Cisco 4700 Series Application Control Engine Appliance Quick Start Guide

Software Version A3(1.0)

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-17506-01

CONTENTS

Preface

This guide provides the following information:

- An overview of the major functions and features of the Cisco 4700 Series Application Control Engine (ACE) appliance
- Instructions on how to initially configure the ACE to allow traffic and basic load balancing
- Instructions on how to configure the ACE to provide various scalability and security capabilities
- References to find the information in the documentation set

This preface contains the following major sections:

- Audience
- How to Use This Guide
- Related Documentation
- Symbols and Conventions
- Obtaining Documentation and Submitting a Service Request
- Open Source License Acknowledgements

Audience

This guide is intended for the following trained and qualified service personnel who are responsible for configuring the ACE:

- Web master
- System administrator
- System operator

How to Use This Guide

This guide is organized as follows:

Chapter	Description
Chapter 1, Overview	Provides an overview of the major functions and features of the ACE
Chapter 2, Setting Up an ACE Appliance	Provides procedures to initially configure the ACE to allow the passing of traffic and remote access
Chapter 3, Creating a Virtual Context	Provides procedures to partition the ACE into virtual contexts for more efficient operation
Chapter 4, Configuring Access Control Lists	Provides procedures to configure an access control list in an ACE to secure your network
Chapter 5, Configuring Role-Based Access Control	Provides procedures to configure a user with permission to perform limited operations and access a subset of your network
Chapter 6, Configuring Server Load Balancing	Provides procedures to configure the ACE to allow basic server load balancing
Chapter 7, Configuring a Load-Balancing Predictor	Provides procedures to select a predefined predictor for server load balancing
Chapter 8, Configuring Server Persistence Using Stickiness	Provides procedures to configure server persistence for requests from a client using stickiness

Chapter	Description
Chapter 9, Configuring SSL Security	Provides procedures to configure SSL security for your network
Chapter 10, Configuring Health Monitoring Using Health Probes	Provides procedures to configure server health monitoring using health probes

If you are already familiar with the ACE appliance and would like to quickly set up the device for basic server load balancing, you can follow the configuration procedures in the following chapters:

- Chapter 2, Setting Up an ACE Appliance
- Chapter 3, Creating a Virtual Context
- Chapter 6, Configuring Server Load Balancing

The remaining chapters allow you to explore additional capabilities of the ACE.

Related Documentation

In addition to this document, the ACE documentation set includes the following documents:

Document Title	Description
Release Note for the Cisco 4700 Series Application Control Engine Appliance	Provides information about operating considerations, caveats, and CLI commands for the ACE appliance.
Cisco 4710 Application Control Engine Appliance Hardware Installation Guide	Provides information for installing the ACE appliance.

Document Title	Description
Cisco 4700 Series Application Control Engine Appliance Device Manager GUI Configuration Guide	Describes how to configure the ACE using the Device Manager GUI and provides background details about the attributes used in the GUI.
Cisco 4700 Series Application Control Engine Appliance Command Reference	Provides an alphabetical list and descriptions of all CLI commands by mode, including syntax, options, and related commands.
Cisco 4700 Series Application Control Engine Appliance Administration Guide	Describes how to perform the following administration tasks on the ACE: • Setting up the ACE • Establishing remote access • Managing software licenses • Configuring class maps and policy maps • Managing the ACE software • Configuring SNMP • Configuring redundancy • Configuring the XML interface • Upgrading the ACE software
Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide	Describes how to operate your ACE in a single context or in multiple contexts and how to configure Role-Based Access Control.

Document Title	Description
Cisco 4700 Series Application Control Engine Appliance Routing and Bridging Configuration Guide	Describes how to configure the following routing and bridging tasks on the ACE: • VLAN interfaces • Routing • Bridging • Dynamic Host Configuration Protocol (DHCP)
Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide	Describes how to configure the following server load-balancing tasks on the ACE: • Real servers and server farms • Class maps and policy maps to load-balance traffic to real servers in server farms • Server health monitoring (probes) • Stickiness • Firewall load balancing • TCL scripts
Cisco 4700 Series Application Control Engine Appliance Security Configuration Guide	Describes how to perform the following ACE security configuration tasks: • Access control lists (ACLs) • User authentication and accounting using a Terminal Access Controller Access Control System Plus (TACACS+), Remote Authentication Dial-In User Service (RADIUS), or Lightweight Directory Access Protocol (LDAP) server • Application protocol and HTTP deep packet inspection • TCP/IP normalization and termination parameters • Network address translation (NAT)

Document Title	Description
Cisco 4700 Series Application Control Engine Appliance SSL Configuration Guide	Describes how to configure the following SSL tasks on the ACE: • SSL certificates and keys • SSL initiation • SSL termination • End-to-end SSL
Cisco 4700 Series Application Control Engine Appliance System Message Guide	Describes how to configure system message logging on the ACE. This guide also lists and describes the system log (syslog) messages generated by the ACE.
Cisco CSM-to-ACE Conversion Tool User Guide	Describes how to use the CSM-to-ACE conversion tool to migrate Cisco Content Switching Module (CSM) running or startup configuration files to the ACE.
Cisco CSS-to-ACE Conversion Tool User Guide	Describes how to use the CSS-to-ACE conversion tool to migrate Cisco Content Services Switches (CSS) running or startup configuration files to the ACE.

Symbols and Conventions

This publication uses the following conventions:

Convention	Description
boldface font	Commands, command options, and keywords are in **boldface**. Bold text also indicates a command in a paragraph.
italic font	Arguments for which you supply values are in *italics*.
[ ]	Elements in square brackets are optional.
{ x | y | z }	Alternative keywords are grouped in braces and separated by vertical bars.

Convention	Description
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`screen` font	Terminal sessions and information the system displays are in `screen` font.
`boldface screen` font	Information you must enter on a command line is in **`boldface screen`** font.
italic screen font	Arguments for which you supply values are in *italic screen* font.
^	The symbol ^ represents the key labeled Control—for example, the key combination ^D in a screen display means hold down the Control key while you press the D key.
< >	Nonprinting characters, such as passwords are in angle brackets.

1. A numbered list indicates that the order of the list items is important.
   a. An alphabetical list indicates that the order of the secondary list items is important.

- A bulleted list indicates that the order of the list topics is unimportant.
  – An indented list indicates that the order of the list subtopics is unimportant.

Notes use the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.

Cautions use the following conventions:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see the monthly *What's New in Cisco Product Documentation*, which also lists all new and revised Cisco technical documentation, at:

http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html

Subscribe to the *What's New in Cisco Product Documentation* as a Really Simple Syndication (RSS) feed and set content to be delivered directly to your desktop using a reader application. The RSS feeds are a free service and Cisco currently supports RSS Version 2.0.

Notices

The following notices pertain to this software license.

OpenSSL/Open SSL Project

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com).

License Issues

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License:

Copyright © 1998-2007 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions, and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.

4. The names “OpenSSL Toolkit” and “OpenSSL Project” must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.
5. Products derived from this software may not be called “OpenSSL” nor may “OpenSSL” appear in their names without prior written permission of the OpenSSL Project.
6. Redistributions of any form whatsoever must retain the following acknowledgment:

   “This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)”.

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT “AS IS”' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License:

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com).

The implementation was written so as to conform with Netscapes SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the

SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:

   "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)".

   The word 'cryptographic' can be left out if the routines from the library being used are not cryptography-related.
4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)".

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY,

OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The license and distribution terms for any publicly available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution license [including the GNU Public License].

CHAPTER 1

Overview

The Cisco 4700 Series Application Control Engine (ACE) appliance performs server load balancing, network traffic control, service redundancy, resource management, encryption and security, and application acceleration and optimization, all in a single network appliance.

This chapter contains a high-level introduction to the following topics:

- ACE Technologies
- Setting Up an ACE Appliance
- Creating Virtual Contexts
- Configuring Access Control Lists
- Configuring Role-Based Access Control
- Configuring a Virtual Server
- Configuring a Load-Balancing Predictor
- Configuring Server Persistence Using Stickiness
- Configuring SSL Security
- Configuring Health Monitoring Using Health Probes

ACE Technologies

Server load balancing helps ensure the availability, scalability, and security of applications and services by distributing the work of a single server across multiple servers.

When you configure server load balancing on your ACE appliance, the ACE decides which server should receive a client request such as a web page or a file. The ACE selects a server that can successfully fulfill the client request most effectively, without overloading the selected server or the overall network.

Table 1-1 shows the ACE technologies that provide network availability, scalability, and security at both the device and network services levels.

Table 1-1 ACE Technologies

<table>
<tr><th>Level</th><th>Availability</th><th>Scalability</th><th>Security</th></tr>
<tr><td rowspan="2">Device</td><td rowspan="2">Device Setup</td><td>Virtual Contexts</td><td>Access Control Lists</td></tr>
<tr><td colspan="2">Role-Based Access Control</td></tr>
<tr><td rowspan="2">Network Services</td><td rowspan="2">Virtual Servers
Health Probes</td><td>Load Balancing Predictors
Server Persistence Using Stickiness</td><td>SSL
Access Control Lists</td></tr>
<tr><td colspan="2">Role-Based Access Control</td></tr>
</table>

At the device level, the ACE provides high network availability by supporting:

- Device redundancy—The high availability support of the ACE allows you to set up a peer ACE device to the configuration so that if one ACE becomes inoperative, the other ACE can take its place immediately.
- Scalability—Supports virtualization by partitioning one ACE device into independent virtual devices, each with its own resource allocation.
- Security—Supports access control lists which restrict access from certain clients or to certain network resources.

At the network service level, the ACE provides:

- High services availability—Supports high-performance server load balancing, which distributes client requests among physical servers and server farms, and provides health monitoring at the server and server farm levels through implicit and explicit health probes.
- Scalability—Supports virtualization using advanced load-balancing algorithms (predictors) to distribute client requests among the virtual devices configured in the ACE. Each virtual device includes multiple virtual servers. Each server forwards client requests to one of the server farms. Each server farm can contain multiple physical servers.

  Although the ACE can distribute client requests among hundreds or even thousands of physical servers, it can also maintain server persistence. With some e-commerce applications, all client requests within a session are directed to the same physical server so that all the items in one shopping cart are contained on one server.
- Services-level security—Establishes and maintains a Secure Sockets Layer (SSL) session between the ACE and its peer which provides secure data transactions between clients and servers.

Setting Up an ACE Appliance

To set up an ACE appliance, you first establish a connection to the ACE and perform the initial device setup required to prepare the ACE for providing application networking services. For more information, see Chapter 2, "Setting Up an ACE Appliance."

Creating Virtual Contexts

Next, you partition the ACE device into multiple virtual contexts, each with its own resource allocation. For more information, see Chapter 3, "Creating a Virtual Context."

Configuring Access Control Lists

Then, you control access to your network resources to guarantee that only desired traffic passes through, and that the appropriate users can access the network resources they need.

You use Access Control Lists (ACLs) to secure your network by permitting or denying traffic to or from a specific IP address or an entire network.

You must configure an ACL for each interface on which you want to permit connections. Otherwise, the ACE will deny all traffic on that interface. An ACL consists of a series of ACL permit-or-deny entries, with criteria for the source IP address, destination IP address, protocol, port, or protocol-specific parameters. Each entry permits or denies inbound or outbound network traffic to the parts of your network specified in the entry.

This guide provides an example of ACL configuration at the device level (see Chapter 4, "Configuring Access Control Lists"). To learn how to configure ACL at the network services level, or how to configure more granular access control security, see the *Cisco 4700 Series Application Control Engine Appliance Security Configuration Guide*.

Configuring Role-Based Access Control

You can manage the complexity of large-network security administration by defining the commands and resources available to each user through Role-Based Access Control (RBAC). RBAC supports network security at both the device and network services levels by defining physical or virtual resources in a domain that the user can access.

For more information, see Chapter 5, "Configuring Role-Based Access Control."

Configuring a Virtual Server

You can configure a virtual server to intercept web traffic to a website and allow multiple real servers (physical servers) to appear as a single server for load-balancing purposes.

Table 1-2 illustrates how the ACE supports scalability through virtual contexts, virtual servers, server farms, and real servers.

Table 1-2 ACE Scalability

ACE	Virtual Context 1	Virtual Server A	Server Farm A	Real Server A1
				Real Server A2
				……
				Real Server An
			Backup Server Farm a	Real Server a1
				Real Server a2
				……
				Real Server an
		Virtual Server B	Server Farm B	Real Server B1
				Real Server B2
				……
				Real Server Bn
	Virtual Context 2	Virtual Server C	Server Farm C	Real Server C1
				Real Server C2
				……
				Real Server Cn
		Virtual Server D	Server Farm D	Real Server D1
				Real Server D2
				……
				Real Server Dn
	……	……	……	……

You can partition your ACE into multiple virtual contexts, each of which has its own set of policies, interfaces, and resources. A virtual server is bound to physical resources that run on a real server in a server farm.

Real servers relate to the actual, physical servers on your network. They can be configured to provide client services or as backup servers.

Related real servers are grouped into server farms. Servers in the same server farm often contain identical content (referred to as mirrored content) so that if one server becomes inoperative, another server can take over its functions immediately. Mirrored content also allows several servers to share the load during times of increased demand.

For more information, see Chapter 6, "Configuring Server Load Balancing."

Configuring a Load-Balancing Predictor

To distribute incoming client requests among the servers in a server farm, you define load-balancing rules called predictors using IP address and port information.

When there is a client request for an application service, the ACE performs server load balancing by deciding which server can successfully fulfill the client request in the shortest amount of time without overloading the server or server farm. Some sophisticated predictors take into account factors such as a server's load, response time, or availability, allowing you to adjust load balancing to each application's particular past.

For more information, see Chapter 7, "Configuring a Load-Balancing Predictor."

Configuring Server Persistence Using Stickiness

You can configure the ACE to allow the same client to maintain multiple simultaneous or subsequent TCP or IP connections with the same real server for the duration of a session. A session is defined as a series of interactions between a client and a server over some finite period of time (from several minutes to several hours). Cisco calls this server persistence feature stickiness.

Many network applications require that customer-specific information be stored persistently across multiple server requests. A common example is a shopping cart used on an e-commerce site. With server load balancing in use, it could potentially be a problem if a back-end server needs information generated at a different server during a previous request.

Depending on how you have configured server load balancing, the ACE sticks a client to an appropriate server after it has determined which load-balancing method to use. If the ACE determines that a client is already stuck to a particular server, then the ACE sends subsequent client requests to that server, regardless of the load-balancing criteria. If the ACE determines that the client is not stuck to a particular server, it applies the normal load-balancing rules to the request.

The combination of the predictor and stickiness enables the application to have scalability, availability, and performance even with persistence for transaction processing.

For more information, see Chapter 8, “Configuring Server Persistence Using Stickiness.”

Configuring SSL Security

Use the SSL security protocol for authentication, encryption, and data integrity in a Public Key Infrastructure (PKI).

SSL configuration in an ACE establishes and maintains an SSL session between the ACE and its peer, enabling the ACE to perform its load-balancing tasks on the SSL traffic. These SSL functions include server authentication, private-key and public-key generation, certificate management, and data packet encryption and decryption.

For more information, see Chapter 9, “Configuring SSL Security.”

Configuring Health Monitoring Using Health Probes

Application services require monitoring to ensure availability and performance. You can configure the ACE to track the health and performance of your servers and server farms by creating health probes. Each health probe that you create can be associated with multiple real servers or server farms.

When you enable ACE health monitoring, the appliance periodically sends messages to the server to determine server status. The ACE verifies the server’s response to ensure that a client can access that server. The ACE can use the server’s response to place the server in or out of service. In addition, the ACE can use the health of servers in a server farm to make reliable load-balancing decisions.

For more information, see Chapter 10, “Configuring Health Monitoring Using Health Probes.”

CHAPTER 2

Setting Up an ACE Appliance

This chapter describes how to set up a Cisco 4700 Series Application Control Engine (ACE) appliance. It includes the following major sections:

Overview

After reading this chapter, you should have a basic understanding of how to configure a ACE appliance with the networking parameters necessary for communicating with a management device to configure server load balancing.

After some initial setup using the CLI, you can complete the procedures in this chapter using the Device Manager GUI.

Before performing the procedures in this section, make sure that you complete the ACE installation instructions as described in the *Cisco 4710 Application Control Engine Appliance Hardware Installation Guide*.

Configuring an ACE involves the following basic steps:

Step 1 Establishing a console connection on the ACE.

Step 2 Enable management connectivity to the ACE through a Gigabit Ethernet port.

