this is the first compression rule for the virtual server, the New Caching Rule dialog box gives you the option to choose whether the rule should be applied to all requests. Select **All Requests**.

      If you wish to apply the rule to only those requests that satisfy a condition, create a new condition by selecting **Create a new condition**. In the New Expression pane, select a Variable/Function and an Operator from the respective drop-down lists and provide a value in the **Value** field.

      Select the `and`/or `operator` from the drop-down list when configuring multiple expressions. Similarly, use the `Not` operator when you want the route to be applied only when the given expression is not true.

      To enter a condition manually, click **Cancel** and then click **Edit Manually**. In the **Condition** field, specify the condition under which the rule should be applied. For information about building condition expressions, click the help button near the Condition field or see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

   c. Click **Next** and then click **Create Compression Rule**.

      The caching rule that you just created is displayed on the Compression Rules page.

In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Editing a Compression Rule

To enable or disable a compression rule, or to change the settings of a rule, do the following:

1. Click the **Name** of the compression rule that you want to change.

   The Edit Compression Rule dialog box is displayed.

   Note: To access the condition builder to edit conditions, select **Requests satisfying the condition** and click **Edit**. The condition builder enables you to delete old expressions and add new ones.

   ---

2. Specify the parameters that you want to change.

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.

3. After making the required changes, click **Save**.

   A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.

   In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Deleting a Compression Rule

To delete a compression rule, click the **Delete** button. At the confirmation prompt, click **OK**.

Configuring Compression Rules Using the CLI

- To create a compression rule for a virtual server, run the `create-compression-rule` command.

  For example, the following command creates a rule named `compress-docs` for the virtual server `soa.example.com` in the configuration `soa`, to cache the requests for which the expression `$uri='^/docs'` evaluates to true.

```
tadm> create-compression-rule --condition="\$uri='^/docs'" --config=soa
--vs=soa.example.com compress-docs
OTD-70201 Command 'create-compression-rule' ran successfully.
```

  Note that the value of the `--condition` option should be a regular expression. For information about building condition expressions, see "Using Variables, Expressions, and String Interpolation" in the *Oracle Traffic Director Configuration Files Reference*.

- To view a list of the compression rules defined for a virtual server, run the `list-compression-rules` command, as shown in the following example:

```
tadm> list-compression-rules --config=soa --vs=soa --verbose --all
rule            condition
------------------------
compress-docs   "$uri = '^/docs'
compress-all    -
```

- To view the current settings of a compression rule, run the `get-compression-rule-prop` command, as shown in the following example:

```
tadm> get-compression-rule-prop --config=soa --vs=soa --rule=compress-docs
fragment-size=8192
condition="$uri = '^/doc'"
compression-level=6
rule=compress-docs
insert-vary-header=true
```

- To change a compression rule, run the `set-compression-rule-prop` command.

  For example, the following command changes the compression level for the rule `compress-docs` to level 6.

```
tadm> set-compression-rule-prop --config=soa --vs=soa.example.com
--rule=compress-docs compression-level=9
OTD-70201 Command 'set-compression-rule-prop' ran successfully.
```

- To delete a compression rule, run the `delete-compression-rule` command, as shown in the following example.

```
tadm> delete-compression-rule --config=soa --vs=soa.example.com compress-docs
OTD-70201 Command 'delete-compression-rule' ran successfully.
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

15.9 Tuning Connections to Origin Servers

Each Oracle Traffic Director virtual server acts as a reverse proxy through which clients outside the network can access critical data and applications hosted on multiple origin servers in the back end. This section describes the parameters that you can tune to improve the performance of Oracle Traffic Director as a reverse-proxy server.

- **Enable keep-alive**: This parameter indicates whether the Oracle Traffic Director virtual server should attempt to use persistent connections to the origin server or create a new connection for each request. It is enabled by default.
- **Keep-alive timeout**: This parameter specifies the maximum duration, in seconds, for which a persistent connection can be kept open. The default timeout duration is 29 seconds.
- **Idle timeout**: This parameter specifies the maximum duration, in seconds, for which a connection to the origin server can remain idle. The default duration is 300 seconds.
- **Always use keep-alive**: This parameter indicates whether the Oracle Traffic Director virtual server can reuse existing persistent connections to origin servers

for all types of requests. If this parameter is not enabled (default), the Oracle Traffic Director virtual server attempts to use persistent connections to the origin server only for the GET, HEAD, and OPTIONS request methods.

- **Proxy buffer size**: This parameter specifies the size of the buffer in which Oracle Traffic Director stores data received from the origin server, before sending the data to the client. Larger the buffer, lower is the number of `write` system calls. The default size of the proxy buffer is 16 kilobytes.

The reverse-proxy settings for connections between an Oracle Traffic Director virtual server and an origin server pool are defined in routes. To change the reverse-proxy settings, you should edit the routes by using either the administration console or the CLI.

Note: In the current release, you cannot configure the proxy buffer size by using the administration console or the CLI.

To configure the proxy buffer size for a route, do the following:

1. Add the `proxy-buffer-size` parameter to the `http-client-config` server application function (SAF) in the *vs_name*`-obj.conf` configuration file of the virtual server that contains the route that you want to edit.

   The *vs_name*`-obj.conf` file is located in the following directory:

   ```
   INSTANCE_HOME/net-config_name/config
   ```

   The following is an example of a route (`route1`) for which the `proxy-buffer-size` and other reverse-proxy parameters have been configured.

   ```
   <Object name="route1">
   ObjectType fn="http-client-config" keep-alive-timeout="31"
   always-use-keep-alive="true" keep-alive="false" timeout="360"
   proxy-buffer-size="32768"
   Route fn="set-origin-server"
   origin-server-pool="origin-server-pool-1"
   </Object>
   ```

2. Save and close the *vs_name*`-obj.conf` file.
3. Run the `pull-config` command to update the configuration store on the administration server and to give effect to this change in all the instances of the configuration.

   ```
   tadm> pull-config --config=config_name node
   ```

   `node` is the name of the node on which you configured the proxy buffer size.

For more information about the `http-client-config` server application function (SAF), see the *Oracle Traffic Director Configuration Files Reference*.

Editing Routes Using the Administration Console

To edit routes by using the administration console, do the following:

1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
2. Click the **Configurations** button that is situated at the upper left corner of the page.

A list of the available configurations is displayed.

3. Select the configuration for which you want to edit routes.
4. In the navigation pane, expand **Virtual Servers**, expand the name of the virtual server for which you want to edit routes, and select **Routes**.

   The Routes page is displayed. It lists the routes that are currently defined for the virtual server.
5. Click the **Name** of the route that you want to edit.

   The Route Settings page is displayed.
6. Specify the reverse-proxy parameters in the following fields on the Route Settings page:

Section of the Route Settings Page	Field/s
General Settings	Keep Alive
	Keep Alive Timeout
Advanced Settings: Client Configuration for Connections with Origin Servers	Always Use Keep Alive
	Idle Timeout

   On-screen help and prompts are provided for all of the parameters.

   When you change the value in a field or tab out of a text field that you changed, the **Save** button near the upper right corner of the page is enabled.

   At any time, you can discard the changes by clicking the **Reset** button.
7. After making the required changes, click **Save**.
   - A message, confirming that the updated configuration was saved, is displayed in the Console Messages pane.
   - In addition, the **Deployment Pending** message is displayed at the top of the main pane. You can either deploy the updated configuration immediately by clicking **Deploy Changes**, or you can do so later after making further changes as described in Section 4.3, "Deploying a Configuration."

Configuring Routes Using the CLI

To change the properties of a route, run the `set-route-prop` command. The following are the names of the reverse-proxy parameters described earlier:

```
keep-alive-timeout
always-use-keep-alive
use-keep-alive
timeout
```

For example, the following command changes the keep-alive timeout duration for the route `route1` in the virtual server `soa.example.com` of the configuration `soa` to 30 seconds.

```
tadm> set-route-prop --config=soa --vs=soa.example.com --rule=route1
keep-alive-timeout=30
```

For the updated configuration to take effect, you should deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

15.10 Solaris-specific Tuning

This section provides tuning information that is specific to Solaris. Note that these are platform-specific tuning tips and any changes that you make could affect other process on the system.

15.10.1 Files Open in a Single Process (File Descriptor Limits)

Different platforms have different limits on the number of files that can be open in a single process at one time. For busy sites, increase that number. On Solaris systems, control this limit by setting `rlim_fd_max` and `rlim_fd_cur` in the `/etc/system` file. For Solaris 11, the default for `rlim_fd_max` is 65536 and the default value for `rlim_fd_cur` is 256.

After making this or any change in the `/etc/system` file, reboot Solaris for the new settings to take effect. In addition, if you upgrade to a new version of Solaris, remove any line added to `/etc/system` and add it again only after verifying that it is still valid.

An alternative way to make this change is by using the `ulimit -n <value>` command. Using this command does not require a system restart. However, this command only changes the login shell, whereas editing the `etc/system` file affects all shells.

15.10.2 Failure to Connect to HTTP Server

If clients experience connection timeouts when an Oracle Traffic Director instance is heavily loaded, you can increase the size of the HTTP listener backlog queue. To increase this setting, edit the HTTP listener's listen queue value.

In addition to this, you must also increase the limits within the Solaris TCP/IP networking code. There are two parameters that are changed by executing the following commands:

```
ipadm set-prop -p _conn_req_max_q=4096 tcp

ipadm set-prop -p _conn_req_max_q0=4096 tcp
```

These two settings increase the maximum number of two Solaris listen queues that can fill up with waiting connections. The setting `_conn_req_max_q` increases the number of completed connections waiting to return from an `accept()` call. The setting `_conn_req_max_q0` increases the maximum number of connections with the handshake incomplete. The default values for `_conn_req_max_q` and `_conn_req_max_q0` are 128 and 1024, respectively.

You can monitor the effect of these changes by using the `netstat -s` command and looking at the `tcpListenDrop`, `tcpListenDropQ0`, and `tcpHalfOpenDrop` values. Review them before adjusting these values. If the counters are not zero, adjust the value to 2048 initially, and continue monitoring the `netstat` output.

Do not accept more connections than Oracle Traffic Director is able to process. The value of 2048 for the parameters `tcpListenDrop`, `tcpListenDropQ0`, and `tcpHalfOpenDrop` typically reduces connection request failures, and improvement has been seen with values as high as 4096.

The HTTP listener's listen queue setting and the related Solaris `_conn_req_max_q` and `_conn_req_max_q0` settings are meant to match the throughput of Oracle Traffic

Director. These queues act as a buffer to manage the irregular rate of connections coming from web users. These queues allow Solaris to accept the connections and hold them until they are processed by Oracle Traffic Director.

15.10.3 Tuning TCP Buffering

TCP buffering can be tuned by using the `send_buf` and `recv_buf` parameters. For more information about these parameters, see Table 15–1, " Tuning Solaris for Performance Benchmarking".

15.10.4 Reduce File System Maintenance

UNIX file system (UFS) volumes maintain the time that each file was accessed. If the file access time updates are not important in your environment, you can turn them off by adding the `noatime` parameter to the data volume's mount point in `/etc/vfstab`. For example:

```
/dev/dsk/c0t5d0s6 /dev/rdsk/c0t5d0s6 /data0 ufs 1 yes noatime
```

Note: The `noatime` parameter does not turn off the access time updates when the file is modified, but only when the file is accessed.

For ZFS, you can use the `zfs set` command to modify any settable dataset property. The following example sets the `atime` property to `off` for `tank/home`.