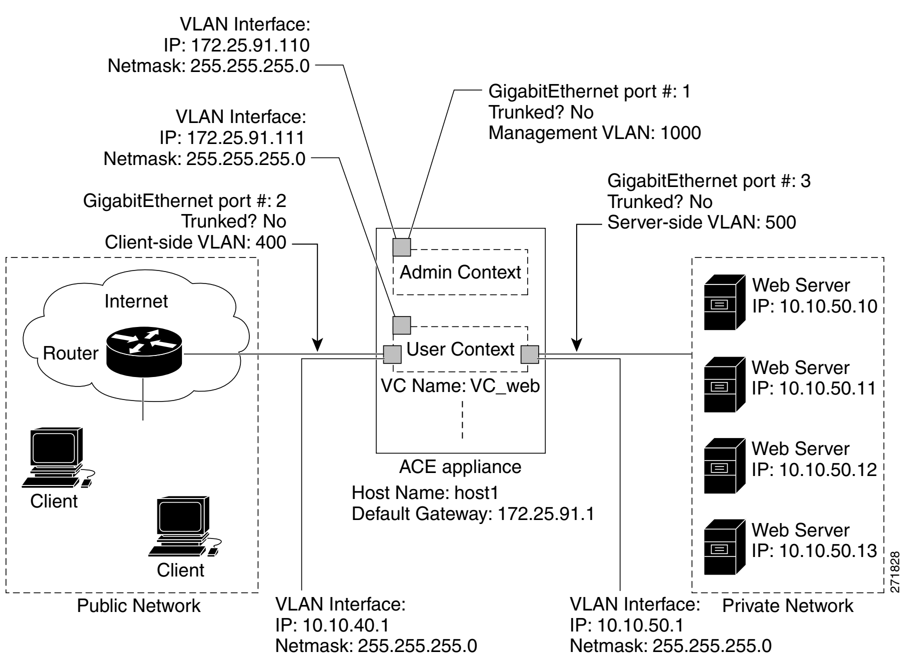
Step 3 Log in to the ACE.

Step 4 Configure a second Gigabit Ethernet port for client-side connectivity.

Step 5 Configure a third Gigabit Ethernet port for server-side connectivity.

This chapter describes how to set up an ACE appliance using the example network setup illustrated in Figure 2-1.

Figure 2-1 Example Network Setup



The configuration of the example setup is as follows:

- VLAN 1000 is assigned to the first Gigabit Ethernet port and is used for management traffic for both the Admin context and a user context.

Note A virtual local area network (VLAN) is a logical division of a computer network within which information can be transmitted for all devices to receive. VLANs enable you to segment a switched network so that devices in one VLAN do not receive information packets from devices in another VLAN.

- VLAN 400 is assigned to the second Gigabit Ethernet port and is used for client-side traffic.
- VLAN 500 is assigned to the third Gigabit Ethernet port and is used for server-side traffic.
- None of the three Gigabit Ethernet ports used are trunked.
- A management VLAN interface is configured for the Admin context with VLAN 1000 and IP address 172.25.91.110.
- A management VLAN interface is configured for the user context VC_web with VLAN 1000 and IP address 172.25.91.111.
- A client-side VLAN interface is configured for the user context VC_web with VLAN 400 and IP address 10.10.40.10.
- A server-side VLAN interface is configured for the user context VC_web with VLAN 500 and IP address 10.10.50.1.
- Four web servers are available to the ACE for load-balancing client requests.

Establishing a Console Connection on the ACE

The ACE has one standard RS-232 serial port on its rear panel that operates as the console port. You can establish a direct serial connection between the ACE and your terminal (or a PC with terminal software) by making a serial connection to this console port. The integrated serial port accepts a 9-pin female D-shell connector. Use a straight-through cable to connect the ACE to the terminal or a PC. For more instructions on connecting a console cable to your ACE appliance, see the *Cisco 4710 Application Control Engine Appliance Hardware Installation Guide*.

The ACE appliance has four physical Ethernet interface ports. All VLANs are assigned to these ports. The four Ethernet ports provide the physical connection between the ACE and the servers, PCs, routers, and other devices. You can configure the Ethernet ports to provide an interface for connecting to 10-Mbps, 100-Mbps, or 1000-Mbps networks. After the VLANs are assigned, you can configure the corresponding VLAN interfaces so that the ACE can provide different networking functions for different VLANs.

Note Only the Admin context is directly accessible through the console port; all other contexts can be accessed through Telnet or SSH sessions on the Ethernet ports.

After making the console connection, you can use any terminal communications application to access the ACE CLI.

Note If the appliance is not on, press the power button on the front of the ACE to start the boot process. See the *Cisco 4710 Application Control Engine Appliance Hardware Installation Guide* for details.

Access the ACE CLI using HyperTerminal for Windows by following these steps:

Step 1 Launch HyperTerminal.

The Connection Description window appears (Figure 2-2).

Figure 2-2 HyperTerminal—Connection Description

Step 2 Enter a name for your connection in the Name field.

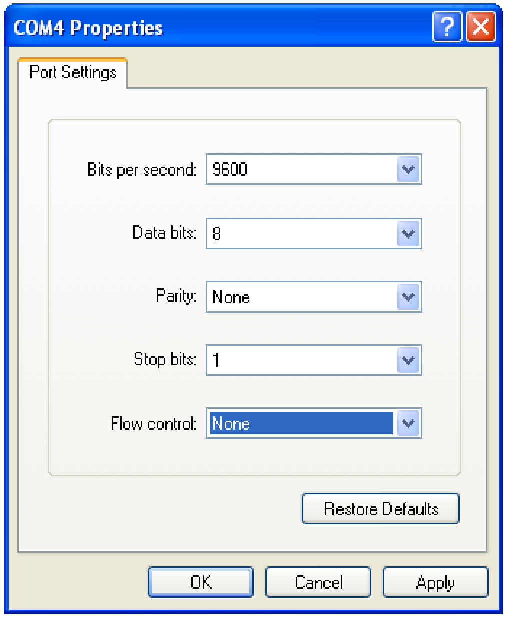
Step 3 Click **OK**. The Connect To window appears (Figure 2-3).

Figure 2-3 HyperTerminal—Connect To

Step 4 From the Connect using drop-down list, choose the COM port to which the device is connected.

Step 5 Click **OK**. The Port Properties window appears (Figure 2-4).

Figure 2-4 HyperTerminal—Port Properties



271857

Step 6 Set the port properties:

- Bits per second = 9600
- Data bits = 8
- Parity = none
- Stop bits = 1
- Flow control = None

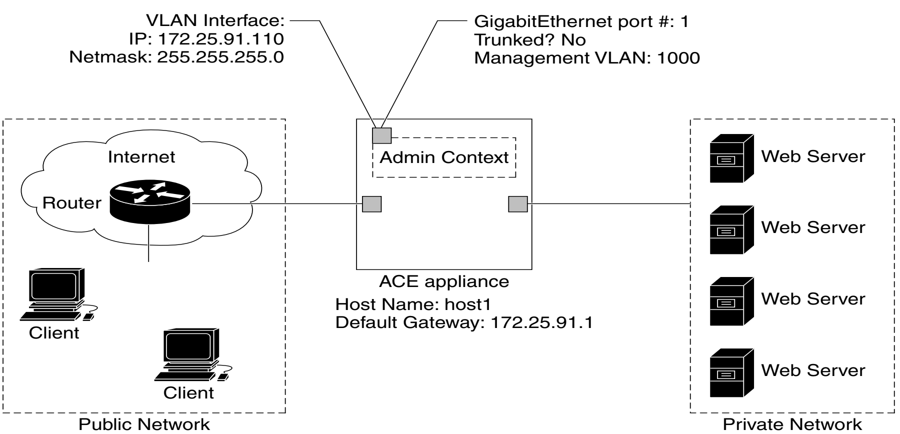
Step 7 Click **OK** to connect.

Enabling Management Connectivity Using the Setup Script

When you boot the ACE for the first time and the ACE does not detect a startup configuration file, a setup script guides you through the process of configuring a management VLAN on the ACE through one of its Gigabit Ethernet ports to enable connectivity to the Device Manager GUI.

After running the setup script, the management VLAN is allocated to the specified Gigabit Ethernet port and the VLAN interface is configured on the ACE, as illustrated in Figure 2-5.

Figure 2-5 Configuration After the Setup Script is Executed



Configure the ACE using the setup script by following these steps:

Step 1 At the login prompt, log into the ACE by entering the login username admin and password. By default, the username and password are admin. For example, enter:

```
Starting sysmgr processes.. Please wait...Done!!!

switch login: admin
Password: admin
```

Step 2 At the Enter the new password for “admin”: prompt, change the default Admin password. If you do not change the default Admin password, after you upgrade the ACE software you will only be able to log in to the ACE through the console port.

```
Enter the new password for "admin": xxxxx
Confirm the new password for "admin": xxxxx
admin user password successfully changed.
```

Step 3 At the Enter the new password for “www”: prompt, change the default www user password. If you do change the default www user password, the www user will be disabled and you will not be able to use Extensible Markup Language (XML) to remotely configure an ACE until you change the default www user password.

```
Enter the new password for "www": xxxxx
Confirm the new password for "www": xxxxx
www user password successfully changed.

This script will perform the configuration necessary for a user to
manage the ACE Appliance using the ACE Device Manager. The management
port is a designated Ethernet port which has access to the same
network as your management tools including the ACE Device Manager. You
will be prompted for the Port Number, IP Address, Netmask and Default
Route (optional).

Enter 'ctrl-c' at any time to quit the script
```

Caution At this point, you should consider whether you plan to configure the ACE using the Device Manager GUI or using the CLI. If you have a trunking network setup, or if your VLAN 1000 has been used, you should bypass the following setup script and use the CLI at “Setting Up an ACE Appliance Using the CLI.”

Step 4 At the “Would you like to enter the basic configuration dialog? (yes/no)” prompt, press **Enter** to continue the setup. To bypass setup and directly access the CLI, type **no**.

```
Would you like to enter the basic configuration dialog? (yes/no) [y]:
```

Note The ACE provides a default response in brackets [ ] for each question in the setup script. Accept the default response to a configuration prompt by pressing **Enter**.

Step 5 Select port 1 to carry management VLAN communication by pressing **Enter**.

```
Enter the Ethernet port number to be used as the management port
(1-4):? [1]:
```

Step 6 Assign an IP address for the management VLAN interface by entering **172.25.91.110**.

```
Enter the management port IP Address (n.n.n.n): [192.168.1.10]:
172.25.91.110
```

Step 7 Accept the default subnet mask for the management VLAN interface by pressing **Enter**.

```
Enter the management port Netmask(n.n.n.n): [255.255.255.0]:
```

Step 8 Assign the IP address of the gateway router (the next-hop address for this route) by entering **172.25.91.1**.

```
Enter the default route next hop IP Address (n.n.n.n) or <enter> to
skip this step: 172.25.91.1
```

Step 9 Examine the entered values.

```
Summary of entered values:

Management Port: 1
Ip address 172.25.91.110
Netmask: 255.255.255.0
Default Route: 172.25.91.1
```

Step 10 Review the configuration details by pressing **d**.

```
Submit the configuration including security settings to the ACE
Appliance? (yes/no/details): [y]:

interface gigabitEthernet 1/3
  switchport access vlan 1000
  no shut
access-list ALL extended permit ip any any
class-map type management
match-any remote_access
  match protocol xml-https any
```

```
  match protocol dm-telnet any
  match protocol icmp any
  match protocol telnet any
  match protocol ssh any
  match protocol http any
  match protocol https any
  match protocol snmp any
policy-map type management first-match remote_mgmt_allow_policy
  class remote_access
    permit
interface vlan 1000
  ip address 172.25.91.110 255.255.255.0
  access-group input ALL
  service-policy input remote_mgmt_allow_policy
  no shutdown
ssh key rsa
ip route 0.0.0.0 0.0.0.0 172.25.91.1
```

Step 11 Accept this configuration by pressing **Enter**; otherwise, press **n**.

```
Submit the configuration including security settings to the ACE
Appliance? (yes/no/details): [y]:
```

Step 12 After you select **y**, the following message appears.

```
Configuration successfully applied. You can now manage this ACE
Appliance by entering the url 'https://172.25.91.110' into a web
browser to access the Device Manager GUI.
```

After you have completed the setup script, the command prompt appears.

```
switch/Admin#
```

After you specify a Gigabit Ethernet port, port mode, and management VLAN, the setup script automatically applies the following default configuration:

- A Management VLAN is allocated to the specified Ethernet port.
- An extended IP access list that allows IP traffic originating from any other host addresses.
- A traffic classification is created for management protocols HTTP, HTTPS, ICMP, SSH, Telnet, and XML-HTTPS. HTTPS is dedicated to connectivity with the Device Manager GUI.
- A VLAN interface is configured on the ACE.

Assigning a Name to the ACE

The hostname is used for the command-line prompts and default configuration filenames. When you establish sessions to multiple devices, the hostname helps you keep track of which ACE you are entering commands to. By default, the hostname for the ACE is switch.

For example, change the hostname of the ACE from switch to host1 by entering:

```
switch/Admin# Config
switch/Admin(config)# hostname host1
```

The prompt appears with the new hostname.

```
host1/Admin(config)#
```

Setting Up an ACE Appliance Using the Device Manager GUI

You can set up an ACE appliance using the Device Manager GUI or the CLI. This section describes how to set up an ACE using the GUI, and includes the following topics:

- Logging in to the ACE
- Configuring a Second Gigabit Ethernet Interface Port
- Configuring a Third Gigabit Ethernet Interface Port

Logging in to the ACE

You can access the ACE Device Manager GUI through a web-based interface. Log in to the Device Manager by following these steps:

Step 1 Navigate to the ACE Device Manager by entering the secure HTTPS address or hostname of the ACE in the address field of a web browser. For the example setup shown earlier in Figure 2-1, enter:

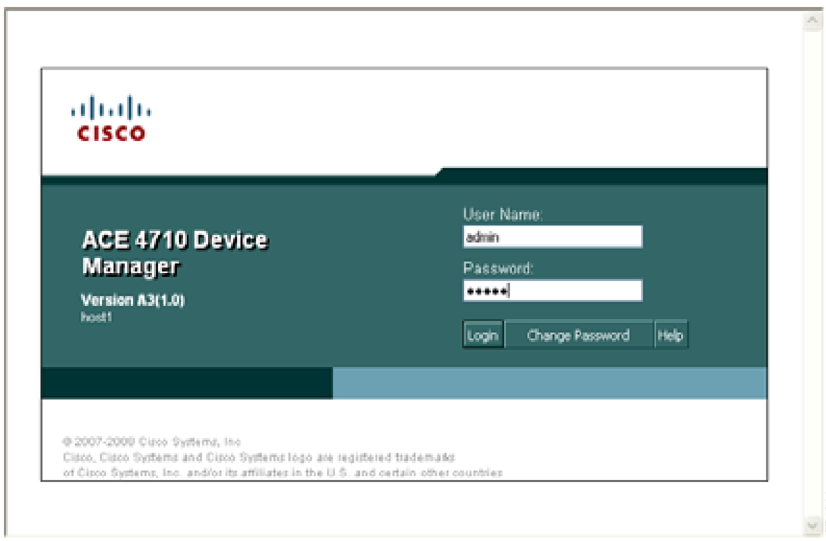
https://172.25.91.110/

Step 2 Click **Yes** at the prompt to accept (trust) and install the signed certificate from Cisco Systems, Inc. To avoid having to approve the signed certificate every time you log in to the Device Manager, accept the certificate.

The Device Manager GUI Login window appears (Figure 2-6).

Note Because this product is regularly updated, you may notice minor variations between the figures in this manual and the windows that appear in the software version you are running.

Figure 2-6 Device Manager GUI Login Window

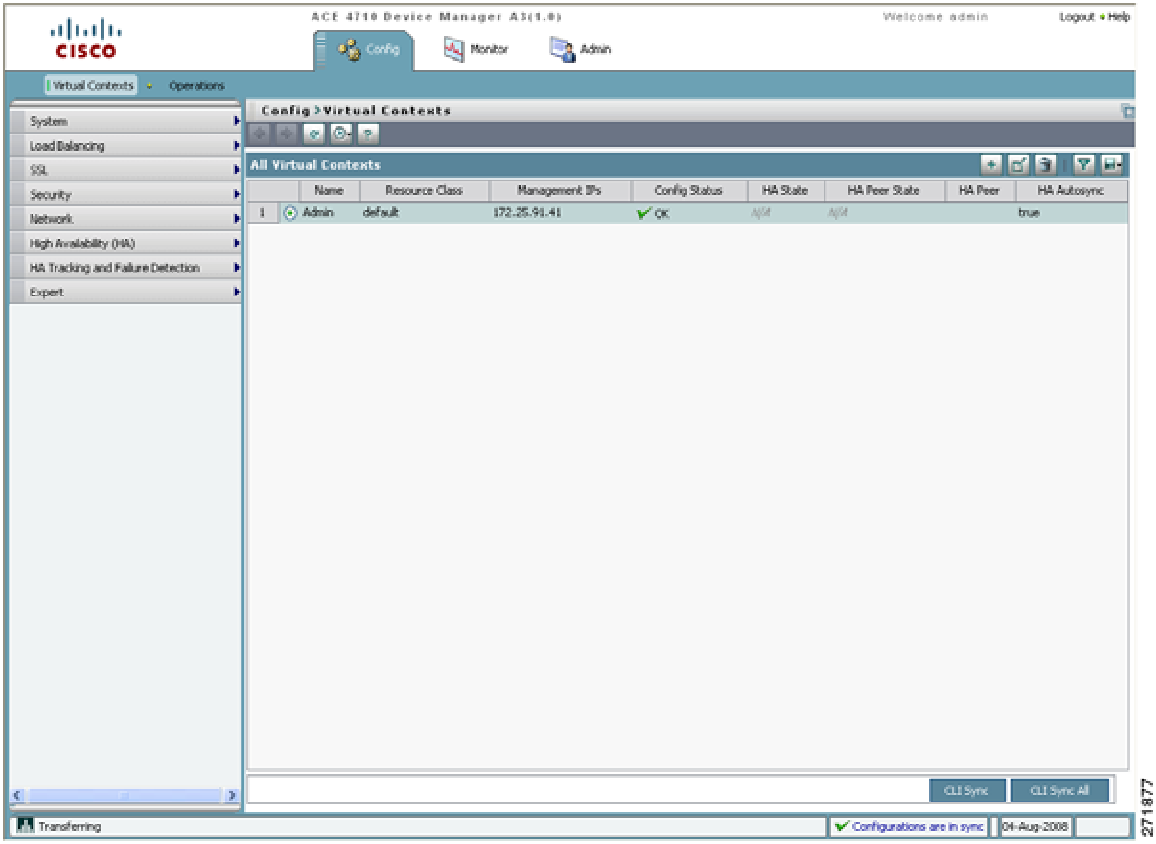


Step 3 In the User Name field, type **admin** for the admin user account.