```
zfs set atime=off tank/home
```

15.10.5 Long Service Times on Busy Volumes or Disks

An Oracle Traffic Director instance's responsiveness depends greatly on the performance of the disk subsystem. The `iostat` utility can be used to monitor how busy the disks are and how rapidly they complete I/O requests (the `%b` and `svc_t` `columns`, respectively). Service times are not important for disks that are less than 30% busy. However, for busier disks, service times should not exceed about 20 milliseconds. If busy disks have slower service times, improving disk performance can help performance substantially.If some disks are busy while others are lightly loaded, balance the load by moving some files from the busy disks to the idle disks.

15.10.6 Short-Term System Monitoring

Solaris offers several tools for keeping track of system behavior. Although you can capture their output in files for later analysis, the tools listed below are primarily meant for monitoring system behavior in real time:

- The `iostat -x 60` command reports disk performance statistics at 60-second intervals.

  To see how busy each disk is, take a look at the `%b` column. For any disk that is busy more than 20% of the time, pay attention to the service time as reported in the `svct` column. Other columns provide information about I/O operation rates, amount of data transferred, and so on.

- The `vmstat 60` command summarizes virtual memory activity and some CPU statistics at 60-second intervals.

  Take a look at the `sr` column to keep track of the page scan rate and take action if it is too high. In addition, monitor the `us`, `sy`, and `id` columns to see how heavily the

CPUs are being used. Note that you need to keep plenty of CPU power in reserve to handle sudden bursts of activity. Also keep track of the `r` column to see how many threads are competing for CPU time. If this remains higher than about four times the number of CPUs, reduce the server's concurrency.

- The `mpstat 60` command provides detailed view of the CPU statistics, while the `dlstat show-link -i 60` command summarizes network activity.

15.10.7 Long-Term System Monitoring

While it is important to monitor system performance with the tools mentioned above, collecting longer-term performance histories is equally important, as it can help you detect trends. For example, a baseline record of a system will help you find out what has changed if the system starts behaving poorly. Enable the system activity reporting package by doing the following:

- Run the following command:

  ```
  svcadm enable system/sar
  ```

- Run the command `crontab -e sys` and remove the `#` comment characters from the lines with the `sa1` and `sa2` commands. You can adjust how often the commands run and the time depending on your site's activity profile. For an explanation of the format of this file see the `crontab` man page.

  This command causes the system to store performance data in files in the `/var/adm/sa` directory, where they are retained for one month by default. You can then use the `sar` command to examine the statistics for time periods of interest.

15.10.8 Tuning for Performance Benchmarking

The following table shows the operating system tuning for Solaris used when benchmarking for performance and scalability. These values are an example of how you can tune your system to achieve the desired result.

Table 15–1 Tuning Solaris for Performance Benchmarking

Parameter	Scope	Default Value	Tuned Value	Comments
`rlim_fd_cur`	`/etc/system`	256	65536	Soft limit
`rlim_fd_max`	`/etc/system`	65536	65536	Process open file descriptors limit; accounts for the expected load (for the associated sockets, files, and pipes if any).
`_time_wait_interval`	`ipadm set-prop`	60000	600000	Set on clients as well.
`_conn_req_max_q`	`ipadm set-prop`	128	1024	
`_conn_req_max_q0`	`ipadm set-prop`	1024	4096	
`_ip_abort_interval`	`ipadm set-prop`	300000	600000	
`_keepalive_interval`	`ipadm set-prop`	7200000	9000000	For high traffic web sites, lower this value.

Table 15–1 (Cont.) Tuning Solaris for Performance Benchmarking

Parameter	Scope	Default Value	Tuned Value	Comments
`_rexmit_ interval_ initial`	`ipadm set-prop`	1000	3000	If re-transmission is greater than 30-40%, increase this value.
`_rexmit_ interval_max`	`ipadm set-prop`	60000	100000	
`_rexmit_ interval_min`	`ipadm set-prop`	200	3000	
`smallest_ anon_port`	`ipadm set-prop`	32768	65535	Set on clients as well.
`send_buf`	`ipadm set-prop`	49152	128000	To increase the transmit buffer.
`recv_buf`	`ipadm set-prop`	128000	1048576	To increase the receive buffer.

16

Diagnosing and Troubleshooting Problems

This chapter describes the methods and information sources you can use for diagnosing and solving problems that you might encounter while using Oracle Traffic Director.

This chapter contains the following sections:

- Roadmap for Troubleshooting Oracle Traffic Director
- Solutions to Common Errors
- Frequently Asked Questions
- Contacting Oracle for Support

16.1 Roadmap for Troubleshooting Oracle Traffic Director

This section provides the sequence of tasks you can perform to diagnose and solve problems with Oracle Traffic Director.

1. Verify whether the system configuration is correct.

   For information about the supported platforms and operating systems, see the Oracle Fusion Middleware Supported System Configurations at:

   `http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html`

2. Look for a solution to the problem in Section 16.2, "Solutions to Common Errors."
3. Check whether the information in Section 16.3, "Frequently Asked Questions" helps you understand or solve the problem.
4. Try to diagnose the problem.
   a. Review the messages logged in the server log. Look for messages of type `WARNING`, `ERROR`, and `INCIDENT_ERROR`.

      For messages of type `WARNING` and `ERROR`, try to solve the problem by following the directions, if any, in the error message.

      An `INCIDENT_ERROR` message indicates a serious problem caused by unknown reasons. You should contact Oracle for support.

   b. Increase the verbosity of the server log, and try to reproduce the problem.

      Oracle Traffic Director supports several log levels for the server log, as described in Table 12–1, " Server Log Levels". The default log level is `NOTIFICATION:1`. The least verbose log level is `INCIDENT_ERROR`, at which only serious error messages are logged. At the `TRACE:1`, `TRACE:16`, or `TRACE:32`

levels, the logs are increasingly verbose, but provide more detailed information, which can be useful for diagnosing problems.

Increase the log verbosity and then try to reproduce the problem. When the problem occurs again, review the messages logs for pointers to the cause of the problem.

For information about changing the server log level, see Section 12.3, "Configuring Log Preferences."

c. Restore the instances to a previous configuration.

When you redeploy a modified configuration to its instances, a backup of the previous configuration is stored in a zip file on the administration server.

Restore the instances to an appropriate previous configuration as described in Section 4.9, "Restoring a Configuration from a Backup," and check whether the problem persists.

If the problem does not occur with the previous configuration, it is clear that the problem is caused by a change made in the current configuration.

5. Contact Oracle for support, as described in Section 16.4, "Contacting Oracle for Support."

16.1.1 Troubleshooting High Availability Configuration Issues

This section provides information about the tasks you can perform to diagnose and solve problems with an Oracle Traffic Director high availability configuration.

- Ensure that the interface being used is EoIB (Ethernet over InfiniBand).
- The Oracle Traffic Director configuration must be deployed on two nodes. For more information, see Section 14.2, "Creating and Managing Failover Groups."
- The router ID for each failover group has to be unique.
- Make sure that KeepAlived is installed. In most cases KeepAlived software is installed by default on both the Exalogic compute nodes (or VMs) where Oracle Traffic Director instances are running. To check if KeepAlived is installed, run the following command:

  ```
  rpm -qa | grep keepalived
  ```

  If KeepAlived is correctly installed, an output similar to the following is displayed:

  ```
  keepalived-1.2.2-1.el5
  ```

  Note that if KeepAlived is not installed, the RPM can be found in the Exalogic software repository. For more information about KeepAlived, see Section 14.2.1, "How Failover Works."
- For KeepAlived specific information, check the logs in the `/var/log/messages` directory.
- Make sure to provide the correct VIP address and the appropriate subnet mask (netmask) bit-size for successfully completing the high availability configuration. In addition, ensure that you provide the netmask bits and not the actual netmask value. For more information, see Section 14.2.3, "Creating Failover Groups."
- If you are using Oracle Traffic Director 11.1.1.6.0, note that it does not support high availability with more than one Oracle Traffic Director configuration on the same pair of administration nodes.

16.2 Solutions to Common Errors

This section provides solutions to the following problems:

- Section 16.2.1, "Startup failure: could not bind to port"
- Section 16.2.2, "Unable to start server with HTTP listener port 80"
- Section 16.2.3, "Unable to restart SSL/TLS-enabled server after changing the PKCS#11 token pin"
- Section 16.2.4, "Unable to start the SNMP subagent"
- Section 16.2.5, "Unable to communicate with the administration server: connection refused"
- Section 16.2.6, "Oracle Traffic Director consumes excessive memory at startup"
- Section 16.2.7, "Operating system error: Too many open files in system"
- Section 16.2.8, "Unable to stop instance after changing the temporary directory"
- Section 16.2.9, "Unable to restart the administration server"
- Section 16.2.10, "Oracle Traffic Director does not maintain session stickiness"

16.2.1 Startup failure: could not bind to port

This error occurs when one or more HTTP listeners in the configuration are assigned to a TCP port number that is already in use by another process.

```
[ERROR:32] startup failure: could not bind to port port (Address already in use)
[ERROR:32] [OTD-10380] http-listener-1: http://host:port: Error creating socket
(Address already in use)
[ERROR:32] [OTD-10376] 1 listen sockets could not be created
[ERROR:32] server initialization failed
```

You can find out the process that is listening on a given port by running the following command:

```
> netstat -npl | grep :port | grep LISTEN
```

If the configured HTTP listener port is being used by another process, then either free the port or change it as described in Section 10.3, "Modifying a Listener."

16.2.2 Unable to start server with HTTP listener port 80

This error occurs if you configure an HTTP listener port up to `1024` (say `80`) and attempt to start the Oracle Traffic Director instance as a non-`root` user.

The following messages are written to the server log:

```
[ERROR:32] [OTD-10376] 1 listen sockets could not be created
[ERROR:32] [OTD-10380] http-listener-1: http://soa.example.com:80:
 Error creating socket (No access rights)
```

Port numbers up to `1024` are assigned by the Internet Assigned Numbers Authority (IANA) to various services. These port numbers are accessible only by the `root` user.

To solve this problem, you can do one of the following:

- Configure the Oracle Traffic Director listener with a port number higher than `1024` (say, `8080`), and create an IP packet-filtering rule to internally redirect requests received at port `80` to the configured Oracle Traffic Director port, as shown in the following examples:

```
# /sbin/iptables -t nat -A PREROUTING -p tcp -m tcp --dport 80 -j REDIRECT
--to-ports 8080
# /sbin/iptables -t nat -A PREROUTING -p udp -m udp --dport 80 -j REDIRECT
--to-ports 8080
```

Make sure that the `iptables` service is started by default when the server restarts by running the `chkconfig` command, as shown in the following example:

```
# chkconfig --level 35 iptables on
```

- If `xinetd` is installed in the system, create a file (named `otd`, for example) in the `/etc/xinetd.d/` directory with the following entry:

```
service otd
{
type = UNLISTED
disable = no
socket_type = stream
protocol = tcp
user = root
wait = no
port = 80
redirect = 127.0.0.1 8080
}
```

This entry redirects all incoming TCP traffic received on port `80` to port `8080` on the local machine.

For more information, see the Linux `xinetd` documentation.

16.2.3 Unable to restart SSL/TLS-enabled server after changing the PKCS#11 token pin

This error occurs when, for an SSL-enabled configuration, you set or change the PKCS#11 token pin, and then deploy the updated configuration while the instances are running.

The following messages are written to the server log:

```
[ERROR:32] [OTD-10094] NSS PKCS #11 initialization failed
(SEC_ERROR_BAD_PASSWORD: Security password entered is incorrect.)
[ERROR:32] [OTD-10492] New configuration not installed
[ERROR:32] [OTD-10520] The new configuration is incompatible with the existing
 configuration (Enabling PKCS #11 or SSL requires a server restart)
```

To solve this problem, start the instance by using the `start-instance` CLI command or by clicking the **Start/Restart Instances** button in the administration console. At the resulting prompt, enter the pin for each token that is protected with a pin.

To avoid this error, after you set or change the PKCS#11 token pin for an SSL-enabled configuration, first stop the running instances, deploy the changes, and then start the instances.

16.2.4 Unable to start the SNMP subagent

This error usually occurs when the configured SNMP subagent port is being used by another process.

The following message is written to the administration server log.

```
OTD-63410 The SNMP subagent failed to start.
```

Check whether the configured port for the SNMP subagent on the node is already used by another process, by using the following command.

```
> netstat -npl --udp | grep :port
```

To solve this problem, either free the port or change it in the *INSTANCE_HOME*/admin-server/config/snmpagt.conf file, as described in Section 13.6.2, "Configuring the SNMP Subagent."

16.2.5 Unable to communicate with the administration server: connection refused

This error occurs when you run the `tadm` command in the following situations:

- The value specified for the `--port` option is not correct.
- The `--port` option was not specified, and the administration server is running on a port other than the default SSL port `8989`.

Run the command again with the correct value for the `--port` option.

16.2.6 Oracle Traffic Director consumes excessive memory at startup

When you start an Oracle Traffic Director instance, the values for certain parameters—maximum number of keep-alive connections, size of the connection queue, and maximum number of connections to origin servers—are assigned automatically based on the system's file descriptor limit.

If the file descriptor limit is very high, the auto-assigned values for undefined parameters can be needlessly high, causing Oracle Traffic Director to consume an excessive amount of memory. To avoid this problem, explicitly configure the maximum number of keep-alive connections (Section 15.2.3.3), the size of the connection queue (Section 15.2.1.4), and the maximum number of connections to individual origin servers (Section 7.3).

16.2.7 Operating system error: Too many open files in system

This operating system error occurs in Linux when the number of allocated file descriptors reaches the limit for the system.

The following message is written to the server log:

```
[ERROR:16] [OTD-10546] Insufficient file descriptors for optimum configuration.
```

To avoid this error, increase the file descriptor limit on Linux from the default of 1024 to a reasonable number. For more information, see Section 15.3, "Tuning the File Descriptor Limit."

16.2.8 Unable to stop instance after changing the temporary directory

This error occurs when, after changing the temporary directory for a configuration, you deploy the change without stopping the instances, and then attempt to stop the instances later. The temporary directory is the directory (on the administration node) in which the process ID and socket information for the instances of the configuration are stored.

When this error occurs, the following message is written to the server log:

```
OTD-63585 An error occurred while stopping the server. For details, see the server log.
```

To Avoid This Error

If you change the temporary directory for a configuration, you should first stop all the instances of the configuration, deploy the changes, and then start the instances.

To Solve This Problem

Kill the Oracle Traffic Director instance.

1. Find out the current temporary directory for the configuration by doing one of the following:
   - Run the `get-config-prop` CLI command, as shown in the following example:

     ```
     tadm> get-config-prop --config=soa temp-path
     /tmp/net-test-a46e5844
     ```

   - Log in to the administration console, select the required configuration, and select **Advanced Settings**. On the resulting page, look for the Temporary Directory field.

   Note the path to the temporary directory.
2. Find out the process ID of the running instance by running the following command:

   ```
   cat temp_dir/pid
   ```

   `temp_dir` is the full path to the temporary directory that you noted in step 1.

   Note the process ID that this command returns.
3. Kill the process, by running the following command:

   ```
   kill pid
   ```

   `pid` is the process ID that you noted in step 2.

16.2.9 Unable to restart the administration server

In Linux systems, the cron script tmpwatch, located at `/etc/cron.daily/tmpwatch`, is set to execute everyday by default. This script removes all files that are older than 240 hours (10 days) from all `/tmp` directories in the administration server. Hence, if the administration server is not restarted for more than 10 days, the default pid file is removed. This in turn prevents the administration server from being restarted after 10 days.

To Avoid This Problem

- **Change `temp-path` location**: In the file, `<otd-home>/admin-server/config/server.xml`, change the `temp-path` value to a location where the server user has exclusive rights. For example, change it to, `<temp-path>/var/tmp/https-test-1234</temp-path>`. In addition, make sure that the new `temp-path` is not being monitored by the tmpwatch script.
- **Change the cron script**: Remove the value `240 /tmp` from the cron script for tmpwatch. Use the `-X/--exclude-pattern` option to exclude a directory from being monitored by tmpwatch. For more information about using this option, see the man-page for tmpwatch.

16.2.10 Oracle Traffic Director does not maintain session stickiness

Oracle Traffic Director can maintain session stickiness as follows:

Cookie Based Session Persistence

This is a common scenario where clients accept cookies from web or application servers. In this scenario, Oracle Traffic Director, while load balancing HTTP traffic, ensures session persistence using its own cookie. This ensures that sticky requests, requests containing HTTP Session cookie, are routed to the same back-end application server where this session cookie originated.

Oracle Traffic Director 11.1.1.5 needs to be explicitly configured to honor session persistence when a back-end application server uses HTTP Session cookie other than the default `JSESSIONID`. On the other hand, Oracle Traffic Director 11.1.1.6 honors session persistence on receiving any cookie from the origin server.

Note: Oracle Traffic Director needs additional patches within WebLogic 10.3.x to maintain URI based session stickiness.

URI Based Session Persistence

This is not a very common scenario. In this case, cookies are disabled on clients and back-end web or application servers maintain session persistence by appending HTTP session information to the URI.

In this scenario, Oracle Traffic Director can honor session persistence if the back-end application server appends Oracle Traffic Director's `JRoute` cookie to the URI. Origin servers like WebLogic Server 10.3.6.2 and higher, 12.1 and higher, and GlassFish 2.0 and higher have the ability to append this `JRoute` cookie to the URI. Hence, Oracle Traffic Director is able to maintain URI based session persistence only with these origin servers.

16.3 Frequently Asked Questions

This section contains the following subsections:

16.3.1 How do I reset the password for the administration server user?

Run the `reset-admin-password` CLI command as described in the "Changing the Administrator User Name and Password Using the CLI" section.

16.3.2 What is a "configuration"?

A configuration, in Oracle Traffic Director terminology, is a collection of configurable elements (metadata) that determine the run-time behavior of an Oracle Traffic Director instance.

For more information, see Section 1.4, "Oracle Traffic Director Terminology."

16.3.3 How do I access the administration console?

See Section 2.3.2, "Accessing the Administration Console."

16.3.4 Why do I see a certificate warning when I access the administration console for the first time?

The browser displays a warning because the administration server has a self-signed certificate. To proceed, you should choose to trust the certificate.

16.3.5 Can I manually edit configuration files?

The files in the configuration store are updated automatically when you edit a configuration by using either the administration console or the CLI. Unless otherwise instructed in the Oracle Traffic Director documentation, DO NOT edit the files in the configuration store manually.

For the configuration changes to take effect, you should deploy the configuration to the instances as described in Section 4.3, "Deploying a Configuration."

If you inadvertently edit a configuration file of an instance, and if you want to retain the change and replicate it in all the instances of the configuration, you can *pull* the configuration from the instance to the administration server, as described Section 4.5, "Synchronizing Configurations Between the Administration Server and Nodes."

16.3.6 In the administration console, what is the difference between saving a configuration and deploying it?

When you save a configuration, the changes you made are saved in the configuration store on the administration server. For the changes to take effect in the instances of the configuration, you must deploy the configuration as described in Section 4.3, "Deploying a Configuration."

16.3.7 Why is the "Deployment Pending" message displayed in the administration console?

The **Deployment Pending** message is displayed in the administration console when you change a configuration and save it on the administration server. It indicates that the changes are yet to be copied over to the instances of the configuration.

If you have finished making the required configuration changes, you can deploy the changes to all of the instances by clicking **Deploy Changes** in the administration console or by running the `deploy-config` CLI command, as described in Section 4.3, "Deploying a Configuration."

16.3.8 Why is the "Instance Configuration Deployed" message is displayed in the administration console?

The **Instance Configuration Deployed** message is displayed in the administration console when you manually edit the configuration files of an instance. It indicates that the configuration files of one or more instances are different from the corresponding configuration files stored in the configuration store on the administration server. For information about what you should do in this situation, see Section 4.5, "Synchronizing Configurations Between the Administration Server and Nodes."

16.3.9 Why does the administration console session end abruptly?

If an administration console session remains inactive for 30 minutes, it ends automatically. You should log in again.

16.3.10 How do I access the CLI?

See Section 2.3.1, "Accessing the Command-Line Interface."

16.3.11 Why does "tadm --user=admin --host=myhost subcommand" take me into a command shell instead of executing the specified subcommand?

For the specified CLI subcommand to be executed, it must precede all the options, including the common options like `--user` and `--host`. Otherwise, the CLI assumes that you attempting to invoke the shell mode.

16.3.12 Why is a certificate warning message displayed when I tried to access the CLI for the first time?

The CLI connects to the SSL port of the administration server. The administration server has a self-signed certificate. The message that you see when you connect to the administration server for the first time is a prompt to choose whether you trust the certificate. Make sure that you are connecting to the correct server and port, and enter `y` to trust the certificate. For subsequent invocations of the CLI, the warning message is not displayed.

16.3.13 How do I find out the short names for the options of a CLI command?

See help for the command, by running the command with the `--help` option.

16.3.14 Can I configure the CLI to not prompt for a password every time I access it?

Yes. Store the password in a file in the format, `tadm_password=`*`password`*. When you run a CLI command, specify the full path to the password file by using the --password-file option.

16.3.15 Why am I unable to select TCP as the health-check protocol when dynamic discovery is enabled?

When dynamic discovery is enabled, Oracle Traffic Director needs to send, at a specified interval, an HTTP request containing specific headers to determine whether the origin servers specified in the pool are Oracle WebLogic Server instances and whether the servers belong to a cluster. The response to a TCP health-check request would not provide the necessary information to determine the presence of Oracle WebLogic Server instances. So when dynamic discovery is enabled, the health-check protocol must be set to HTTP.

16.3.16 After I changed the origin servers in a pool to Oracle WebLogic Servers, they are not discovered automatically, though dynamic discovery is enabled. Why?

If dynamic discovery is enabled, when the Oracle Traffic Director instance starts, it determines whether or not the configured origin server is an Oracle WebLogic Server instance.

So if you initially configured, say, an Oracle GlassFish Server instance as the origin server, then at startup, Oracle Traffic Director determines that the origin server is not an Oracle WebLogic Server instance. Subsequently, if you replace the origin server with an Oracle WebLogic Server instance, then for Oracle Traffic Director to determine afresh that the origin server is now an Oracle WebLogic Server instance, you must either restart the Oracle Traffic Director instances or reconfigure them.

If you want to change the origin servers from Oracle WebLogic Server instances to other servers, or vice versa, without restarting the instances, do the following:

1. Create a new origin-server pool with the required origin servers, and delete the old pool. For more information, see Chapter 6, "Managing Origin-Server Pools."
2. Update the appropriate routes to point to the new pool, as described in Section 8.4, "Configuring Routes."
3. Reconfigure the Oracle Traffic Director instances by using the `reconfig-instance` CLI command, as described in Section 5.4, "Updating Oracle Traffic Director Instances Without Restarting."

16.3.17 How do I view the request and response headers sent and received by Oracle Traffic Director?

You can enable logging of the request and response headers in the server log by modifying the appropriate route, using either the administration console or the CLI.

- **Using the administration console**
  1. Log in to the administration console, as described in Section 2.3.2, "Accessing the Administration Console."
  2. Click the **Configurations** button that is situated at the upper left corner of the page.

     A list of the available configurations is displayed.
  3. Select the configuration for which you want to configure routes.
  4. In the navigation pane, expand **Virtual Servers**, expand the name of the virtual server for which you want to configure routes, and select **Routes**.

     The Routes page is displayed. It lists the routes that are currently defined for the selected virtual server.
  5. Click the **Name** of the route that you want to configure.

     The Route Settings page is displayed.
  6. Go to the **Advanced Settings** section of the Route Settings page, and scroll down to the **Client Configuration for Connections with Origin Servers** subsection.
  7. Select the **Log Headers** check box.
  8. Click **Save**.
  9. Click **Deploy Changes**.
- **Using the CLI**

  Run the `set-route-prop` command, as shown in the following example:

  ```
  tadm> set-route-prop --config=soa --vs=soa.example.com --route=default-route
  log-headers=true
  ```

  This command enables logging of the headers that Oracle Traffic Director sends to, and receives from, the origin servers associated with the route named `default-route` in the virtual server `soa.example.com` of the configuration `soa`.

  For the updated configuration to take effect, deploy it to the Oracle Traffic Director instances by using the `deploy-config` command.

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the commands with the `--help` option.

The headers are logged in the server log as shown in the following example:

```
[2011-11-11T03:45:00.000-08:00] [net-test] [NOTIFICATION] [OTD-11008] []
 [pid: 8184] for host 10.177.243.152 trying to OPTIONS / while trying to GET
 /favicon.ico, service-http reports: request headers sent to origin
server(soa.example.com:1900) :[[
OPTIONS / HTTP/1.1
Proxy-agent: Oracle-Traffic-Director/11.1.1.7
Surrogate-capability: otd="Surrogate/1.0"
Host: dadvma0178:1900
Proxy-ping: true
X-weblogic-force-jvmid: unset
Via: 1.1 net-test
Connection: keep-alive
]]
[2011-11-11T03:45:00.000-08:00] [net-test] [NOTIFICATION] [OTD-11009] []
 [pid: 8184] for host 10.177.243.152 trying to OPTIONS / while trying to GET
 /favicon.ico, service-http reports: response headers received from origin
server(soa.example.com:1900) :[[
HTTP/1.1 200 OK
date: Fri, 11 Nov 2011 11:45:00 GMT
server: Apache/2.2.17 (Unix)
allow: GET,HEAD,POST,OPTIONS,TRACE
content-length: 0
keep-alive: timeout=5, max=100
connection: Keep-Alive
content-type: text/html]
```

16.3.18 How do I enable SSL/TLS for an Oracle Traffic Director instance?

See Section 11.2, "Configuring SSL/TLS Between Oracle Traffic Director and Clients."

16.3.19 How do I find out which SSL/TLS cipher suites are supported and enabled?

See Section 11.2.4, "Configuring SSL/TLS Ciphers for a Listener."

16.3.20 How do I view a list of installed certificates?

See Section 11.4.4, "Viewing a List of Certificates."

16.3.21 How do I issue test requests to an SSL/TLS-enabled Oracle Traffic Director instance?

Run the following command:

```
$ openssl s_client -host hostname -port portnumber -quiet
```

- If you omit the `-quiet` option, information about the SSL/TLS connection—such as the server DN, certificate name, and the negotiated cipher suite—is displayed.
- For testing with a specific cipher, specify the `-cipher` option.

After the SSL/TLS connection is established, enter an HTTP request, as shown in the following example.

```
GET /
```

For more information, see the `s_client` man page.

16.3.22 How do I analyze SSL/TLS connections?

Several tools are available to observe request and response data over SSL/TLS connections. One such tool is `ssltap`, which serves as a simple proxy between the client and the Oracle Traffic Director and displays information about the connections that it forwards.

Run the following command:

```
$ ssltap -l -s otd_host:otd_port
```

For example, to observe the communication between clients and the SSL/TLS-enabled Oracle Traffic Director listener `soa.example.com:1905`, run the following command:

```
$ ssltap -l -s soa.example.com:8080
```

The following messages are displayed:

```
Looking up "localhost"...
Proxy socket ready and listening
```

By default, `ssltap` listens on port 1924. Connect to `https://localhost:1924` by using your browser.

You will see an output similar to the following:

```
Connection #1 [Tue Oct 25 04:29:46 2011]
Connected to localhost:8080
--> [
(177 bytes of 172)
SSLRecord { [Tue Oct 25 04:29:46 2011]
   type    = 22 (handshake)
   version = { 3,1 }
   length  = 172 (0xac)
   handshake {
      type = 1 (client_hello)
      length = 168 (0x0000a8)
         ClientHelloV3 {
            client_version = {3, 1}
            random = {...}
            session ID = {
                length = 0
                contents = {...}
            }
            cipher_suites[29] = {
                (0x00ff) TLS_EMPTY_RENEGOTIATION_INFO_SCSV
                (0xc00a) TLS/ECDHE-ECDSA/AES256-CBC/SHA
                (0xc014) TLS/ECDHE-RSA/AES256-CBC/SHA
                (0x0039) TLS/DHE-RSA/AES256-CBC/SHA
                (0x0038) TLS/DHE-DSS/AES256-CBC/SHA
                (0xc00f) TLS/ECDH-RSA/AES256-CBC/SHA
                (0xc005) TLS/ECDH-ECDSA/AES256-CBC/SHA
                (0x0035) TLS/RSA/AES256-CBC/SHA
                (0xc007) TLS/ECDHE-ECDSA/RC4-128/SHA
                (0xc009) TLS/ECDHE-ECDSA/AES128-CBC/SHA
                (0xc011) TLS/ECDHE-RSA/RC4-128/SHA
                (0xc013) TLS/ECDHE-RSA/AES128-CBC/SHA
                (0x0033) TLS/DHE-RSA/AES128-CBC/SHA
                (0x0032) TLS/DHE-DSS/AES128-CBC/SHA
```