Step 4 In the Password field, type the new password that you entered in Step 2 in "Enabling Management Connectivity Using the Setup Script."

Step 5 Click **Login**. The default window that appears is the Virtual Contexts window with the Admin context listed, as shown in Figure 2-7.

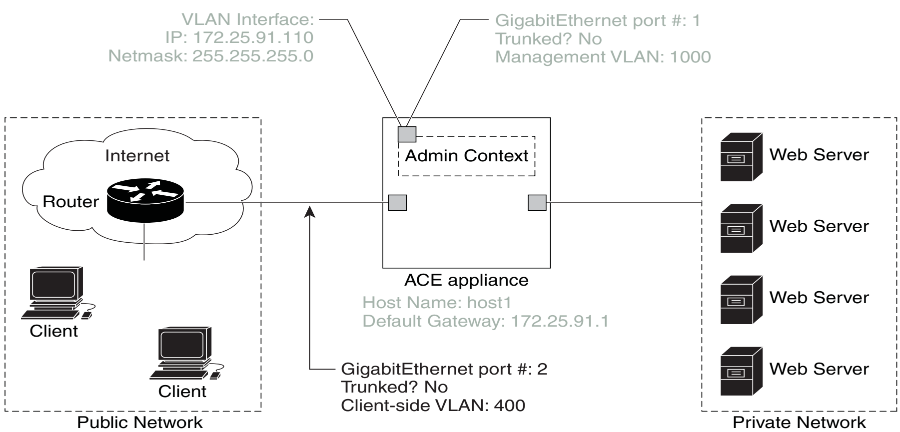
Figure 2-7 Virtual Contexts Pane (Admin Context)



Configuring a Second Gigabit Ethernet Interface Port

You can configure a second Gigabit Ethernet interface port to connect to clients. For the example configuration, you will configure Gigabit Ethernet interface port 2 as illustrated in Figure 2-8 (previously configured settings are grayed out).

Figure 2-8 Configuring a Second Gigabit Ethernet Interface Port to Connect to Clients

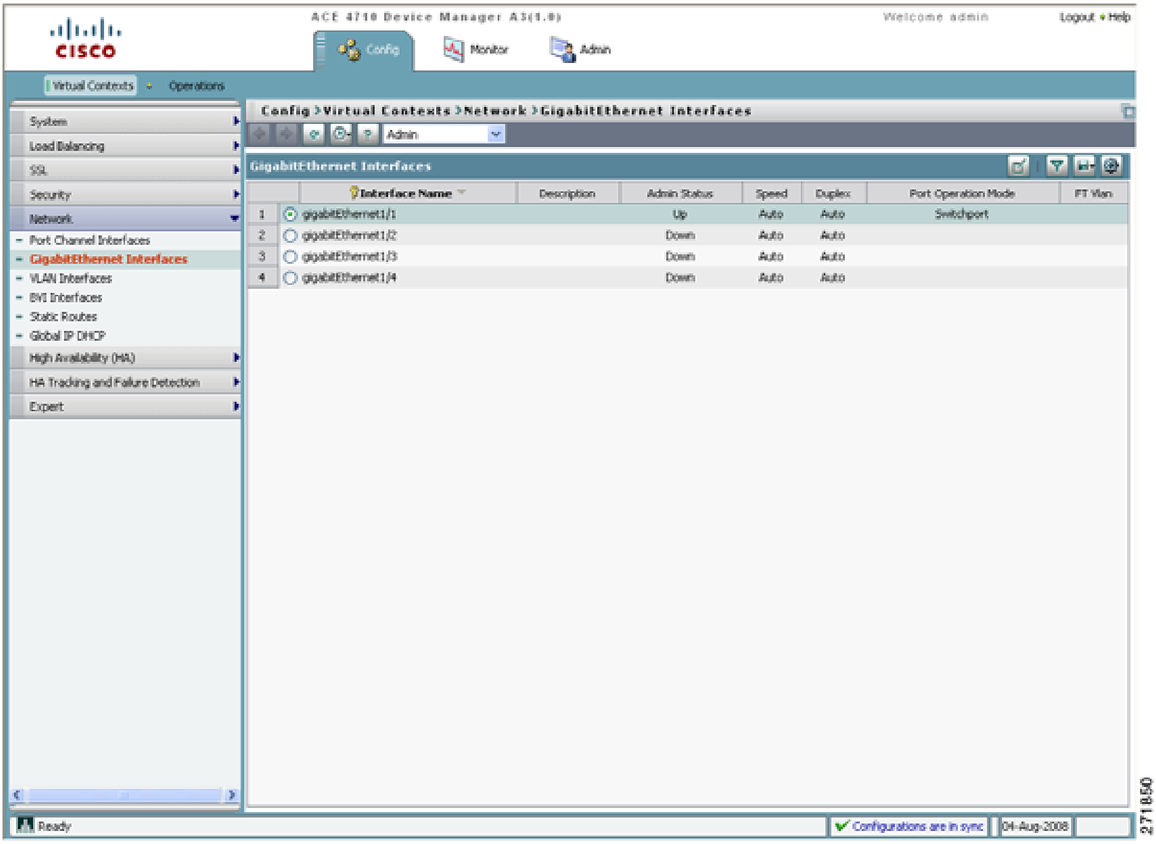


Configure a second Gigabit Ethernet port by following these steps:

Step 1 Choose **Config** > **Virtual Contexts** > **Network** > **GigabitEthernet Interfaces**. The GigabitEthernet Interfaces pane appears (Figure 2-9).

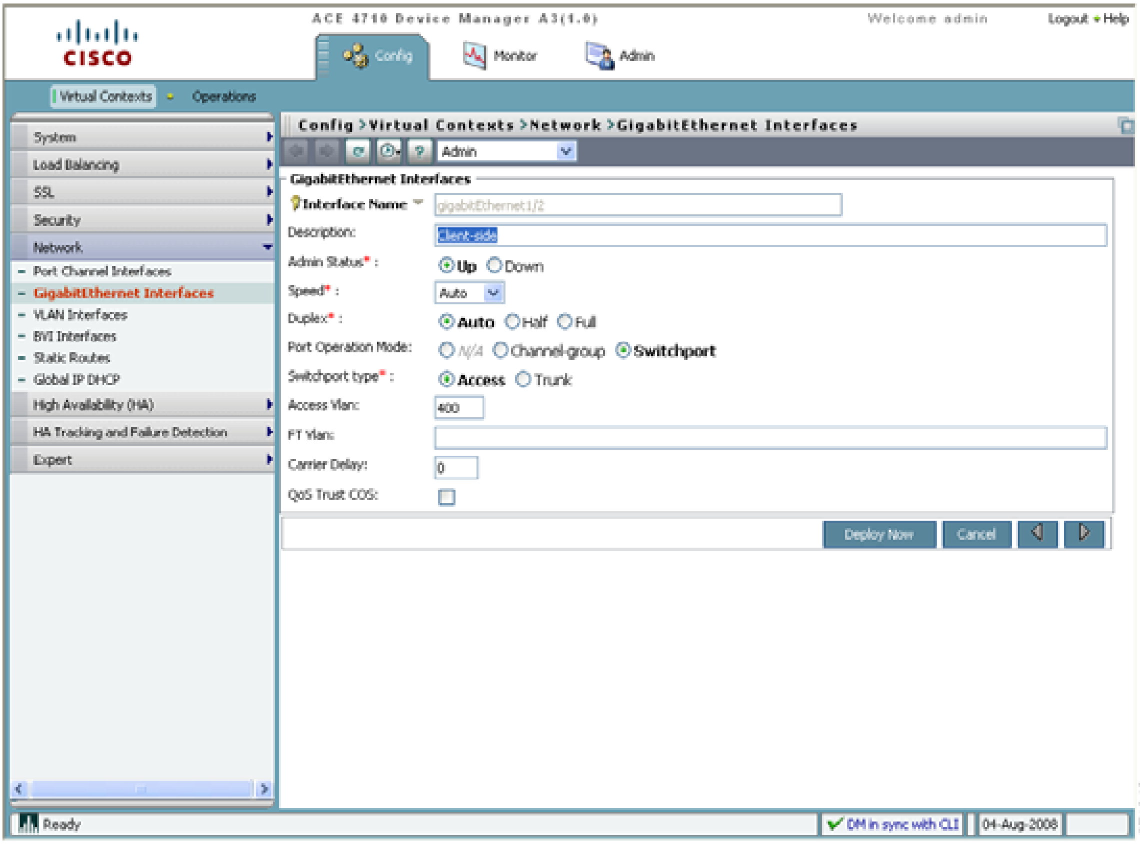
Note Only users authenticated in the Admin context can configure the Gigabit Ethernet interface ports.

Figure 2-9 GigabitEthernet Interfaces Pane—gigabitEthernet 1/2



Step 2 In the GigabitEthernet Interfaces pane, choose **gigabitEthernet 1/2**, and then click **Edit** to define attributes for the port. The GigabitEthernet Interfaces window appears (Figure 2-10).

Figure 2-10 GigabitEthernet Interfaces Window—gigabitEthernet 1/2

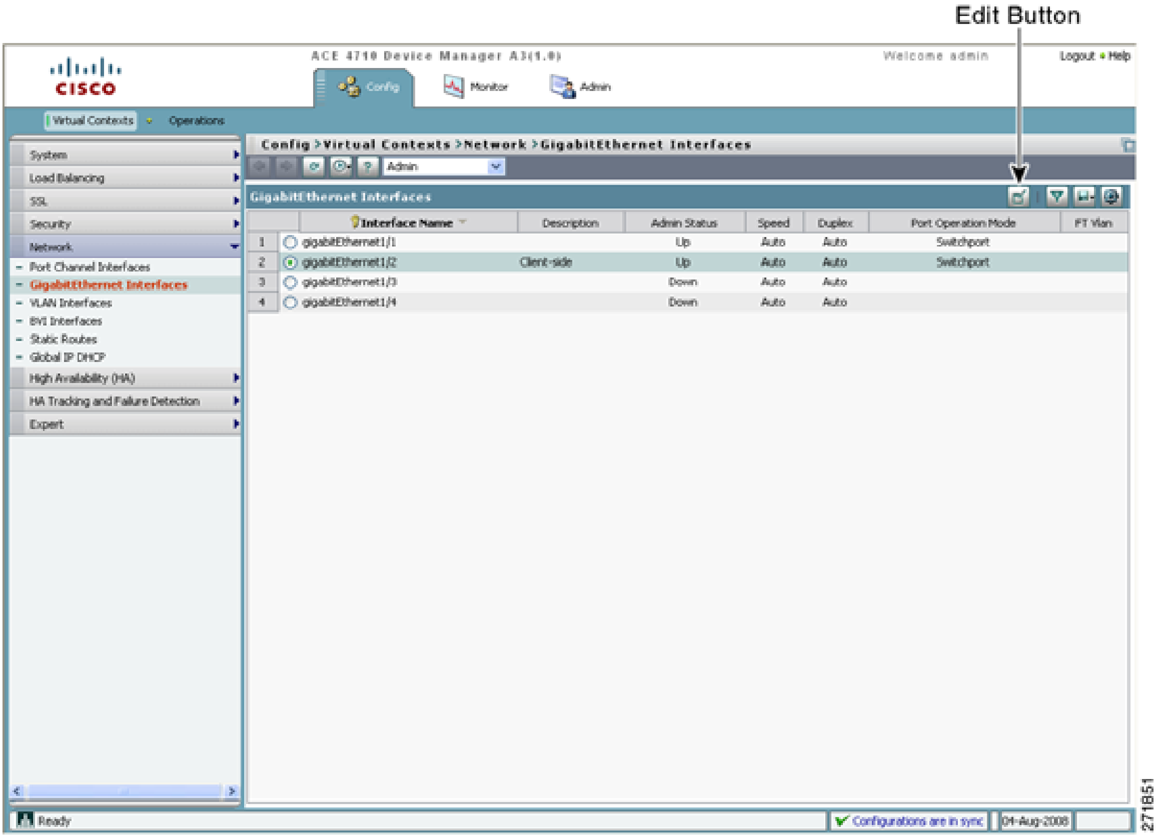


Step 3 Enter the following attributes for port 2. Leave the remaining attributes blank or with their default values.

- Admin Status: Up
- Speed: Auto
- Port Operation Mode: Switchport
- Switchport type: Access
- Access Vlan: 400

Step 4 Click **Deploy Now** to save these settings and to return to the GigabitEthernet Interfaces pane (Figure 2-11).

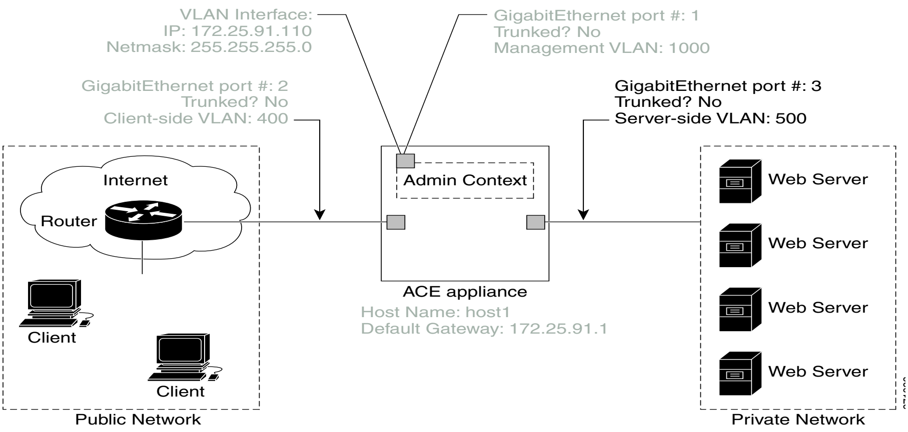
Figure 2-11 GigabitEthernet Interfaces Pane with Ethernet Port 2 Configured



Configuring a Third Gigabit Ethernet Interface Port

You can configure a third Gigabit Ethernet interface port to connect to the servers. For the example configuration, you will configure Gigabit Ethernet interface port 3 as illustrated in Figure 2-12 (previously configured settings are grayed out.)