```
                (0xc00c) TLS/ECDH-RSA/RC4-128/SHA
                (0xc00e) TLS/ECDH-RSA/AES128-CBC/SHA
                (0xc002) TLS/ECDH-ECDSA/RC4-128/SHA
                (0xc004) TLS/ECDH-ECDSA/AES128-CBC/SHA
                (0x0004) SSL3/RSA/RC4-128/MD5
                (0x0005) SSL3/RSA/RC4-128/SHA
                (0x002f) TLS/RSA/AES128-CBC/SHA
                (0xc008) TLS/ECDHE-ECDSA/3DES-EDE-CBC/SHA
                (0xc012) TLS/ECDHE-RSA/3DES-EDE-CBC/SHA
                (0x0016) SSL3/DHE-RSA/3DES192EDE-CBC/SHA
                (0x0013) SSL3/DHE-DSS/DES192EDE3CBC/SHA
                (0xc00d) TLS/ECDH-RSA/3DES-EDE-CBC/SHA
                (0xc003) TLS/ECDH-ECDSA/3DES-EDE-CBC/SHA
                (0xfeff) SSL3/RSA-FIPS/3DESEDE-CBC/SHA
                (0x000a) SSL3/RSA/3DES192EDE-CBC/SHA
            }
            compression[1] = {
                (00) NULL
            }
            extensions[55] = {
              extension type server_name, length [29] = {
   0: 00 1b 00 00  18 64 61 64  76 6d 61 30  31 37 38 2e  | .....soa.
  10: 75 73 2e 6f  72 61 63 6c  65 2e 63 6f  6d           | example.com
              }
              extension type elliptic_curves, length [8] = {
   0: 00 06 00 17  00 18 00 19                            | ........
              }
              extension type ec_point_formats, length [2] = {
   0: 01 00                                               | ..
              }
              extension type session_ticket, length [0]
            }
          }
   }
}
```

This is the SSL/TLS *client hello* sent from the browser to the Oracle Traffic Director instance. Note the list of cipher suites sent by the browser. These are the cipher suites that the browser is configured to handle, sorted in order of preference. The server selects one of the cipher suites for the handshake. If the server is not configured any of the cipher suites indicated by the client, the connection fails. In the above example, the session ID is empty, indicating that the browser does not have any cached SSL/TLS session with the specified server.

The Oracle Traffic Director instance's response would be similar to the following output:

```
<-- [
(823 bytes of 818)
SSLRecord { [Tue Oct 25 04:29:46 2011]
   type    = 22 (handshake)
   version = { 3,1 }
   length  = 818 (0x332)
   handshake {
      type = 2 (server_hello)
      length = 77 (0x00004d)
         ServerHello {
            server_version = {3, 1}
            random = {...}
            session ID = {
                length = 32
```

```
                contents = {...}
             }
             cipher_suite = (0x0035) TLS/RSA/AES256-CBC/SHA
             compression method = (00) NULL
             extensions[5] = {
               extension type renegotiation_info, length [1] = {
   0: 00                                                       | .
               }
             }
          }
       type = 11 (certificate)
       length = 729 (0x0002d9)
          CertificateChain {
             chainlength = 726 (0x02d6)
             Certificate {
                size = 723 (0x02d3)
                data = { saved in file 'cert.001' }
             }
          }
       type = 14 (server_hello_done)
       length = 0 (0x000000)
    }
}
]
--> [
```

The server selected the cipher suite, `TLS/RSA/AES256-CBC/SHA` and a session ID, which the client will include in subsequent requests.

The server also sent its certificate chain for the browser to verify. `ssltap` saved the certificates in the file `cert.001`. You can examine the certificates with any tool that can parse X.509 certificates. For example, run the following command:

```
$ openssl x509 -in cert.001 -text -inform DER
```

> **Note:** `ssltap` is a single threaded proxy server. So if you issue multiple requests through it, the requests will get serialized. If you need to analyze a specific problem with your application that only occurs on concurrent requests through SSL/TLS, try running multiple `ssltap` instances.

16.3.23 How do I view details of SSL/TLS communication between Oracle Traffic Director instances and Oracle WebLogic Server origin servers?

Configure SSL debugging for the Oracle WebLogic Server instance by adding the `-Dssl.debug=true` system property in the start script of the serve. For more information, see "SSL Debugging" in the *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

Increase the verbosity of the Oracle Traffic Director server log by setting the log level to `TRACE:32`, as described in Section 12.3, "Configuring Log Preferences."

16.3.24 Why are certain SSL/TLS-enabled origin servers marked offline after health checks, even though the servers are up?

This error can occur for the following origin servers:

- SSL/TLS-enabled origin servers that are configured in the origin-server pool by using IP addresses instead of host names.
- Dynamically discovered, SSL/TLS-enabled Oracle WebLogic Server origin servers. Oracle Traffic Director refers to them using their IP addresses rather than the host names.

While Oracle Traffic Director refers to such origin servers by using their IP addresses, the certificates of the origin servers contain the servers' host names. So, in response to health-check requests, when the origin servers present certificates, Oracle Traffic Director attempts, unsuccessfully, to validate them. The SSL/TLS handshake fails. As a result, the health checks show such origin servers to be offline. Note that server-certificate validation is enabled by default.

If you set the server-log level to `TRACE:32`, you can view the message logged for this failure, as shown in the following example:

```
[2011-11-21T09:50:54-08:00] [net-soa] [TRACE:1] [OTD-10969] [] [pid: 22466]
 trying to OPTIONS /, service-http reports: error sending request
 (SSL_ERROR_BAD_CERT_DOMAIN: Requested domain name does not match the server's
certificate.)
```

To solve this problem, disable validation of the origin-server certificates for the required virtual-server routes, by running the `set-route-prop` CLI command, as shown in the following example:

```
tadm> set-route-prop --config=soa  --vs=vs1 --route=route1
validate-server-cert=false
```

For the updated route to take effect, deploy the configuration by running the `deploy-config` command, as shown in the following example:

```
tadm> deploy-config soa
```

For more information about the CLI commands mentioned in this section, see the *Oracle Traffic Director Command-Line Reference* or run the command with the `--help` option.

16.3.25 Does Oracle Traffic Director rewrite the source IP address of clients before forwarding requests to the origin servers?

The default behavior of Oracle Traffic Director is to rewrite the source IP address. However, Oracle Traffic Director does send the client IP address in an additional request header `Proxy-client-ip`. You can set up Oracle Traffic Director to block or forward `Proxy-client-ip` and other request headers by configuring the appropriate route as described in Section 8.4.

Note that Oracle Traffic Director cannot maintain case sensitivity of the HTTP request headers while forwarding them to origin servers.

16.3.26 Why does Oracle Traffic Director return a 405 status code?

If an HTTP request does not meet the conditions specified in any of the defined routes and there is no default (=unconditional) route in the configuration, then Oracle Traffic Director returns the 405 status code. This error indicates that Oracle Traffic Director did not find any valid route for the request. This situation can occur only if the default route, which is used when the request does not meet the conditions specified in any of the other routes, is deleted manually in the obj.conf configuration file. To solve this issue the administrator must create a valid route.

Note: The default (=unconditional) route cannot be deleted through the administration console and the CLI, and should not be deleted manually.

16.3.27 What is the minimum supported JDK version, and JAVA_HOME variable?

Oracle Traffic Director Release 11.1.1.9 mandates JDK 7 u60 as the minimum supported JDK version. Oracle Traffic Director 11.1.1.6 bundled JDK for its own administration purposes. Oracle Traffic Director 11.1.1.9 bundles JDK 7.

Review the status of your JDK installation as follows:

- If you did not use your own JDK at the time of installing Oracle Traffic Director 11.1.1.6/7, then you do not need to consider JDK version while upgrading to Oracle Traffic Director 11.1.1.9.
- If you did use your own JDK while installing Oracle Traffic Director 11.1.1.6/.7, then you will need to now provide JDK version 7 Update 60 or above as the JDK version while upgrading to Oracle Traffic Director 11.1.1.9.

You must also have a correctly set `JAVA_HOME` variable. If you have `JAVA_HOME` set to JDK 6 in your environment. and run an Oracle Traffic Director CLI command, then you may see the following error:

```
 $ORACLE_HOME/bin/tadm configure-server --user=user1 --instance-home=$INSTANCE_
HOME/instance1  --server-user=root

Exception in thread "main" java.lang.UnsupportedClassVersionError:
com/sun/web/admin/cli/shelladapter/WSadminShell : Unsupported major.minor version
51.0
        at java.lang.ClassLoader.defineClass1(Native Method)
        at java.lang.ClassLoader.defineClassCond(ClassLoader.java:631)
        at java.lang.ClassLoader.defineClass(ClassLoader.java:615)
        at java.security.SecureClassLoader.defineClass(SecureClassLoader.java:141)
        at java.net.URLClassLoader.defineClass(URLClassLoader.java:283)
        at java.net.URLClassLoader.access$000(URLClassLoader.java:58)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:197)
        at java.security.AccessController.doPrivileged(Native Method)
        at java.net.URLClassLoader.findClass(URLClassLoader.java:190)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:306)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:301)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:247)
Could not find the main class: com.sun.web.admin.cli.shelladapter.WSadminShell.
Program will exit.
```

Workaround: remove the `JAVA_HOME` in your environment.

16.4 Contacting Oracle for Support

If you have a service agreement with Oracle, you can contact Oracle Support (http://support.oracle.com) for help with Oracle Traffic Director problems.

Before Contacting Oracle Support

Before contacting Oracle Support, do the following:

- Try all the appropriate diagnostics and troubleshooting guidelines described in this document *Oracle Traffic Director Administrator's Guide*).

- Check whether the problem you are facing, or a similar problem, has been discussed in the OTN Discussion Forums at http://forums.oracle.com/.

  If the information available on the forum is not sufficient to help you solve the problem, post a question on the forum. Other Oracle Traffic Director users on the forum might respond to your question.

- To the extent possible, document the sequence of actions you performed just before the problem occurred.
- Where possible, try to restore the original state of the system, and reproduce the problem using the documented steps. This helps to determine whether the problem is reproducible or an intermittent issue.
- If the issue can be reproduced, try to narrow down the steps for reproducing the problem. Problems that can be reproduced by small test cases are typically easier to diagnose when compared with large test cases.

  Narrowing down the steps for reproducing problems enables Oracle Support to provide solutions for potential problems faster.

Information You Should Provide to Oracle Support

When you contact Oracle for support, provide the following information.

- The release number of Oracle Traffic Director.

  To find out the release number, run the following command:

  ```
  > $ORACLE_HOME/bin/tadm --version
  Oracle Traffic Director 11.1.1.7.0 Administration Command Line B01/14/2013
  09:08
  ```

- A brief description of the problem, including the actions you performed just before the problem occurred.
- If you need support with using the administration interfaces, the name of the command-line subcommand or the title of the administration-console screen for which you require help.
- Zip file containing the configuration files for the configuration in which you encountered the error.

  INSTANCE_HOME/admin-server/config-store/*config_name*/current.zip

- Zip file containing the configuration files for the last error-free configuration.

  INSTANCE_HOME/admin-server/config-store/*config_name*/backup/*date_time*.zip

- The latest server and access log files.

Note: When you send files to Oracle Support, remember to provide the MD5 checksum value for each file, so that Oracle Support personnel can verify the integrity of the files before using them for troubleshooting the problem.

A

Metrics Tracked by Oracle Traffic Director

This appendix lists the metrics for which Oracle Traffic Director tracks and maintains statistics.

A.1 Instance Metrics

This section lists the metrics that Oracle Traffic Director tracks for individual instances. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided. Metrics that are not available through SNMP or in the stats-xml report are marked NA.

Table A–1 Instance Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of seconds the instance has been running	instanceUptime	server: secondsRunning
Number of requests processed	instanceRequests	request-bucket: countRequests
Number of octets received	instanceInOctets	request-bucket: countBytesReceived
Number of octets transmitted	instanceOutOctets	request-bucket: countBytesTransmitted
Number of 2xx (Successful) responses issued	instanceCount2xx	request-bucket: count2xx

Table A–1 (Cont.) Instance Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of 3xx (Redirection) responses issued	instanceCount3xx	request-bucket: count3xx
Number of 4xx (Client Error) responses issued	instanceCount4xx	request-bucket: count4xx
Number of 5xx (Server Error) responses issued	instanceCount5xx	request-bucket: count5xx
Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued	instanceCountOther	request-bucket: countOther
Number of 200 (OK) responses issued	instanceCount200	request-bucket: count200
Number of 302 (Moved Temporarily) responses issued	instanceCount302	request-bucket: count302
Number of 304 (Not Modified) responses issued	instanceCount304	request-bucket: count304
Number of 400 (Bad Request) responses issued	instanceCount400	request-bucket: count400
Number of 401 (Unauthorized) responses issued	instanceCount401	request-bucket: count401
Number of 403 (Forbidden) responses issued	instanceCount403	request-bucket: count403
Number of 404 (Not Found) responses issued	instanceCount404	request-bucket: count404
Number of 503 (Unavailable) responses issued	instanceCount503	request-bucket: count503
Average load in the last 1 minute	instanceLoad1Minute Average	server: load1MinuteAverage
Average load in the last 5 minutes	instanceLoad5Minute Average	server: load5MinuteAverage
Average load for in the last minutes	instanceLoad15Minut eAverage	server: load15MinuteAverage
Number of octets transmitted on the network per second	instanceNetworkInOc tets	server: rateBytesReceived
Number of octets received on the network per second	instanceNetworkOut Octets	server: rateBytesTransmitted
Average number of requests served in the last 1 minute	instanceRequests1Min uteAverage	server: requests1MinuteAverage
Average number of requests served in the last 5 minutes	instanceRequests5Min uteAverage	server: requests5MinuteAverage
Average number of requests served in the last 15 minutes	instanceRequests15Mi nuteAverage	server: requests15MinuteAverage
Average number of error responses in the last 1 minute	instanceErrors1Minut eAverage	server: errors1MinuteAverage
Average number of error responses in the last 5 minutes	instanceErrors5Minut eAverage	server: errors5MinuteAverage
Average number of error responses in the last 15 minutes	instanceErrors15Minu teAverage	server: errors15MinuteAverage

Table A–1 (Cont.) Instance Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Average response time for the requests in the last 1 minute	instanceResponseTime1MinuteAverage	server: responseTime1MinuteAverage
Average response time for the requests in the last 5 minutes	instanceResponseTime5MinuteAverage	server: responseTime5MinuteAverage
Average response time for the requests in the last 15 minutes	instanceResponseTime15MinuteAverage	server: responseTime15MinuteAverage
Number of open connections at the time when statistics were gathered	instanceCountOpenConnections	request-bucket: countOpenConnections
Name of the TCP proxy for which this element holds statistics	tcpID	tcp-proxy:name
State of the TCP proxy at the time of gathering the statistics	tcpMode	tcp-proxy:mode
IP addresses (including port) where this TCP proxy listens for requests	tcpInterfaces	tcp-proxy:interfaces
Number of active TCP proxy connections	tcpCountActiveConnections	tcp-request-bucket:countActiveConnections
Total number of requests processed	tcpCountRequests	tcp-request-bucket:countRequests
Total number of requests that were aborted	tcpCountAbortedRequests	tcp-request-bucket:countRequestsAborted
Total number of requests that were closed because of timeout	tcpCountTimeoutRequests	tcp-request-bucket:countRequestsTimedout
Number of bytes received from the client	tcpCountBytesReceived	tcp-request-bucket:countBytesReceived
Number of bytes transmitted to the clients	tcpCountBytesTransmitted	tcp-request-bucket:countBytesTransmitted
Average duration of active time in milliseconds	tcpMilliSecondsConnectionActiveAverage	tcp-request-bucket:millisecondsConnectionActiveAverage

A.2 Process Metrics

This section lists the metrics that Oracle Traffic Director tracks at the process level. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided. Metrics that are not available through SNMP or in the stats-xml report are marked NA.

Table A–2 Process Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of request processing threads currently available	processThreadCount	thread-pool-bucket: countThreads
Number of request processing threads currently idle	processThreadIdle	thread-pool-bucket: countIdleThreads
Number of connections currently in keepalive queue	processKeepaliveCount	keepalive-bucket: countConnections

Table A–2 (Cont.) Process Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Maximum number of connections allowed in keepalive queue	processKeepaliveMax	keepalive-bucket: maxConnections
Number of requests that were processed on connections in the Keep Alive subsystem	NA	keepalive-bucket: countHits
Number of connections in the Keep Alive subsystem that were flushed	NA	keepalive-bucket: countFlushes
Number of times the server could not hand off the connection to a keep-alive thread.	NA	keepalive-bucket: countRefusals
Number of connections that were closed due to Keep Alive subsystem being idle beyond the specified timeout period	NA	keepalive-bucket: countTimeouts
Idle period after which the Keep Alive subsystem should time out	NA	keepalive-bucket: secondsTimeout
Process size in kbytes	processSizeVirtual	process: sizeVirtual
Process resident size in kbytes	processSizeResident	process: sizeResident
Fraction of process memory in system memory	processFractionSystemMemoryUsage	process: fractionSystemMemoryUsage
Total number of active connections for which requests are getting processed	NA	tcp-thread:countActiveConnections

A.3 Thread Pool Metrics

This section lists the metrics that Oracle Traffic Director tracks for server threads. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided.

Table A–3 Thread Pool Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of requests queued for processing by this thread pool.	threadPoolCount	thread-pool-bucket: countQueued
Largest number of requests that have been queued simultaneously	threadPoolPeak	thread-pool-bucket: peakQueued
Maximum number of requests allowed in the queue	threadPoolMax	thread-pool-bucket: max-threads

A.4 Connection Queue Metrics

This section lists the connection-queue metrics that Oracle Traffic Director tracks. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided. Metrics that are not available through SNMP or in the stats-xml report are marked NA.

Table A–4 Connection Queue Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of connections currently in connection queue	connectionQueueCount	connection-queue-bucket: countQueued
Total number of connections that have been added to this connection queue since startup	connectionQueueTotal Queued	connection-queue-bucket: countTotalQueued
Total number of ticks spent by connections on this queue	connectionQueueTime Queued	connection-queue-bucket: ticksTotalQueued
Average length of the queue in the last one minute	NA	connection-queue-bucket: countQueued1MinuteAverage
Average length of the queue in the last one minutes	NA	connection-queue-bucket: countQueued5MinuteAverage
Average length of the queue in the last fifteen minutes	NA	connection-queue-bucket: countQueued15MinuteAverage
Largest number of connections that have been queued simultaneously	connectionQueuePeak	connection-queue-bucket: peakQueued
Maximum number of connections allowed in connection queue	connectionQueueMax	connection-queue-bucket: maxQueued
Total number of connections added to this connection queue since the instance started	connectionQueueTotalC onnections	connection-queue-bucket: countTotalConnections
Number of connections rejected due to connection queue overflow	connectionQueueOverfl ows	connection-queue-bucket: countOverflows

A.5 Compression and Decompression Metrics

This section lists the metrics for response data that Oracle Traffic Director compresses and decompresses. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided.

Table A–5 Compression and Decompression Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Total number of requests compressed	countRequestsCompressed	compression-bucket: countRequests
Total number of input bytes for compression	countBytesForCompression	compression-bucket: bytesInput
Total number of output bytes after compression	countBytesCompressed	compression-bucket: bytesOutput
Average compression per page	pageCompressionAverage	compression-bucket: pageCompressionAverage
Overall compression ratio	compressionRatio	compression-bucket: compressionRatio
Total number of requests decompressed	countRequestsDecompressed	decompression-bucket: countRequests
Total number of input bytes for decompression	countBytesForDecompression	decompression-bucket: bytesInput

Table A–5 (Cont.) Compression and Decompression Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Total number of output bytes after decompression	countBytesDecompressed	decompression-bucket: bytesOutput

A.6 Virtual Server Metrics

This section lists the metrics that Oracle Traffic Director tracks for individual virtual servers. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided.

Table A–6 Virtual Server Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of requests processed	vsRequests	request-bucket: countRequests
Number of octets received	vsInOctets	request-bucket: countBytesReceived
Number of octets transmitted	vsOutOctets	request-bucket: countBytesTransmitted
Number of 2xx (Successful) responses issued	vsCount2xx	request-bucket: count2xx
Number of 3xx (Redirection) responses issued	vsCount3xx	request-bucket: count3xx
Number of 4xx (Client Error) responses issued	vsCount4xx	request-bucket: count4xx
Number of 5xx (Server Error) responses issued	vsCount5xx	request-bucket: count5xx
Number of other (neither 2xx, 3xx, 4xx, nor 5xx) responses issued	vsCountOther	request-bucket: countOther