Figure 2-12 Configuring a Third Gigabit Ethernet Interface Port to Connect to the Servers



Configure a third Gigabit Ethernet port by following these steps:

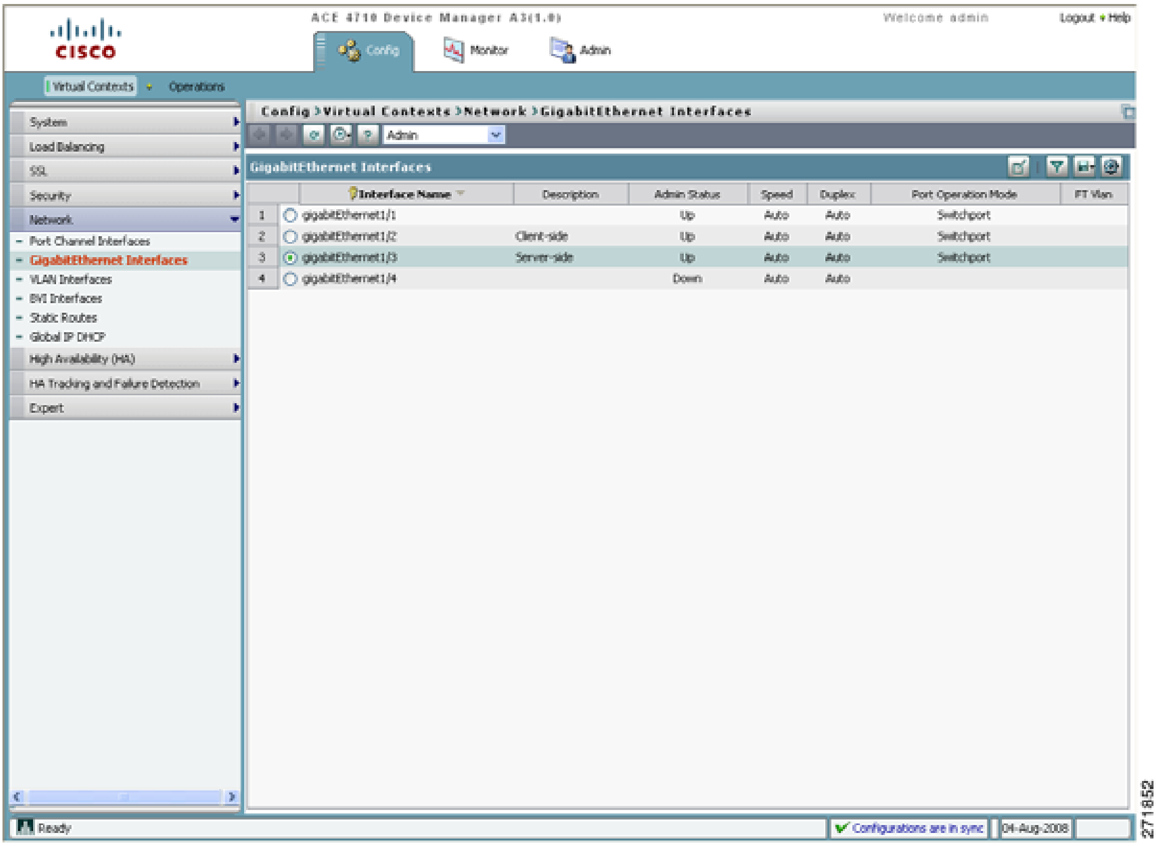
Step 1 In the GigabitEthernet Interfaces pane, choose **gigabitEthernet 1/3**, and then click **Edit** to define attributes for the port. The GigabitEthernet Interfaces window appears (Figure 2-10).

Step 2 Enter the following attributes for port 3. Leave the remaining attributes blank or with their default values.

- Admin Status: Up
- Speed: Auto
- Port Operation Mode: Switchport
- Switchport type: Access
- Access VLAN: 500

Step 3 Click **Deploy Now** to save these settings and to return to the GigabitEthernet Interfaces pane (Figure 2-13).

Figure 2-13 GigabitEthernet Interfaces Pane with Ethernet Port 3 Configured



Setting Up an ACE Appliance Using the CLI

You can set up an ACE appliance using the Device Manager GUI or the CLI. This section describes how to set up an ACE using the CLI, and includes the following topics:

Logging in to the ACE

After you have established a direct serial connection between the ACE and your terminal or a PC (see the "Establishing a Console Connection on the ACE" section), you can set up the ACE using the CLI.

When the setup script displays the "Would you like to enter the basic configuration dialog? (yes/no):" prompt, enter **no** to access the CLI. Log in to the ACE by following these steps:

Step 1 At the login prompt, enter **admin**. For the password, type the new password that you entered in Step 2 in the "Enabling Management Connectivity Using the Setup Script" section.

```
host1 login: admin
Password: xxxxx
```

You are ready to use the ACE CLI when the following prompt appears.

```
host1/Admin#
```

Step 2 Set the **terminal session-timeout** command to 0 to prevent this current session from timing out. By default, a session on the ACE is automatically logged out after 5 minutes of inactivity.

```
host1/Admin# terminal session-timeout 0
host1/Admin#
```

Configuring the First Gigabit Ethernet Port

You can configure a Gigabit Ethernet interface port for the ACE management traffic. For the example configuration, you will configure Gigabit Ethernet interface port 1. Configure the first Gigabit Ethernet port by following theses steps:

Step 1 Configure a Layer 2 Gigabit Ethernet port on the ACE by using the **interface gigabitEthernet** *slot_number/port_number* command in configuration mode.

Note The slot_number specifies the physical slot on the ACE that contains the Ethernet ports. For the current release of the ACE appliance, this selection is always 1.

Configure Gigabit Ethernet port 1 and enter interface configuration mode by entering:

```
host1/Admin# config
host1/Admin(config)# interface gigabitEthernet 1/1
host1/Admin(config-if)#
```

Step 2 Enable the Gigabit Ethernet port by using the **no shutdown** command in interface configuration mode. Disable a running Gigabit Ethernet port by using the **shutdown** command; bring one up by using the **no shutdown** command.

```
host1/Admin(config-if)# no shutdown
```

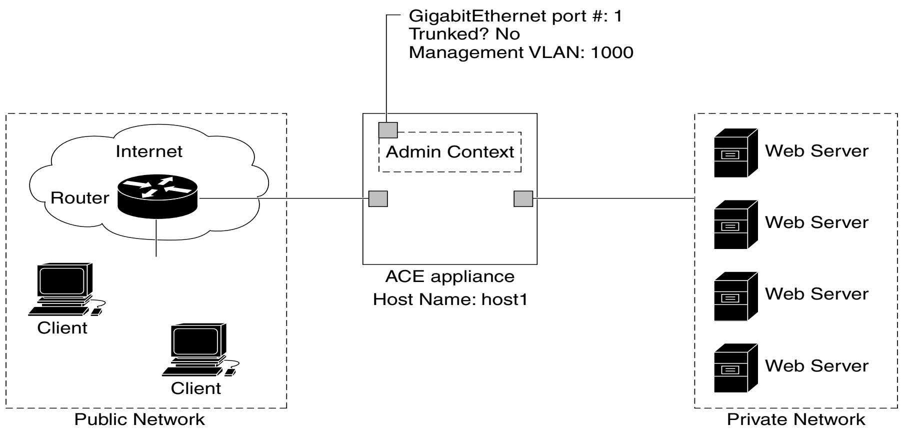
Step 3 Display the configuration of the interface by using the **do** command with the **show interface** command.

```
host1/admin(config-if)# do show interface vlan 1000
```

Allocating the First Gigabit Ethernet Port to a VLAN

After you configure an Gigabit Ethernet port, the next step is to allocate it to a VLAN. For the example configuration, you will allocate the first Gigabit Ethernet port to VLAN 1000, as illustrated in Figure 2-14 (previously configured settings are grayed out.)

Figure 2-14 Allocating the First Gigabit Ethernet Port to a VLAN



Allocate the port to a VLAN by following these steps:

Step 1 Assign one or more VLAN numbers to the Gigabit Ethernet port by using the **switchport trunk allowed vlan** *vlan_list* command in interface configuration mode. The vlan_list argument can include:

- A single VLAN number
- Beginning and ending VLAN numbers separated by a hyphen
- Specific VLAN numbers separated by commas

Valid entries are 1 through 4094. Do not enter any spaces in a hyphenated range or in a comma-separated list of numbers in the vlan_list argument.

Note You can associate a VLAN number with only one Gigabit Ethernet port.

Add VLAN 1000 to the defined list of VLANs currently set for Gigabit Ethernet port 1 by entering:

```
host1/Admin(config)# interface gigabitEthernet 1/1
host1/Admin(config-if)# switchport access allowed vlan 1000
```

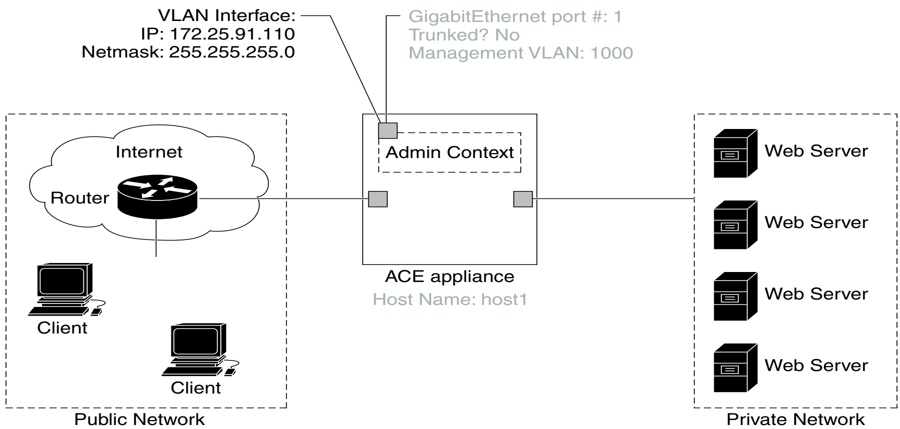
Step 2 Enable VLAN access for the specified Layer 2 Gigabit Ethernet port by using the **no shutdown** command in interface configuration mode.

```
host1/Admin(config-if)# no shutdown
host1/Admin(config-if)# exit
host1/Admin(config)#
```

Configuring a Management VLAN Interface on the ACE

You can provide management connectivity to the ACE by assigning an IP address to the VLAN interface on the ACE. For the example configuration, you will assign an IP address 172.25.91.110 and a subnet mask of 255.255.255.0 to VLAN 1000, as illustrated in Figure 2-15 (previously configured settings are grayed out).

Figure 2-15 Configuring a Management VLAN Interface on the ACE



Configure a VLAN interface on the ACE by following these steps:

Step 1 Access interface configuration mode for the VLAN 1000.

```
host1/Admin(config)# interface vlan 1000
host1/Admin(config-if)#
```

Step 2 Assign an IP address of 172.25.91.110 and a subnet mask of 255.255.255.0 to the VLAN interface for management connectivity.

```
host1/Admin(config-if)# ip address 172.25.91.110 255.255.255.0
```

Step 3 (Optional) Provide a description for the interface.

```
host1/Admin(config-if)# description Management connectivity on VLAN
1000
```

Step 4 Enable the VLAN interface.

```
host1/Admin(config-if)# no shutdown
```

Step 5 Display the configuration of VLAN 1000.

```
host1/Admin(config-if)# do show interface vlan 1000
```

Step 6 Verify network connectivity by using the **ping** command. This command verifies the connectivity of a remote host or server by sending echo messages from the ACE.

```
host1/Admin(config-if)# do ping 172.25.91.110
```

Step 7 Exit the interface configuration mode.

```
host1/Admin(config-if)# exit
host1/Admin(config)#
```

Configuring a Second Gigabit Ethernet Interface Port

You can configure a second Gigabit Ethernet interface port to connect to clients. For the example configuration, you will configure Gigabit Ethernet interface port 2 as illustrated in Figure 2-8. Configure the second Gigabit Ethernet Interface port by following these steps:

Step 1 Add VLAN 400 to the defined list of VLANs currently set for Gigabit Ethernet port 2.

```
host1/Admin(config)# interface gigabitEthernet 1/2
host1/Admin(config-if)# switchport access vlan 400
```

Step 2 Enable the Gigabit Ethernet port.

```
host1/Admin(config-if)# no shutdown
host1/Admin(config-if)# exit
host1/admin(config)#
```

Configuring a Third Gigabit Ethernet Interface Port

You can configure a third Gigabit Ethernet interface port to connect to the servers. For the example configuration, you will configure Gigabit Ethernet interface port 3 as illustrated in Figure 2-12. Configure the third Gigabit Ethernet Interface port by following these steps:

Step 1 Add VLAN 500 to the defined list of VLANs currently set for Gigabit Ethernet port 3.

```
host1/Admin(config)# interface gigabitEthernet 1/3
host1/Admin(config-if)# switchport access allowed vlan 500
```

Step 2 Enable the Ethernet port.

```
host1/Admin(config-if)# no shutdown
host1/Admin(config-if)# exit
host1/admin(config)#
```

Configuring Remote Management Access to the ACE

Before remote network access can occur on the ACE through an Ethernet port, you must create a traffic policy that identifies the network management traffic that can be received by the ACE. Configure remote management access to the ACE by following these steps:

Step 1 Create a management-type class map named REMOTE_ACCESS that matches any traffic.

```
host1/Admin(config)# class-map type management match-any REMOTE_ACCESS
host1/Admin(config-cmap-mgmt)#
```

Step 2 (Optional) Provide a description for the class map.

```
host1/Admin(config-cmap-mgmt)# description Remote access traffic match
```

Step 3 Configure the match protocol to permit traffic based on the SSH, Telnet, and ICMP protocols for any source address.

```
host1/Admin(config-cmap-mgmt)# match protocol ssh any
host1/Admin(config-cmap-mgmt)# match protocol telnet any
host1/Admin(config-cmap-mgmt)# match protocol icmp any
host1/Admin(config-cmap-mgmt)# exit
host1/Admin(config)#
```

Step 4 Create a REMOTE_MGMT_ALLOW_POLICY policy map for traffic destined to an ACE interface.

```
host1/Admin(config)# policy-map type management first-match
REMOTE_MGMT_ALLOW_POLICY
host1/Admin(config-pmap-mgmt)#
```

Step 5 Apply the previously created REMOTE_ACCESS class map to this policy.

```
host1/Admin(config-pmap-mgmt)# class REMOTE_ACCESS
host1/Admin(config-pmap-mgmt-c)#
```

Step 6 Allow the ACE to receive the configured class map management protocols.

```
host1/Admin(config-pmap-mgmt-c)# permit
host1/Admin(config-pmap-mgmt-c)# exit
host1/Admin(config-pmap-mgmt)# exit
host1/Admin(config)#
```

Step 7 Access interface configuration mode for the VLAN to which you want to apply the policy map.

```
host1/Admin(config)# interface vlan 1000
host1/Admin(config-if)#
```

Step 8 Apply the REMOTE_MGMT_ALLOW_POLICY policy map to the interface.

```
host1/Admin(config-if)# service-policy input REMOTE_MGMT_ALLOW_POLICY
```

Step 9 Display the REMOTE_MGMT_ALLOW_POLICY policy applied to the interface.

```
host1/Admin(config-if)# do show service-policy
REMOTE_MGMT_ALLOW_POLICY

Status     : ACTIVE
-----------------------------------------
Interface: vlan 1000
  service-policy: REMOTE_MGMT_ALLOW_POLICY
```

Step 10 Save your configuration changes from the running configuration to the startup configuration.

```
host1/Admin(config-if)# do copy running-config startup-config

Generating configuration....
running config of context VC_web saved

host1/Admin(config-if)# exit
host1/Admin(config)# exit
```

Step 11 Display the running configuration.

```
host1/Admin(config)# do show running-config

Generating configuration....

class-map type management match-any REMOTE_ACCESS
  description Remote access traffic match
  2 match protocol telnet any
  3 match protocol ssh any
  4 match protocol icmp any

policy-map type management first-match REMOTE_MGMT_ALLOW_POLICY
  class REMOTE_ACCESS
    permit

interface vlan 1000
  description Management connectivity on VLAN 1000
  ip address 172.25.91.110 255.255.255.0
  service-policy input REMOTE_MGMT_ALLOW_POLICY
  no shutdown
interface vlan 400
  description client connectivity on VLAN 400
  ip address 10.10.40.10 255.255.255.0
  no shutdown
```

Accessing the ACE through a Telnet Session

After you have completed the previous configurations, you can use Telnet to access the ACE through an Ethernet port by using its IP address. Access the ACE through Telnet by following these steps:

Step 1 Initiate a Telnet session from a remote host to the ACE. For example, access the ACE from the VLAN IP address of 172.25.91.110 by entering:

```
remote_host# telnet 172.25.91.110

Trying 172.25.91.110... Open
```

Step 2 At the prompt, log in to the ACE. Enter **admin** as the user name and for the password, type the new password that you entered in the Step 2 in "Enabling Management Connectivity Using the Setup Script" section.

```
host1 login: admin
Password: xxxxx
```

Step 3 Display the Telnet session.

```
host1/Admin# show telnet
```

In this chapter, you have set up your ACE appliance so that you can use the ACE Device Manager or CLI to perform server load-balancing configuration tasks through a remote management interface. Next, you will create a user context for server load balancing.

CHAPTER 3

Creating a Virtual Context

This chapter describes how to create a virtual context for the Cisco 4700 Series Application Control Engine (ACE) appliance.

This chapter contains the following sections:

- Overview
- Creating a Virtual Context Using the Device Manager GUI
- Creating a Virtual Context Using the CLI

Overview

After reading this chapter, you should have a basic understanding of ACE appliance virtualization and be able to partition your ACE into multiple virtual devices or virtual contexts (VCs) for more efficient operation.

Virtualization allows you to create a virtual environment in which a single ACE is partitioned into multiple virtual devices, each functioning as an independent ACE appliance that is configured and managed independently.

You set up virtualization by performing the following configuration steps:

- Configure resource allocation for a virtual context
- Create a virtual context
- Configure access to the virtual context

An example virtual environment will be used throughout this guide, with the user context VC_web, for the web traffic through the network. This user context will be associated with the custom resource class RS_web.

In this chapter, you will create a virtual context. In subsequent chapters, you will create a virtual server within the virtual context. The virtual server is associated with a server farm and real servers. The example setup is illustrated in Table 3-1.