Number of 200 (OK) responses issued	vsCount200	request-bucket: count200
Number of 302 (Moved Temporarily) responses issued	vsCount302	request-bucket: count302
Number of 304 (Not Modified) responses issued	vsCount304	request-bucket: count304
Number of 400 (Bad Request) responses issued	vsCount400	request-bucket: count400
Number of 401 (Unauthorized) responses issued	vsCount401	request-bucket: count401
Number of 403 (Forbidden) responses issued	vsCount403	request-bucket: count403
Number of 404 (Not Found) responses issued	vsCount404	request-bucket: count404
Number of 503 (Unavailable) responses issued	vsCount503	request-bucket: count503
The total number of upgrade requests processed	websocketCountUpgradedRequests	websocket-bucket:countUpgradeRequests

Table A–6 (Cont.) Virtual Server Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of WebSocket requests that were denied upgrade by origin server	websocketCountUpgradeRejectedRequests	websocket-bucket:countUpgradeRequestsRejected
Number of WebSocket requests that were denied upgrade by server	websocketCountFailedStrictRequests	websocket-bucket:countUpgradeRequestsFailed
Number of active WebSocket connections	websocketCountActiveConnections	websocket-bucket:countActiveConnections
Total number of requests that were aborted	websocketCountAbortedRequests	websocket-bucket:countRequestsAborted
Total number of requests that were closed because of timeout	websocketCountTimeoutRequests	websocket-bucket:countRequestsTimedout
Number of bytes received from the clients	websocketCountBytesReceived	websocket-bucket:countBytesReceived
Number of bytes transmitted to the clients	websocketCountBytesTransmitted	websocket-bucket:countBytesTransmitted
Average duration of active time in millisecond	websocketMillisecondsConnectionActiveAverage	websocket-bucket:millisecondsConnectionActiveAverage
Total number of requests intercepted by webapp firewall	wafCountInterceptedRequests	webapp-firewall-bucket:countRequestsIntercepted
Total number of requests allowed by webapp firewall (allow action)	wafCountAllowedRequests	webapp-firewall-bucket:countRequestsAllowed
Total number of denied requests (deny action)	wafCountDeniedRequests	webapp-firewall-bucket:countRequestsDenied
Total number of dropped requests (drop action)	wafCountDroppedRequests	webapp-firewall-bucket:countRequestsDropped
Total number of redirected requests (redirect action)	wafCountRedirectedRequests	webapp-firewall-bucket:countRequestsRedirected
Total number of detected denied requests (deny action)	wafCountDenyDetectedRequests	webapp-firewall-bucket:countRequestsDenyDetected
Total number of detected dropped requests (drop action)	wafCountDropDetectedRequests	webapp-firewall-bucket:countRequestsDropDetected
Total number of detected redirected requests (redirect action)	wafCountRedirectDetectedRequests	webapp-firewall-bucket:countRequestsRedirectDetected

A.7 CPU Metrics

This section lists the CPU-related metrics that Oracle Traffic Director tracks. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided.

Table A–7 CPU Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Percentage of the time that the CPU is idle	cpuIdleTime	cpu-info: percentIdle
Percentage of the time the CPU is spending in user space	cpuUserTime	cpu-info: percentUser
Percentage of the time the CPU is spending in kernel space	cpuKernelTime	cpu-info: percentKernel

A.8 Origin Server Metrics

This section lists the metrics that Oracle Traffic Director tracks for origin server pools and origin servers. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided.

Table A–8 Origin Server Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Number of times a request was retried (to same or different origin server)	serverPoolCountRetries	server-pool: countRetries
Type of the server pool		server-pool:type
Type of origin-server, whether a Weblogic server, a generic HTTP server or not unable to detect		origin-server-bucket: type
Flag indicating whether the node was dynamically discovered	originServerDiscoveryStatus	origin-server-bucket: flagDiscovered
Flag indicating whether the node is fully ramped up	originServerRampedupStatus	origin-server-bucket: flagRampedUp
Flag indicating whether the origin server is a backup node	originServerBackupStatus	origin-server-bucket: flagBackup
Status indicating whether the origin server is currently marked online	originServerRunningStatus	origin-server-bucket: status
Total time, in seconds, since the origin server was marked online	originServerRunningTime	origin-server-bucket: secondsOnline
Total number of times the origin server was marked offline	originServerCountOffline	origin-server-bucket: countDetectedOffline
Total number of bytes transmitted to the origin server	originServerCountBytesTransmitted	origin-server-bucket: countBytesTransmitted
Total number of bytes received from the origin server	originServerCountBytesReceived	origin-server-bucket: countBytesReceived
Total number of open connections to the origin server for which requests are getting processed	originServerCountActiveConnections	origin-server-bucket: countActiveConnections
Total number of idle connections to the origin server	originServerCountIdleConnections	origin-server-bucket: countIdleConnections

Table A–8 (Cont.) Origin Server Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Total number of active connections belonging to sticky requests when time statistics were collected	originServerCountActiveStickyConnections	origin-server-bucket:countActiveStickyConnections
Total number of times a connection to the origin server was attempted	originServerCountConnectAttempts	origin-server-bucket: countConnectAttempts
Total number of times an attempt to connect to the origin server failed	originServerCountConnectFailures	origin-server-bucket: countConnectFailures
Total number of requests that were aborted when proxying requests with this origin server	originServerCountRequestsAborted	origin-server-bucket: countRequestsAborted
Total number of times the request timed out when sending or receiving data from the origin server	originServerCountRequestsTimedout	origin-server-bucket: countRequestsTimedout
Total number of requests served by the origin server	originServerCountRequests	origin-server-bucket: countRequests
Total number of health check requests	originServerCountHealthCheckRequests	origin-server-bucket:countHealthCheckRequests
Total number of connections closed	originServerCountConnectionsClosed	origin-server-bucket:countConnectionsClosed
Total number of keep-alive connections closed by the origin server	originServerCountConnectionsClosedByOriginServer	origin-server-bucket:countConnectionsClosedByOriginServer
Dynamically calculated keep-alive timeout value for the origin server	originServerSecondsKeepAliveTimeout	origin-server-bucket:secondsKeepAliveTimeout
Total number of sticky requests	originServerCountStickyRequests	origin-server-bucket:countStickyRequests
Dynamic weight detected based on response time (applicable when algorithm is least-response-time)	originServerWeightResponseTime	origin-server-bucket:weightResponseTime
Type of origin-server (generic/weblogic/undetected)	originServerType	origin-server-bucket:type
Average duration of active time in milliseconds	originServerMillisecondsConnectionActiveAverage	origin-server-bucket:millisecondsConnectionActiveAverage

A.9 Failover Instance Metrics

This section lists the metrics for each VIP in the server instance. These metrics show the current state of the failover instance, as well as which nodes are configured as primary and backup for a failover group.

Table A–9 Failover Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Actual current state of this failover instance. An integer (1 if active, 0 for not active).	flagActive	flagActive

Table A–9 (Cont.) Failover Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Name of the node which is configured as backup	backupNode	backupNode
Name of the node which is configured as primary	primaryNode	primaryNode
Virtual IP address of the failover group	virtualIp	virtualIp

A.10 Proxy Cache Metrics

This section lists the caching-related metrics that Oracle Traffic Director tracks. For each metric, the object name in the SNMP MIB and the names of the corresponding element and attribute in the stats-xml report are provided.

Table A–10 Proxy Cache Metrics

Metric	Object Name in the SNMP MIB	stats-xml Element: Attribute
Total number of entries in the cache	proxyCacheCountEntries	cache-bucket: countEntries
Amount of heap space used by cache content	proxyCacheSizeHeap	cache-bucket: sizeHeapCache
Total number of times a cache lookup succeeded	proxyCacheCountContentHits	cache-bucket: countContentHits
Total number of times a cache lookup failed	proxyCacheCountContentMisses	cache-bucket: countContentMisses
Total number of times an entry was served from cache	proxyCacheCountHits	cache-bucket: countHits
Total number of requests that were revalidated from the origin server	proxyCacheCountRevalidationRequests	cache-bucket: countRevalidationRequests
Total number of times the revalidation requests failed	proxyCacheCountRevalidationFailures	cache-bucket: countRevalidationFailures

A.11 DNS Cache Metrics

This section lists the DNS cache lookup metrics that Oracle Traffic Director tracks. For each metric, the element and attribute in the stats-xml report are provided.

Table A–11 DNS Cache Metrics

Metric	stats-xml Element: Attribute
Total number of entries in the cache	dns-bucket: countCacheEntries
Total number of times a cache lookup succeeded	dns-bucket: countCacheHits
Total number of times a cache lookup failed	dns-bucket: countCacheMisses
Number of asynchronous lookups	dns-bucket: countAsyncNameLookups
Total number of asynchronous DNS address lookups performed	dns-bucket: countAsyncAddrLookups
Number of asynchronous DNS lookups currently in progress	dns-bucket: countAsyncLookupsInProgress

B

Web Application Firewall Examples and Use Cases

The attack prevention feature of web application firewall stands between the client and origin servers. If the web application firewall finds a malicious payload, it will reject the request, performing any one of the built-in actions. This section provides some basic information about how web application firewall works and how some rules are used for preventing attacks. For information about managing and configuring web application firewall, see Section 11.7, "Managing Web Application Firewalls."

Some of the features of web application firewall are audit logging, access to any part of the request (including the body) and the response, a flexible rule engine, file-upload interception, real-time validation and buffer-overflow protection.

Web application firewall's functionality is divided into four main areas:

- Parsing: Parsers extract bits of each request and/or response, which are stored for use in the rules.
- Buffering: In a typical installation, both request and response bodies are buffered so that the module generally sees complete requests (before they are passed to the application for processing), and complete responses (before they are sent to clients). Buffering is the best option for providing reliable blocking.
- Logging: Logging is useful for recording complete HTTP traffic, allowing you to log all response/request headers and bodies.
- Rule engine: Rule engines work on the information from other components, to evaluate the transaction and take action, as required.

B.1 Basics of Rules

The web application firewall rule engine is where gathered information is checked for any specific or malicious content.

This section provides information about basic rule-writing syntax, and rule directives for securing Web applications from attacks.

The main directive that is used for creating rules is `SecRule`. The syntax for `SecRule` is:

```
SecRule VARIABLES OPERATOR [TRANSFORMATION_FUNCTIONS, ACTIONS]
```

- VARIABLES: Specify where to check in an HTTP transaction. Web application firewall pre-processes raw transaction data, which makes it easy for rules to focus on the logic of detection. A rule must specify one or more variables. Multiple rules can be used with a single variable by using the `|` operator.

- OPERATORS: Specify how a *transformed* variable is to be analyzed. Operators always begin with an @ character, and are followed by a space. Only one operator is allowed per rule.
- TRANSFORMATION_FUNCTIONS: Change input in some way before the rule operator is run. A rule can specify one or more transformation functions.
- ACTIONS: Specify the required action if the rule evaluates to true, which could be, display an error message, step on to another rule, or some other task.

Here is an example of a rule:

```
SecRule ARGS|REQUEST_HEADERS "@rx <script" msg:'XSSAttack',deny,status:404
```

- `ARGS` and `REQUEST_HEADERS` are variables (request parameters and request headers, respectively).
- `@rx` is the regular expression operator. It is used to match a pattern in the variables.

  In the example, the pattern is `<script`.
- `msg`, `deny` and `status` are actions to be performed if a pattern is matched.

  The rule in the example is used to avoid XSS attacks, which is done by checking for a `<script` pattern in the request parameters and header, and an XSS Attack log message is generated. Any matching request is denied with a 404 status response.

B.2 Rules Against Major Attacks

This section provides information about some rules that are used for preventing major attacks on Web applications.

B.2.1 Brute Force Attacks

Brute force attacks involve an attacker repeatedly trying to gain access to a resource by guessing usernames, passwords, e-mail addresses, and similar credentials. Brute force attacks can be very effective if no protection is in place, especially when users choose passwords that are short and easy to remember.

A good way to defend against brute force attacks is to allow a certain number of login attempts, after which the login is either delayed or blocked. Here is an example of how this can be accomplished using Oracle Traffic Director web application firewall.

If your login verification page is situated at `yoursite.com/login` and is served by the virtual server `waf-vs`, then the following rules, in `waf-vs.conf` file configured at the virtual server level, will keep track of the number of login attempts by the users:

```
# Block further login attempts after 3 failed attempts

# Initalize IP collection with user's IP address
SecAction "initcol:ip=%{REMOTE_ADDR},pass,nolog"

# Detect failed login attempts
SecRule RESPONSE_BODY "Unauthorized" "phase:4,pass,setvar:ip.failed_
logins=+1,expirevar:ip.failed_logins=60"

# Block subsequent login attempts
SecRule IP:FAILED_LOGINS "@gt 2" deny
```

The rules initialize the IP collection and increment the field `IP:FAILED_LOGINS` after each failed login attempt. When more than three failed logins are detected, further attempts are blocked. The `expirevar` action is used to reset the number of failed login

attempts to zero after 60 seconds, so the block will be in effect for a maximum of 60 seconds.

To use the persistent collection, IP, you should specify the path to store the persisted data using the `SecDataDir` directive. Since the scope of this directive is Main, it should be specified at the server level. This can be accomplished as follows:

```
# The name of the debug log file
SecDebugLog ../logs/brute_force_debug_log

# Debug log level
SecDebugLogLevel 3

# Enable audit logging
SecAuditEngine On

# The name of the audit log file
SecAuditLog ../logs/brute_force_audit_log

# Path where persistent data is stored
SecDataDir "/var/run/otd/waf/"
```

If this rules file is called `waf-server.conf` , `<instance-dir>/config/server.xml` would look like this:

```
<server>
...
...
   <webapp-firewall-ruleset>/waf-rules/waf-server.conf</webapp-firewall-ruleset>
...
...
    <virtual-server>
      <name>waf-vs</name>
      <host>yoursite.com</host>
      ...
      <object-file>waf-vs-obj.conf</object-file>
      <webapp-firewall-ruleset>/waf-rules/waf-vs.conf</webapp-firewall-ruleset>
    </virtual-server>
...
...
</server>
```

Web application firewall and response body processing (equivalent of `SecResponseBodyAccess` directive) should be enabled for the `/login` URI in `waf-vs-obj.conf`. `waf-vs-obj.conf` would look like this:

```
<Object name="default">
<If $uri eq "/login">
AuthTrans fn="webapp-firewall" process-response-body="on"
</If>
...
...
</Object>
```

After 3 failed attempts to login, audit log would have the following message:

```
--5c4adf36-A--
[19/Mar/2013:05:06:57 --0700] ygfh3010000000000,0 127.0.0.1 49619 127.0.0.1 5021
--5c4adf36-B--
GET /acl/acl02.html HTTP/1.1
user-agent: curl/7.15.5 (x86_64-redhat-linux-gnu) libcurl/7.15.5 OpenSSL/0.9.8b
zlib/1.2.3 libidn/0.6.5
```

```
accept: */*
host: yoursite.com
authorization: Basic YWxwaGE6YmV0YQ==

--5c4adf36-F--
HTTP/1.1 403 Forbidden
status: 403 Forbidden
content-length: 208
content-type: text/html

--5c4adf36-H--
Message: Warning. Unconditional match in SecAction. [file
"/waf-rules/waf-vs.conf"] [line "10"]
Message: Access denied with code 403 (phase 2). Operator GT matched 2 at
IP:failed_logins. [file "/waf-rules/waf-vs.conf"] [line "25"]
Action: Intercepted (phase 2)
Stopwatch: 1363694817000000 898560 (- - -)
Stopwatch2: 1363694817000000 898560; combined=370, p1=14, p2=336, p3=0, p4=0,
p5=19, sr=131, sw=1, l=0, gc=0
Producer: ModSecurity for Apache/2.6.7 (http://www.modsecurity.org/).
Server: Oracle Traffic Director/11.1.1.7

--5c4adf36-Z--
```

B.2.2 SQL Injection

SQL injection attacks can occur if an attacker is able to supply data to a Web application that is then used in unsanitized form in an SQL query. This can cause the SQL query to do something that is completely different from what was intended by the developers of the Web application. For example, an attacker can try deleting all records from a MySQL table, like this:

```
http://www.example.com/login.php?user=user1';DELETE%20FROM%20users--
```

This can be prevented by using the following directives:

```
SecDefaultAction "phase:2,log,auditlog,deny,status:403"
SecRule ARGS
"(select|create|rename|truncate|load|alter|delete|update|insert|desc)\s*"
"t:lowercase,msg:'SQL Injection'"
```

Whenever the web application firewall engine spots such a request, something similar to the following code is logged to `audit_log`:

```
--3923b655-A--
[20/Mar/2013:02:58:35 --0700] Xkjx6010000000000,0 127.0.0.1 35971 127.0.0.1 5021
--3923b655-B--
GET /acl/acl02.html?user=user1';DELETE%20FROM%20users-- HTTP/1.1
host: waf.test.com
connection: close

--3923b655-F--
HTTP/1.1 403 Forbidden
status: 403 Forbidden
content-length: 208
content-type: text/html
connection: close

--3923b655-H--
Message: Access denied with code 403 (phase 2). Pattern match
"(select|create|rename|truncate|load|alter|delete|update|insert|desc)\\s*" at
```

```
ARGS:user. [file "/waf-rules/sql_injection_attack.conf"] [line "2"] [msg "SQL
Injection"]
Action: Intercepted (phase 2)
Stopwatch: 1363773515000000 668049 (- - -)
Stopwatch2: 1363773515000000 668049; combined=131, p1=8, p2=104, p3=0, p4=0,
p5=19, sr=0, sw=0, l=0, gc=0
Producer: ModSecurity for Apache/2.6.7 (http://www.modsecurity.org/).
Server: Oracle Traffic Director/11.1.1.7

--3923b655-Z--
```

In response to the attack, `SecDefaultAction` is applied. in which case the request is denied and logged, and the attacker receives a 403 error. If you would like a different action to take place, such as redirect the request to an HTML page with a customized warning content, specify it in the rule, as follows:

```
SecRule ARGS
"(select|create|rename|truncate|load|alter|delete|update|insert|desc)\s*"
"t:lowercase,msg:'SQL Injection',redirect:http://yoursite.com/invalid_request.html
```

B.2.3 XSS Attacks

Cross-site scripting (XSS) attacks occur when user input is not properly sanitized and ends up in pages sent back to users. This makes it possible for an attacker to include malicious scripts in a page by providing them as input to the page. The scripts will be no different from scripts included in pages by creators of the website, and will thus have all the privileges of an ordinary script within the page, such as the ability to read cookie data and session IDs.

Here is an example of a simple rule to block `<script` in the request parameter:

```
SecDefaultAction phase:2,deny,status:403,log,auditlog
SecRule REQUEST_COOKIES|REQUEST_COOKIES_NAMES|REQUEST_FILENAME|ARGS_
NAMES|ARGS|XML:/* "(?i:<script.*?>)"
"phase:2,capture,t:none,t:htmlEntityDecode,t:compressWhiteSpace,t:lowercase,block,
msg:'Cross-site Scripting (XSS) Attack',id:'101'"
```

C

Securing Oracle Traffic Director Deployment

This appendix provides information about the steps that you can take to secure your Oracle Traffic Director deployment.

For information about securing access to the Oracle Traffic Director administration server and enabling SSL/TLS, see Chapter 11, "Managing Security."

C.1 Securing Oracle Traffic Director

The following are some of the steps that you can perform to secure Oracle Traffic Director in your environment:

- Configure your system firewall to ensure that:
  - Oracle Traffic Director server instance ports are accessible for external traffic. The default port is 8989. For information about how to find port information for various instances, see Section 3.2, "Viewing a List of Administration Nodes."
  - Oracle Traffic Director administration port is only accessible for internal traffic.
  - Oracle Traffic Director administration node can communicate with the administration server.
- Alternatively you could ensure that Oracle Traffic Director administration nodes can only listen on private interfaces such as `bond0`, which is not available to external traffic. For more information, see Chapter 3, "Managing Administration Nodes."
- Ensure Oracle Traffic Director server instance is running as non-`root` and not listening on all interfaces. For information about starting Oracle Traffic Director instances, see Section 5.3, "Starting, Stopping, and Restarting Oracle Traffic Director Instances."

  Note: For each Oracle Traffic Director configuration that you instantiate on an administration node, a subdirectory named `net-`*`config_name`* is created in the `INSTANCE_HOME` subdirectory.

- Leverage the ability of Oracle Traffic Director to provide high availability as non-`root`. For more information, see Chapter 14, "Configuring Oracle Traffic Director for High Availability."
- Ensure that sufficient file descriptors are available. For more information, see Section 15.3, "Tuning the File Descriptor Limit."

- Ensure that appropriate network level protections are taken care. For more information, see http://www.oracle.com/technetwork/articles/servers-storage-admin/secure-linux-env-1841089.html.

  In addition, you should consider hardening your system. For information about hardening an Oracle Linux system, see http://www.oracle.com/technetwork/articles/servers-storage-admin/tips-harden-oracle-linux-1695888.html.