Table 3-1 Example Virtual Contexts

Virtual Context	Virtual Server	Server Farm	Real Servers
VC_web	VS_web	SF_web	RS_web1
			RS_web2
			RS_web3
			RS_web4

Before you begin configuring your ACE for virtualization, you should become familiar with a few concepts: virtual context, Admin and user contexts, and resource classes.

With ACE virtualization, you can create a virtual environment, called a virtual context, in which a single ACE appears as multiple virtual devices, each configured and managed independently. A virtual context allows you to closely and efficiently manage system resources, ACE users, and the services that you provide to your customers.

By default, the ACE initially provides you an Admin context, with the ability to define up to five user contexts. (With additional licenses, you can define up to 20 contexts.)

As the system administrator, you have full system administrator access to configure and manage the Admin context and all user contexts. Each context can also have its own administrator and log-in mechanism that provides access only to the specific context. When you log in to the ACE using the console or Telnet, you are authenticated in the Admin context.

Although virtualization allows you to create multiple contexts, in the physical world, you still have a single ACE with finite resources, such as the number of concurrent connections. To address this limitation, the ACE provides resource classes that allow you to manage each virtual context's access to physical ACE resources. A resource class is a definition of what portion of an ACE's overall resources will be assigned, at a minimum or maximum, to any given context. One resource class may be associated with one or more contexts.

The ACE is preconfigured with a default resource class for the Admin context. This default resource class is applied to all virtual contexts that you create. It allows a maximum of 100 percent access to all resources by all virtual contexts. When a resource is being used to its maximum limit, the ACE will deny additional requests for that resource from any other virtual contexts. To avoid oversubscribing resources and to help guarantee that resource availability is shared among multiple virtual contexts, you create custom resource classes and associate them with the virtual contexts you define.

Creating a Virtual Context Using the Device Manager GUI

This section describes how to create and configure a virtual context for server load balancing using the ACE Device Manager user interface and contains the following topics:

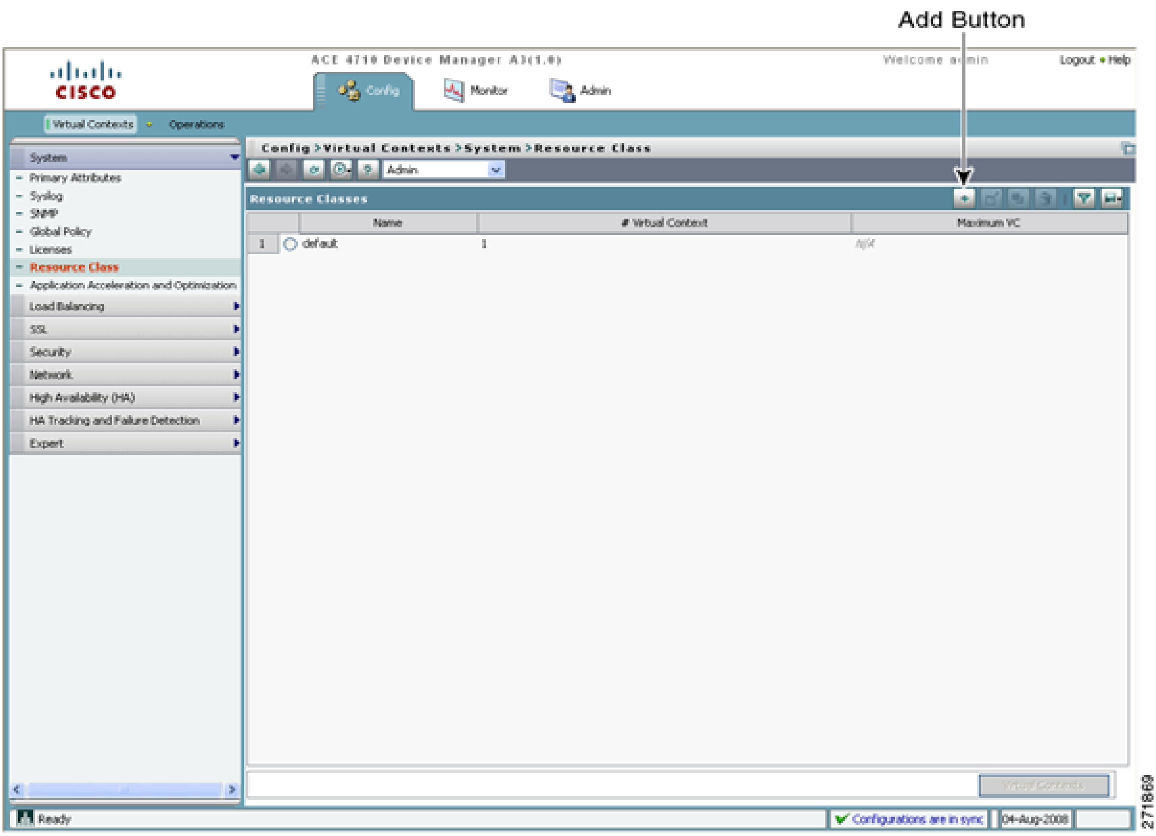
- Creating a Resource Class
- Creating a Virtual Context
- Configuring the Client-Side VLAN Interface
- Configuring the Server-Side VLAN Interface

Creating a Resource Class

Create a resource class by following these steps:

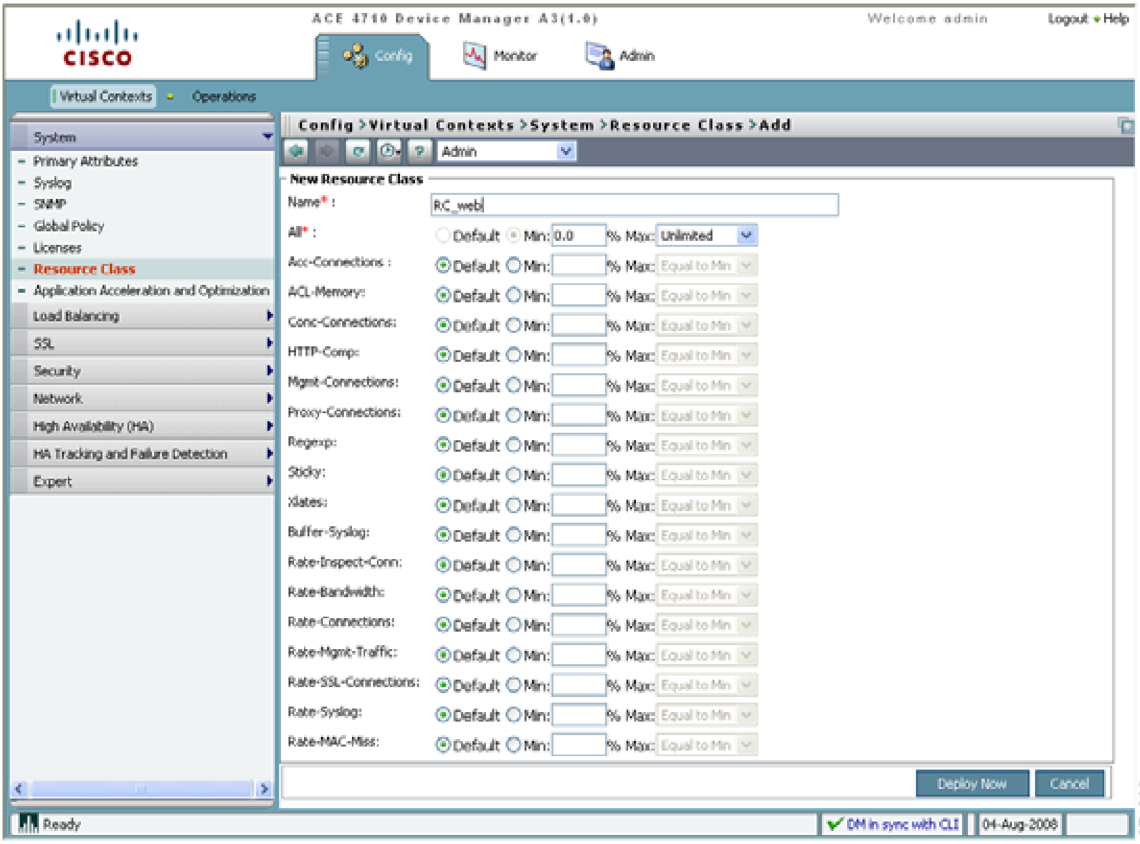
Step 1 Choose **Config** > **Virtual Contexts** > **System** > **Resource Class**. The Resource Classes pane appears.

Figure 3-1 Resource Classes Pane



Step 2 Click **Add**. The New Resource Class window appears (Figure 3-2).

Figure 3-2 New Resource Class Window

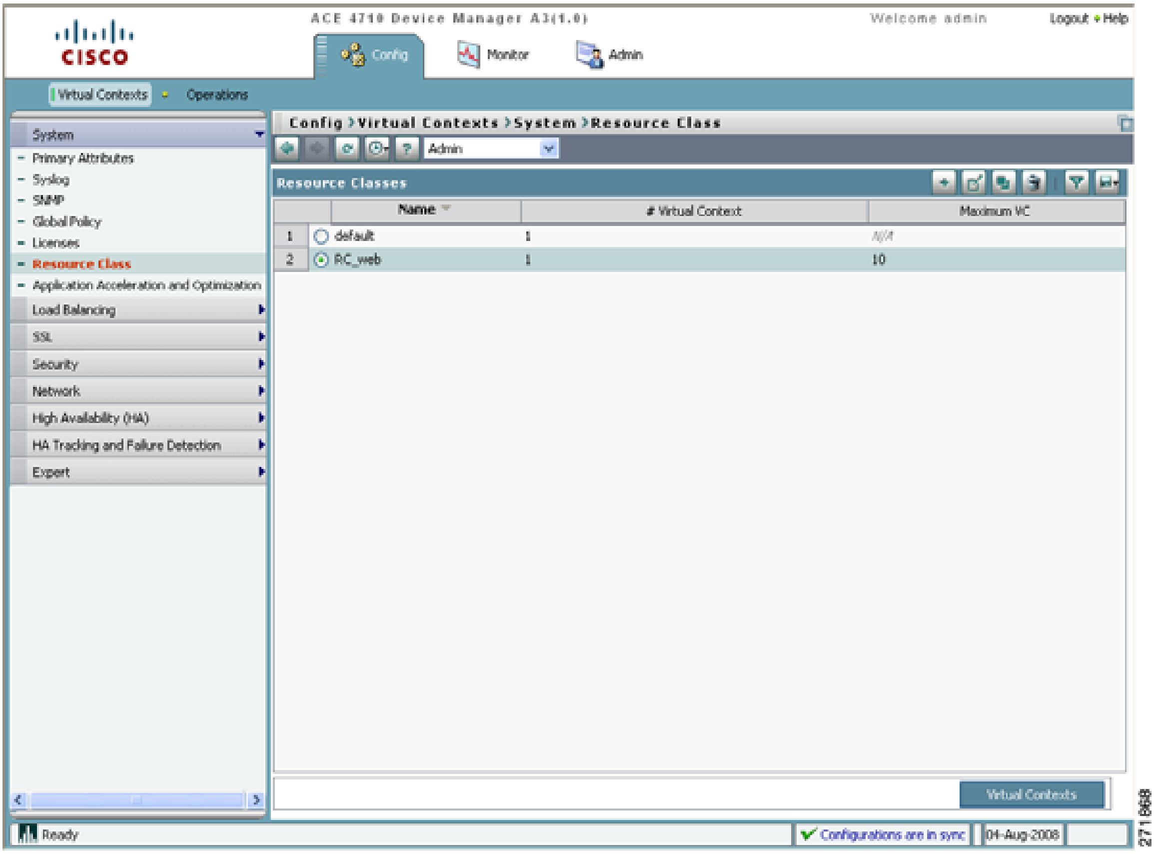


Step 3 Enter the following Resource Class attributes. Leave the remaining attributes blank or with their default values.

- Name: RC_web
- Default Min: 10
- Default Max: Unlimited

Step 4 Click **Deploy Now**. The Resource Classes pane appears with the newly added resource class (Figure 3-3).

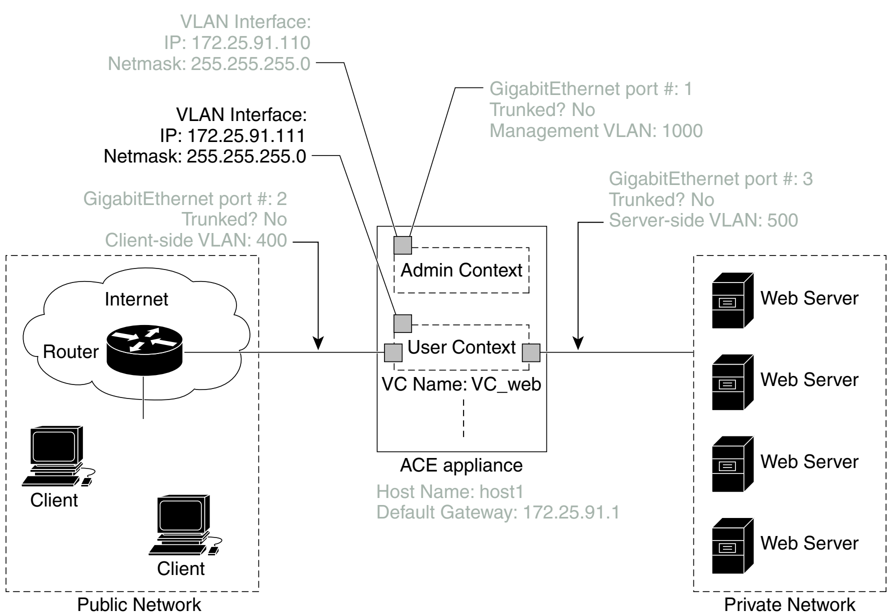
Figure 3-3 Resource Classes Pane with a New Resource Class Added



Creating a Virtual Context

You can create a user context for server load-balancing purposes. For the example configuration, you will create a user context, VC_web, and configure a management VLAN interface to VLAN 1000, as illustrated in Figure 3-4 (previously configured settings are grayed out).

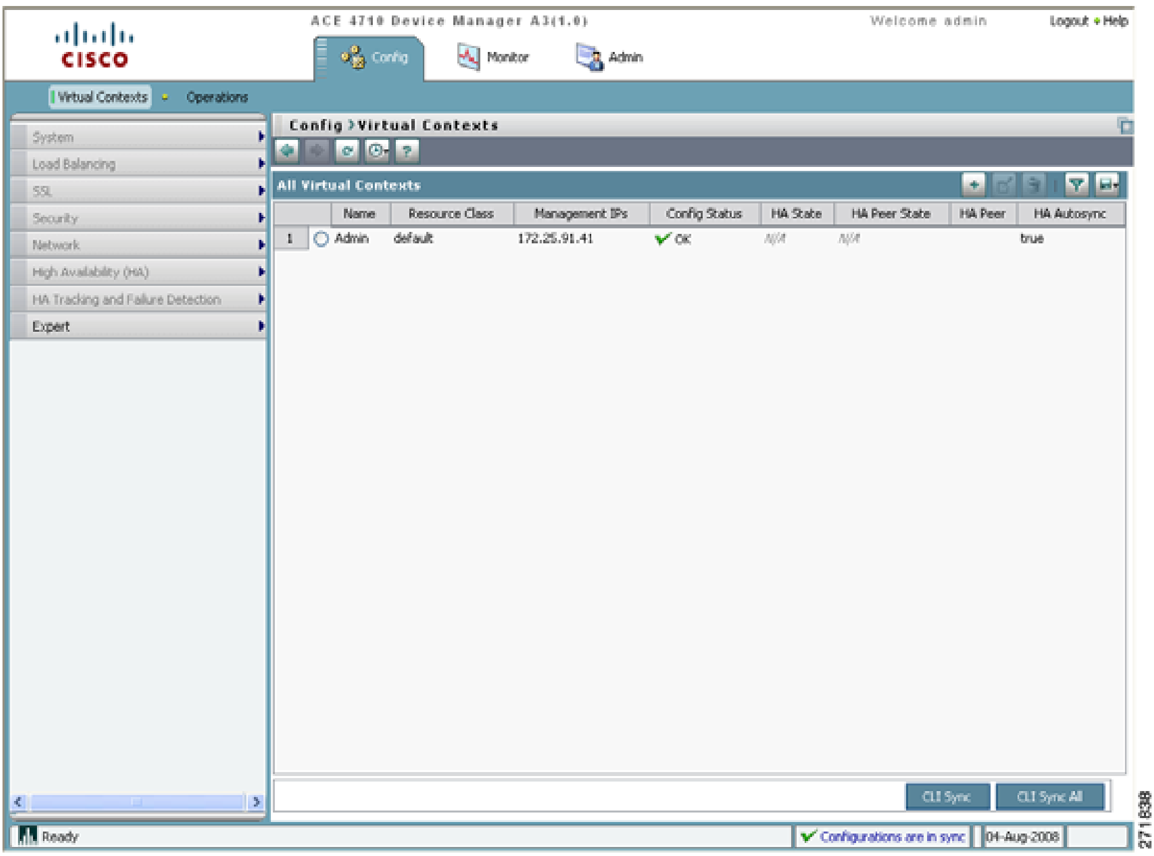
Figure 3-4 Creating a User Context



Create a virtual context by following these steps:

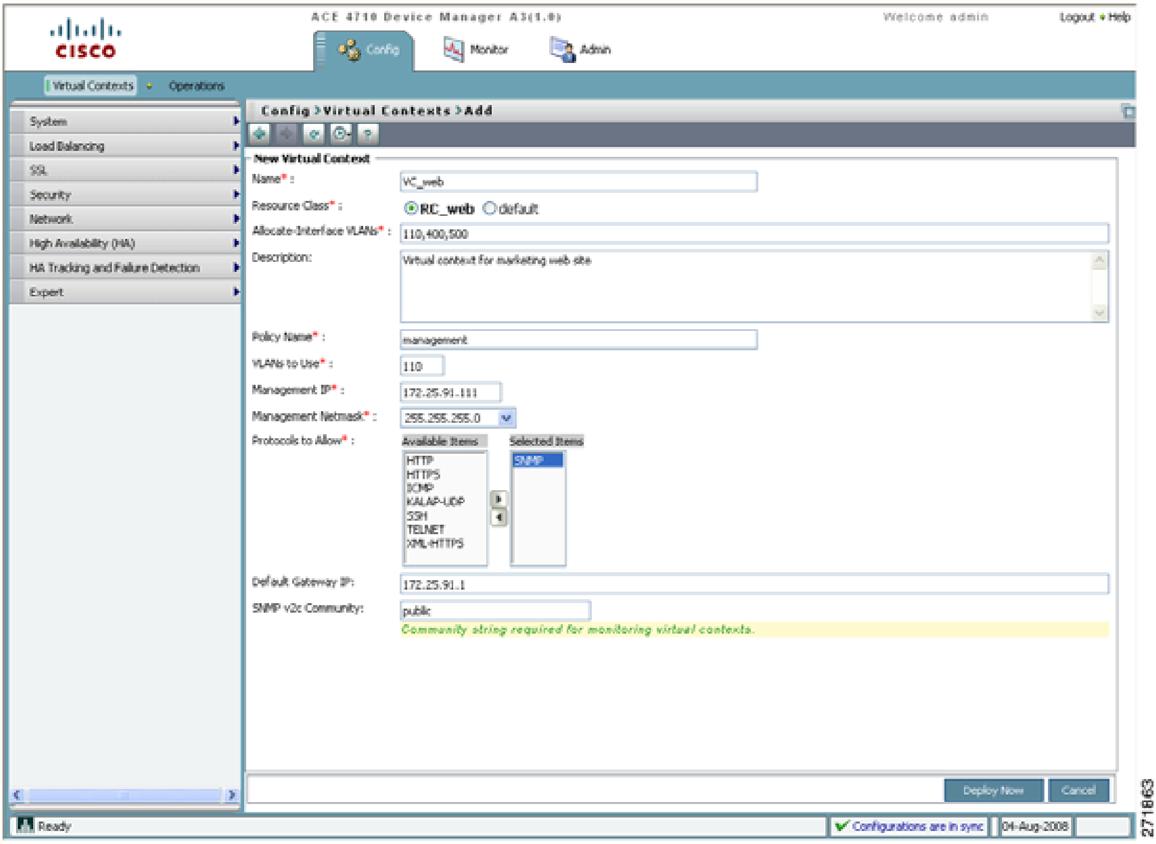
Step 1 Choose **Config** > **Virtual Contexts**. The All Virtual Contexts pane appears (Figure 3-5).

Figure 3-5 All Virtual Contexts Pane



Step 2 Click **Add**. The New Virtual Context window appears (Figure 3-6).

Figure 3-6 New Virtual Context Window



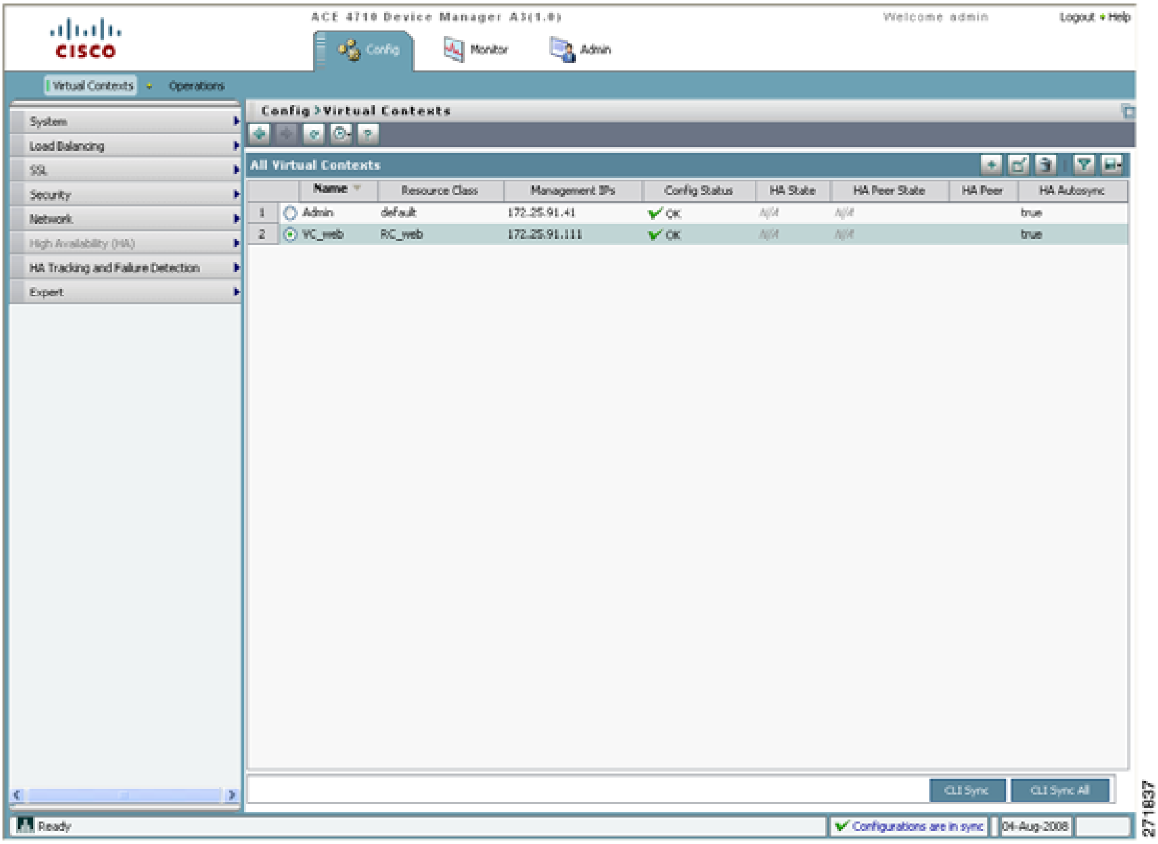
Step 3 Enter the following virtual context attributes. Leave the remaining attributes blank or with their default values.

- Name: VC_web
- Resource Class: RC_web
- Allocate-Interface VLANs: 1000, 400, 500 (these VLANs allow the context to receive the associated traffic)
- Description: Virtual context for marketing website
- Policy Name: Management

- VLANs to Use: 1000 (this VLAN allows for remote management of the context)
- Management IP: 172.25.91.111 (this IP address also allows for remote management of the context)
- Management Netmask: 255.255.255.0
- Protocols to Allow: SNMP (or any protocols that you allow for this virtual context)
- Default Gateway IP: 172.25.91.1

Step 4 Click **Deploy Now** to deploy this context. Then, choose **Virtual Contexts**. The window refreshes with the new virtual context listed in the All Virtual Contexts pane (Figure 3-7).

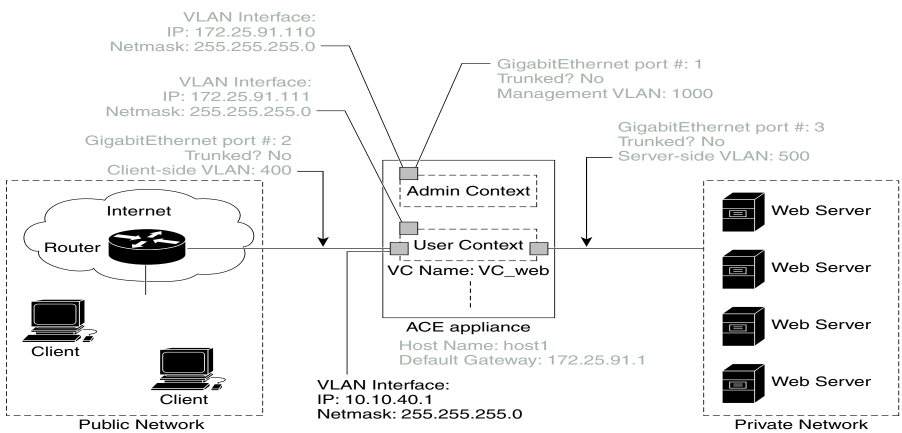
Figure 3-7 All Virtual Contexts Pane After VC_web is Added



Configuring the Client-Side VLAN Interface

You can now configure a client-side VLAN interface, which is the address to which client traffic is sent. For the example configuration, you will configure VLAN 400 (Figure 3-8).

Figure 3-8 Configuring the Client-Side VLAN Interface

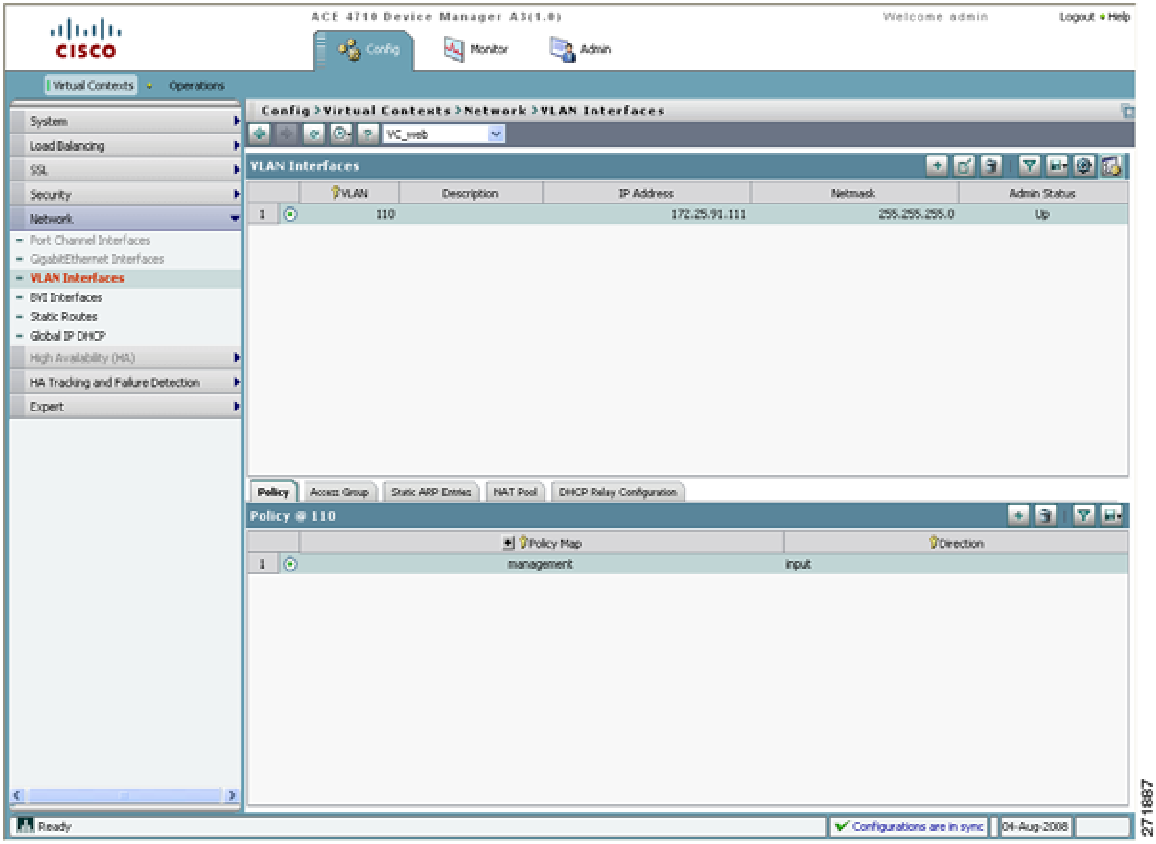


Configure a client-side VLAN interface by following these steps:

Step 1 Choose **VC_web** in the virtual contexts drop-down list.

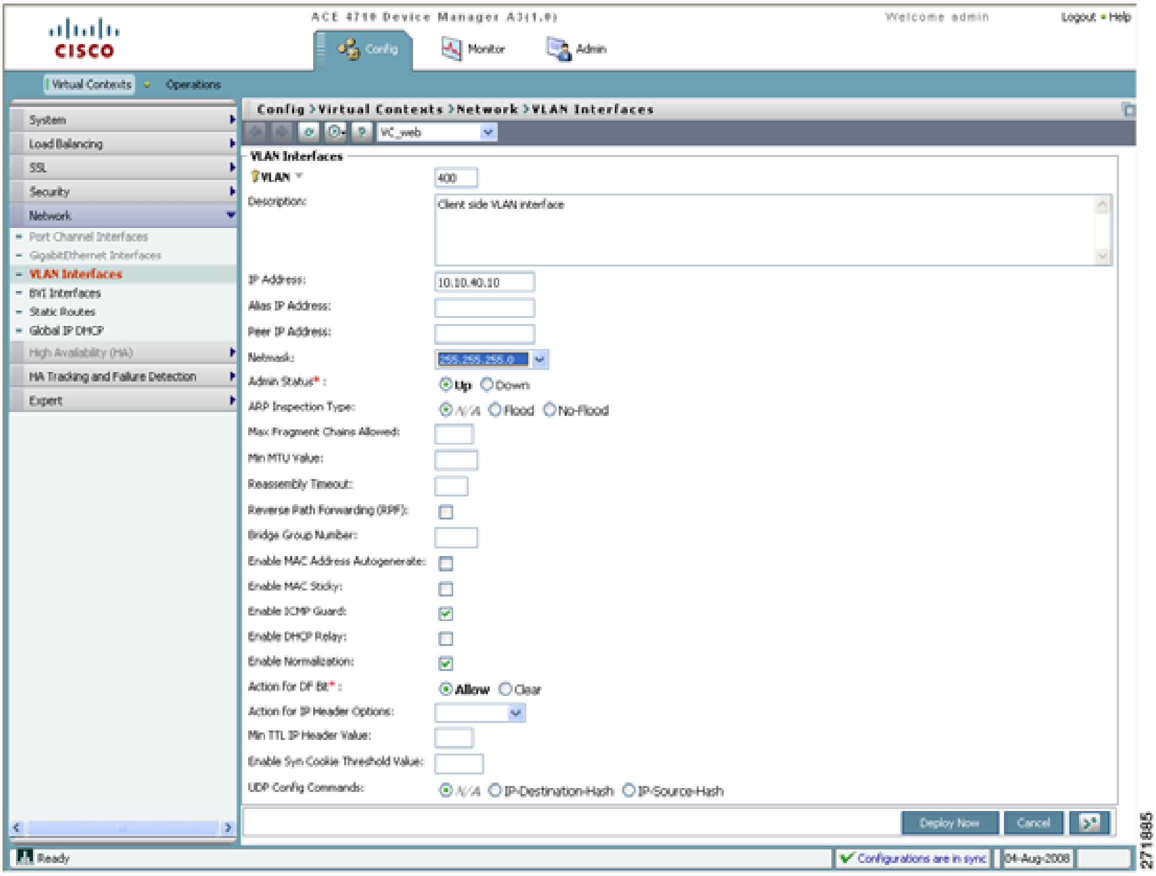
Step 2 Choose **Config** > **Virtual Contexts** > **Network** > **VLAN Interfaces**. The VLAN Interfaces pane appears (Figure 3-9).

Figure 3-9 VLAN Interfaces Pane



Step 3 Click **Add** to add a new VLAN interface. The VLAN Interfaces window appears (Figure 3-10).

Figure 3-10 VLAN Interfaces Window—VLAN 400

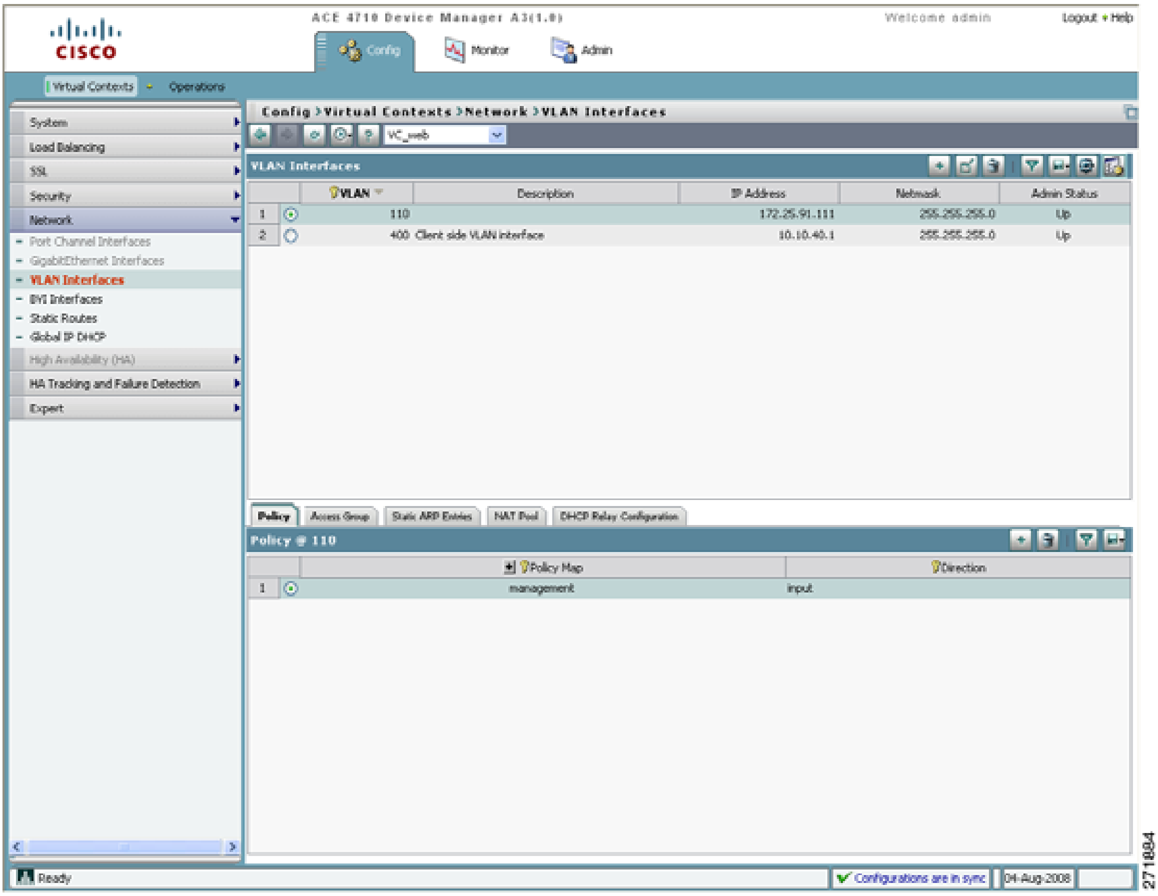


Step 4 Enter the following VLAN attributes. Leave the remaining attributes blank or with their default values.

- VLAN: 400
- Description: Client-side VLAN interface
- IP Address: 10.10.40.10
- Netmask: 255.255.255.0
- Admin Status: Up

Step 5 Click **Deploy Now** at the bottom of the window to save your entry. Then, choose **VLAN Interfaces** to return to the VLAN Interfaces pane (Figure 3-11).

Figure 3-11 VLAN Interface Pane with Two VLANs Configured

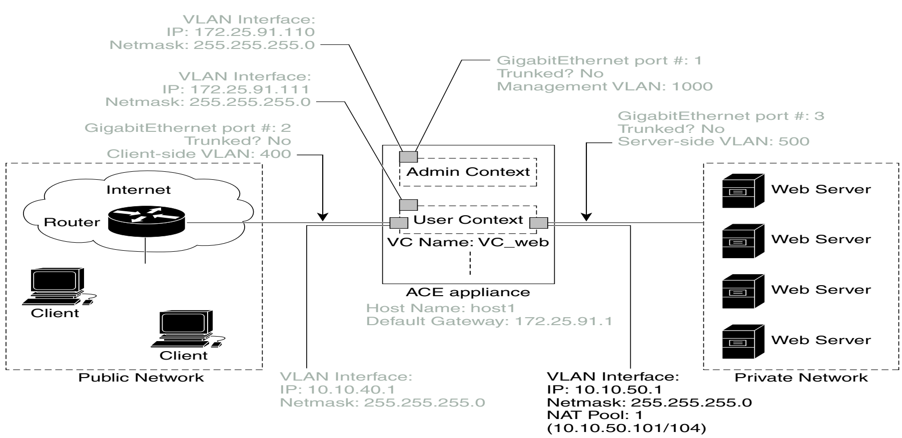


Configuring the Server-Side VLAN Interface

At this point, you can now configure the server-side VLAN interface, which is the address to which traffic is sent. For the example configuration, you will configure VLAN 500 and a NAT pool for the VLAN (Figure 3-12).

Note Network Address Translation (NAT) is designed to simplify and conserve IP addresses. It allows private IP networks that use unregistered IP addresses to connect to the Internet. You configure a NAT pool for the ACE so that the ACE exposes only one address for the entire network to the outside world. This pool, which hides the entire internal network behind that address, offers both security and address conservation.

Figure 3-12 Configuring the Server-Side VLAN Interface



Configure the VLAN interface by following these steps:

Step 1 Make sure that **VC_web** is selected in the virtual contexts drop-down list.

Step 2 Choose **Config** > **Virtual Contexts** > **Network** > **VLAN Interfaces**. The VLAN Interfaces pane appears (Figure 3-11).

Step 3 Click **Add** to add a new VLAN interface. The VLAN Interfaces window appears (Figure 3-10).

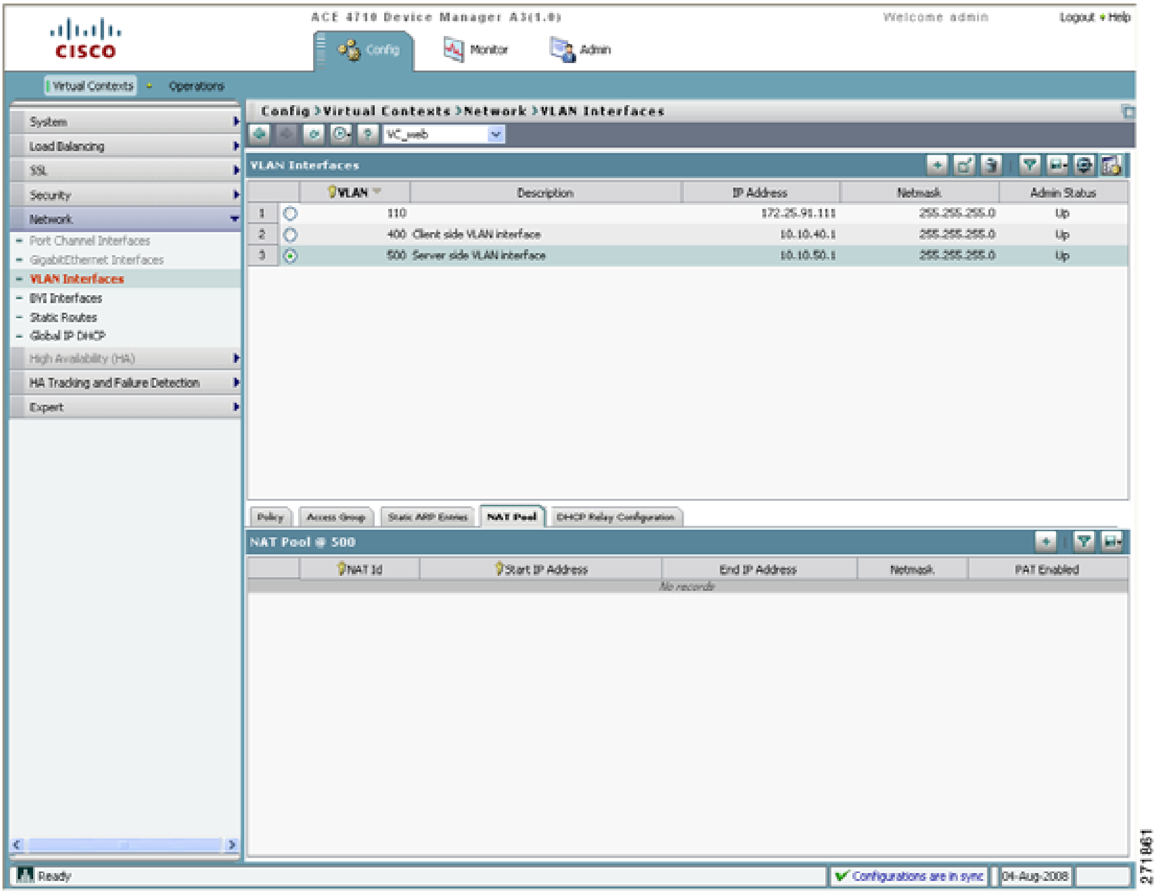
Step 4 Enter the following VLAN attributes. Leave the remaining attributes blank or with their default values.

- VLAN: 500
- Description: Server-side VLAN interface
- IP Address: 10.10.50.1
- Netmask: 255.255.255.0
- Admin Status: Up

Step 5 Click **Deploy Now** at the bottom of the window to save your entry. Then, choose **VLAN Interfaces** to return to the VLAN Interfaces pane.

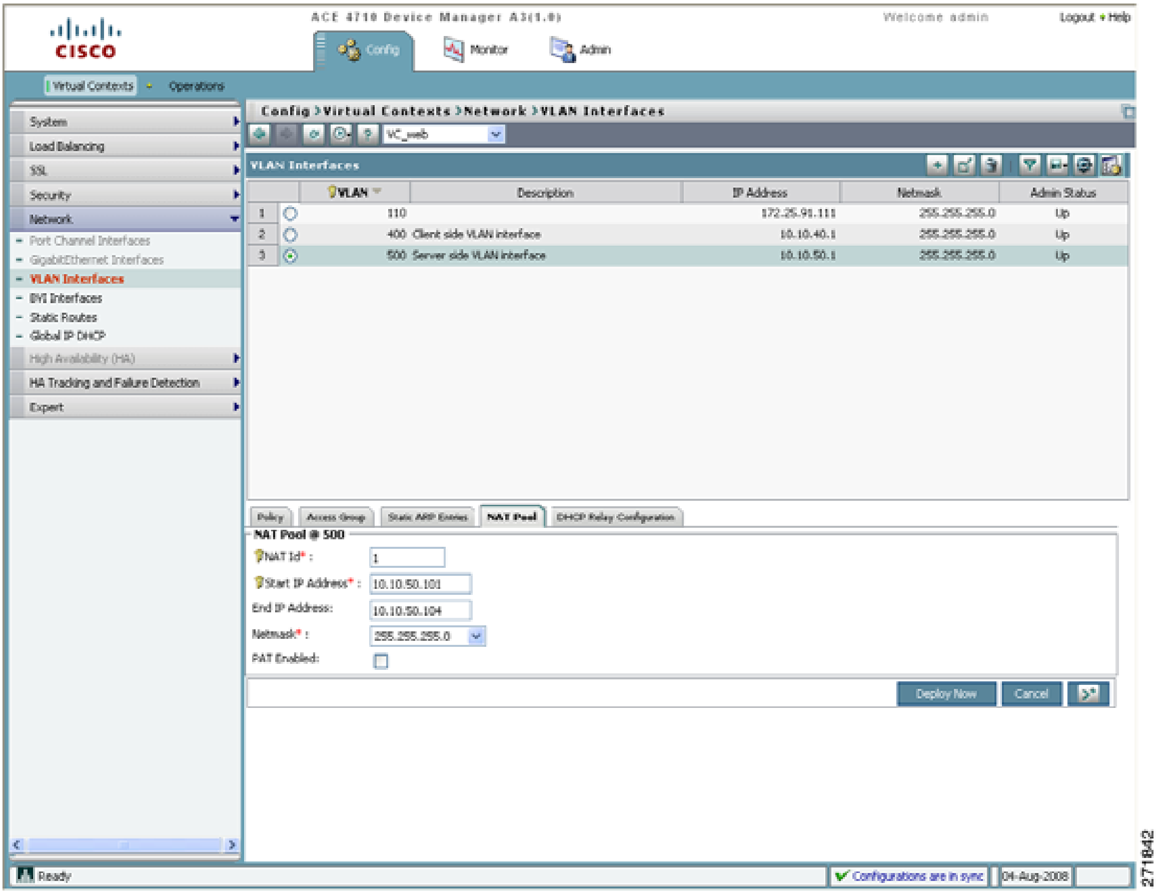
Step 6 Choose the row for VLAN 500, and then choose the **NAT Pool** tab. The NAT Pool pane appears (Figure 3-13).

Figure 3-13 NAT Pool Pane



Step 7 Click **Add** to add a new NAT pool. The NAT Pool pane appears (Figure 3-14).

Figure 3-14 Configuring a NAT Pool

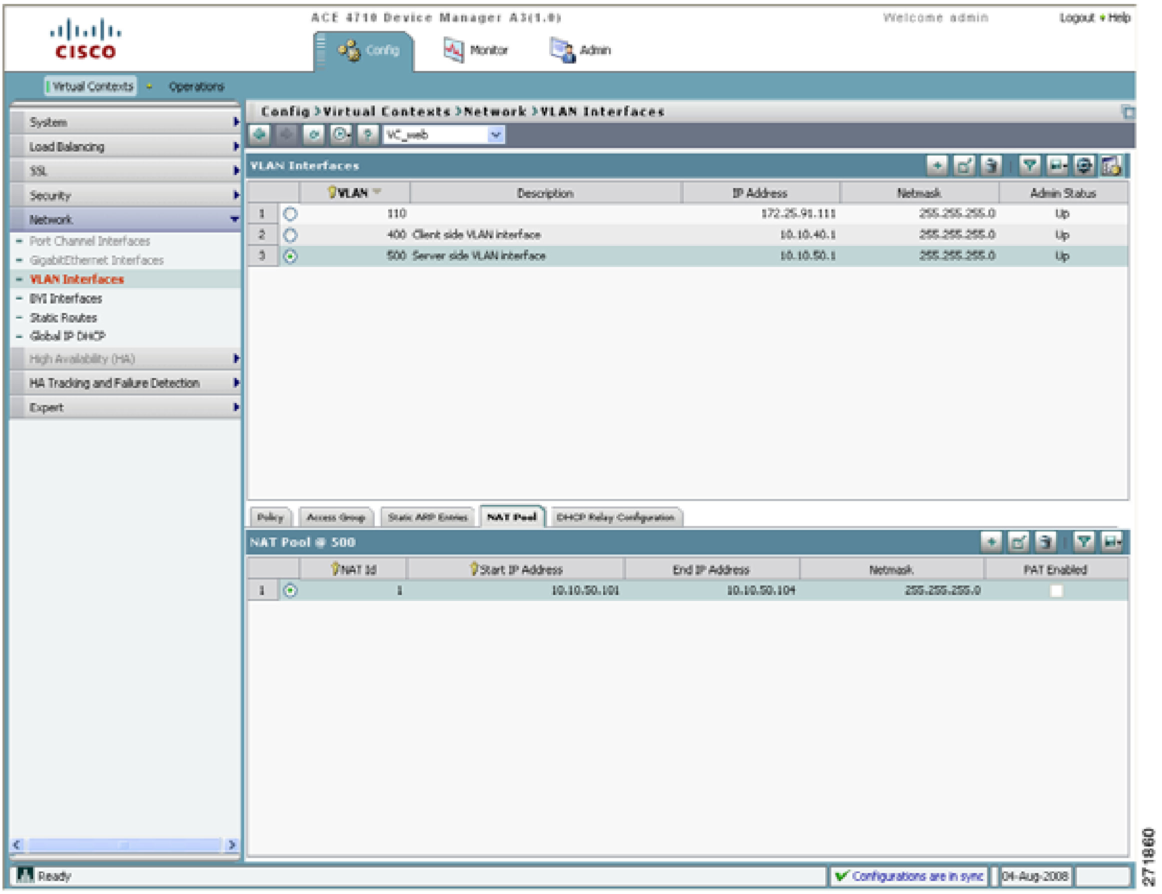


Step 8 Enter the following NAT pool attributes. Leave the remaining attributes blank or with their default values.

- NAT Id: 1
- Start IP Address: 10.10.50.101
- End IP Address: 10.10.50.104
- Netmask: 255.255.255.0

Step 9 Click **Deploy Now** at the bottom of the window to save your entry and return to the NAT Pool pane (Figure 3-15).

Figure 3-15 NAT Pool Pane with a NAT Pool Configured



Creating a Virtual Context Using the CLI

You can create a virtual context using the command-line interface. This section contains the following topics:

- Configuring a Resource Class
- Creating a Virtual Context
- Configuring a Management VLAN Interface to the User Context
- Configuring Remote Management Access to the User Contexts
- Configuring the Client-Side VLAN Interface
- Configuring the Server-Side VLAN Interface

Configuring a Resource Class

Configure a resource class by following these steps:

Step 1 Using the console, log in to the ACE as the system administrator. For example, enter the following command at a command prompt.

```
Telnet 172.25.91.110
```

At the prompt, enter **admin**, then the new password you entered in Step 2 in "Enabling Management Connectivity Using the Setup Script" in Chapter 2.

```
host1 login: admin
Password: xxxxx
```

Step 2 Enter configuration mode.

```
host1/Admin# config
host1/Admin(config)#
```

Step 3 Configure a resource class to limit the resources of a context to 10 percent of the total resources available on the ACE, and exit configuration mode.

```
host1/Admin(config)# resource-class RS_web
host1/Admin(config-resource)# limit-resource all minimum 10 maximum
unlimited
host1/Admin(config-resource)# exit
host1/Admin(config)#
```

Creating a Virtual Context

Create a virtual context by following these steps:

Step 1 Create a new context.

```
host1/Admin(config)# context VC_web
host1/Admin(config-context)#
```

Step 2 Associate three existing VLANs with the context so that the context can receive traffic classified for it.

```
host1/Admin(config-context)# allocate-interface vlan 1000
host1/Admin(config-context)# allocate-interface vlan 400
host1/Admin(config-context)# allocate-interface vlan 500
```

Step 3 Associate the context with the resource class that you created in the previous section, "Configuring a Resource Class."

```
host1/Admin(config-context)# member RC_web
```

Step 4 Change to the VC_web context that you created in Step 1 and exit configuration mode.

```
host1/Admin(config-context)# do changeto VC_web
host1/VC_web(config)# exit
host1/VC_web#
```

Step 5 Display the virtual context configuration.

```
host1/VC_web# show running-config context
```

Step 6 Display the resource class configuration.

```
host1/VC_web# show running-config resource-class
```

Configuring a Management VLAN Interface to the User Context

You can provide management connectivity to the user context by assigning an IP address to the VLAN interface, as illustrated in Figure 3-4. Configure a management VLAN interface by following these steps:

Step 1 Access interface configuration mode for VC_web for the VLAN 1000 on VC_web.

```
host1/VC_web# config
host1/VC_web(config)# interface vlan 1000
host1/VC_web(config -if)#
```

Step 2 Assign an IP address of 172.25.91.111 and a subnet mask of 255.255.255.0 to the VLAN interface for management connectivity.

```
host1/VC_web(config-if)# ip address 172.25.91.111 255.255.255.0
```

Step 3 Enable the VLAN interface.

```
host1/VC_web(config-if)# no shutdown
```

Step 4 Show that VLAN 1000 is active.

```
host1/VC_web(config-if)# do show interface vlan 1000
```

Step 5 Verify network connectivity.

```
host1/VC_web(config-if)# do ping 172.25.91.111
```

Step 6 Display the ARP table.

Note The Address Resolution Protocol (ARP) allows the ACE to manage and learn the mapping of IP to Media Access Control (MAC) information to forward and transmit packets.

```
host1/VC_web(config-if)# do show arp
```

Step 7 Exit configuration mode.

```
host1/VC_web(config-if)# exit
host1/VC_web(config)# exit
host1/VC_web#
```

Configuring Remote Management Access to the User Contexts

Before remote network access can occur on the user context through an Ethernet port, you must create a traffic policy that identifies the network management traffic that can be received by the ACE. Configure remote management access by following these steps:

Step 1 Create a management type class map named REMOTE_ACCESS that matches any traffic.

```
host1/VC_web# config
host1/VC_web(config)# class-map type management match-any
REMOTE_ACCESS
host1/VC_web(config-cmap-mgmt)#
```

Step 2 (Optional) Provide a description for the class map.

```
host1/VC_web(config-cmap-mgmt)# description Remote access traffic
match
```

Step 3 Configure the match protocol to permit traffic based on the SSH, Telnet, and ICMP protocols for any source address.

```
host1/VC_web(config-cmap-mgmt)# match protocol ssh any
host1/VC_web(config-cmap-mgmt)# match protocol telnet any
host1/VC_web(config-cmap-mgmt)# match protocol icmp any
host1/VC_web(config-cmap-mgmt)# exit
host1/VC_web(config)#
```

Step 4 Create a REMOTE_MGMT_ALLOW_POLICY policy map for traffic destined to an ACE interface.

```
host1/VC_web(config)# policy-map type management first-match
REMOTE_MGMT_ALLOW_POLICY
host1/VC_web(config-pmap-mgmt)#
```

Step 5 Apply the REMOTE_ACCESS class map to this policy.

```
host1/VC_web(config-pmap-mgmt)# class REMOTE_ACCESS
host1/VC_web(config-pmap-mgmt-c)#
```

Step 6 Allow the ACE to receive the configured class map management protocols.

```
host1/VC_web(config-pmap-mgmt-c)# permit
host1/VC_web(config-pmap-mgmt-c)# exit
host1/VC_web(config-pmap-mgmt)# exit
host1/VC_web(config)#
```

Step 7 Access interface configuration mode for the VLAN to which you want to apply the policy map.

```
host1/VC_web(config)# interface vlan 1000
host1/VC_web(config-if)#
```

Step 8 Apply the REMOTE_MGMT_ALLOW_POLICY policy map to the interface.

```
host1/VC_web(config-if)# service-policy input REMOTE_MGMT_ALLOW_POLICY
```

Step 9 Display the REMOTE_MGMT_ALLOW_POLICY policy applied to the interface.

```
host1/VC_web(config-if)# do show service-policy
REMOTE_MGMT_ALLOW_POLICY
```

Step 10 Copy your configuration changes from the running configuration to the startup configuration.

```
host1/VC_web(config-if)# do copy running-config startup-config

Generating configuration....
running config of context VC_web saved

host1/VC_web(config-if)# exit
host1/VC_web(config)# exit
```

Step 11 Display the running configuration.

```
host1/VC_web(config)# do show running-config
```

Configuring the Client-Side VLAN Interface

At this point, you can configure a client-side VLAN interface, the address to which the client traffic is sent, as illustrated in Figure 3-8. Configure a client-side VLAN interface by following these steps:

Step 1 Access interface configuration mode for the VLAN 400.

```
host1/VC_web(config)# interface vlan 400
host1/VC_web(config -if)#
```

Step 2 Assign an IP address of 10.10.40.1 and a subnet mask of 255.255.255.0 to the VLAN interface for client connectivity.

```
host1/VC_web(config-if)# ip address 10.10.40.1 255.255.255.0
```

Step 3 (Optional) Provide a description for the interface.

```
host1/VC_web(config-if)# description Client connectivity on VLAN 400
```

Step 4 Enable the VLAN interface.

```
host1/VC_web(config-if)# no shutdown
```

Step 5 Show that VLAN 400 is active.

```
host1/VC_web(config-if)# do show interface vlan 400
```

Step 6 Display the ARP table.

```
host1/VC_web(config-if)# do show arp
```

Step 7 Exit configuration mode.

```
host1/VC_web(config-if)# exit
host1/VC_web(config)# exit
host1/VC_web#
```

Configuring the Server-Side VLAN Interface

Next, you can configure a server-side VLAN interface, the address to which the server traffic is sent, as illustrated in Figure 3-12. Configure the server-side VLAN interface by following these steps:

Step 1 Access interface configuration mode for the VLAN 500.

```
host1/VC_web# config
host1/VC_web(config)# interface vlan 500
host1/VC_web(config -if)#
```

Step 2 Assign an IP address of 10.10.50.1 and a subnet mask of 255.255.255.0 to the VLAN interface for server-side connectivity.

```
host1/VC_web(config-if)# ip address 10.10.50.1 255.255.255.0
```

Step 3 (Optional) Provide a description for the interface.

```
host1/VC_web(config-if)# description Server connectivity on VLAN 500
```

Step 4 Enable the VLAN interface.

```
host1/VC_web(config-if)# no shutdown
```

Step 5 Configure a NAT pool.

```
host1/VC_web(config-if)# nat-pool 1 10.10.50.101 10.10.50.104 netmask
255.255.255.0
```

Step 6 Show that VLAN 500 is active.

```
host1/VC_web(config-if)# do show interface vlan 500
```

Step 7 Display the ARP table.

```
host1/VC_web(config-if)# do show arp
```

Step 8 Exit configuration mode.

```
host1/VC_web(config-if)# exit
host1/VC_web(config)# exit
host1/VC_web#
```

In this chapter, you have partitioned your ACE into an Admin context and a user context VC_web. Each of the virtual contexts is now associated with a resource class that is appropriate to its intended use. You have also configured a management VLAN interface, as well as the client and server VLAN interfaces to the user context.

In the next chapter, you will configure an access control list to secure your network.

CHAPTER 4

Configuring Access Control Lists

This chapter describes how to configure access control lists (ACLs) for the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following sections:

- Overview
- Configuring an ACL Using the Device Manager GUI
- Configuring an ACL Using the CLI

Overview

After reading this chapter, you should have a basic understanding of how to configure an access control list in an ACE to secure your network.

You can use ACLs with the ACE appliance to permit or deny traffic to or from a specific IP address or an entire network. For example, you can permit all e-mail traffic on a circuit, but block Telnet traffic. You can also use ACLs to allow one client to access a part of the network while preventing other clients from doing so.

You must configure an ACL on each interface that you want to permit connections. Otherwise, the ACE will deny all traffic on the interface. An ACL consists of a series of ACL entries, which are permit-or-deny entries with criteria for the source IP address, destination IP address, protocol, port, or protocol-specific parameters. Each entry permits or denies inbound or outbound network traffic to the parts of your network specified in the entry.

The order of the ACL entries is important. When the ACE decides whether to accept or refuse a connection, it tests the packet against each ACL entry in the order in which the entries are listed. After it finds a match, it stops checking entries.

For example, if you create an entry at the beginning of an ACL that explicitly permits all traffic, the ACE skips any other entries in the ACL. An implicit deny all entry exists at the end of every ACL, so you must include entries for every interface on which you want to permit connections. Otherwise, the ACE appliance will deny all traffic on the interface.

Certain applications require special handling of the data portion of a packet as the packets pass through the ACE. The ACE verifies the protocol behavior and identifies unwanted or malicious traffic that attempts to pass through. Based on the specifications of the traffic policy, the ACE performs application protocol inspection to accept or reject the packet to ensure the secure use of applications and services.

For more information on how to configure an ACL to permit or deny specific traffic or resources, see the *Cisco 4700 Application Control Engine Series Appliance Security Configuration Guide*.

The basic steps in configuring an ACL include:

- Creating an ACL
- Adding at least one ACL entry to the ACL
- Associating the ACL with an interface

To configure an ACL, you can use either the ACE Device Manager user interface (GUI) or the CLI.

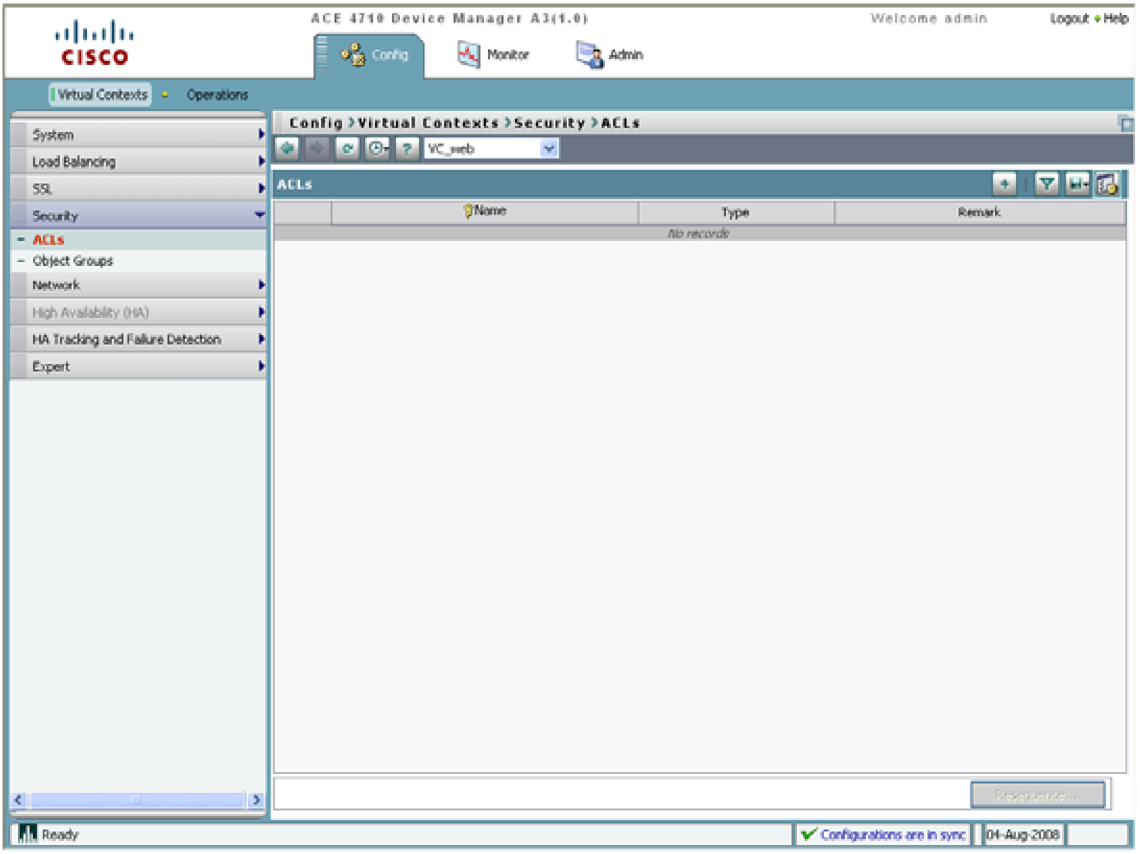
Configuring an ACL Using the Device Manager GUI

Configure an ACL using the ACE Device Manager GUI by following these steps:

Step 1 Choose **VC_web**.

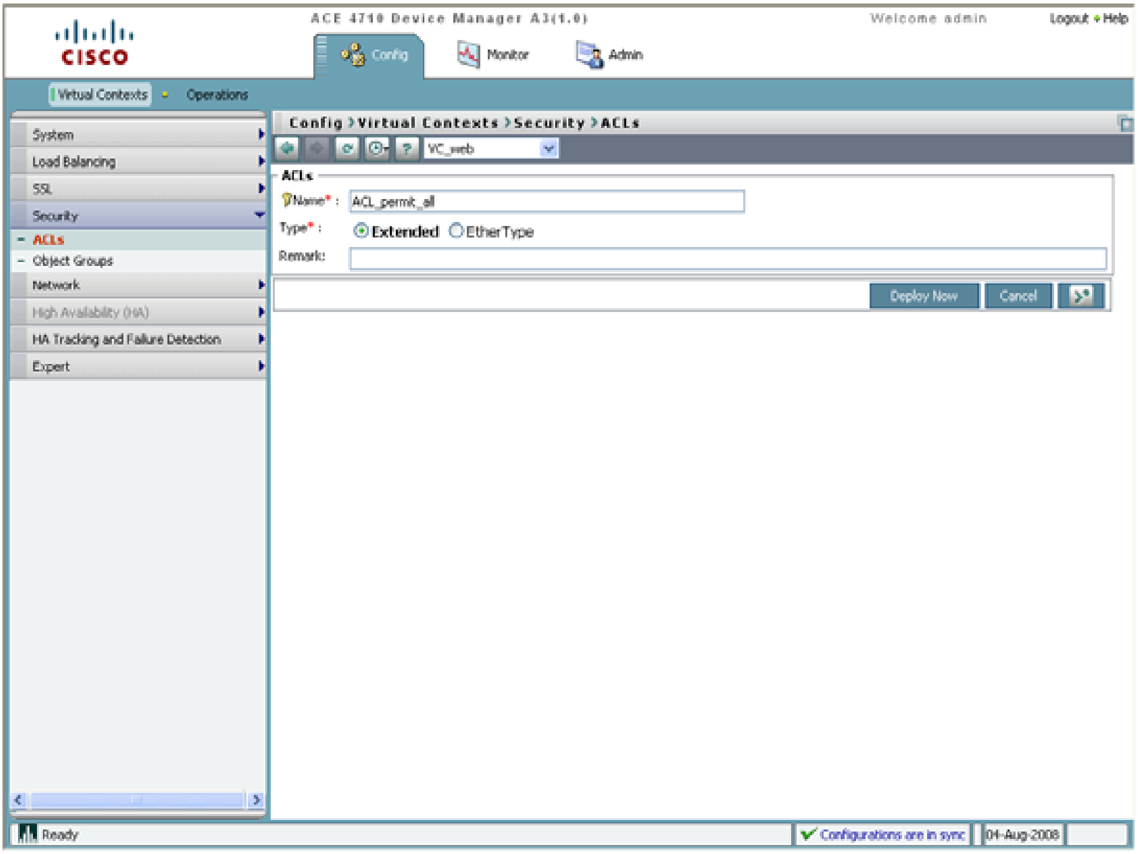
Step 2 Choose **Config** > **Virtual Contexts** > **Security** > **ACLs**. The ACLs pane appears, listing the existing ACLs (Figure 4-1).

Figure 4-1 ACLs Pane



Step 3 Click **Add** to create an ACL. The ACL configuration window appears (Figure 4-2).

Figure 4-2 ACL Configuration Window



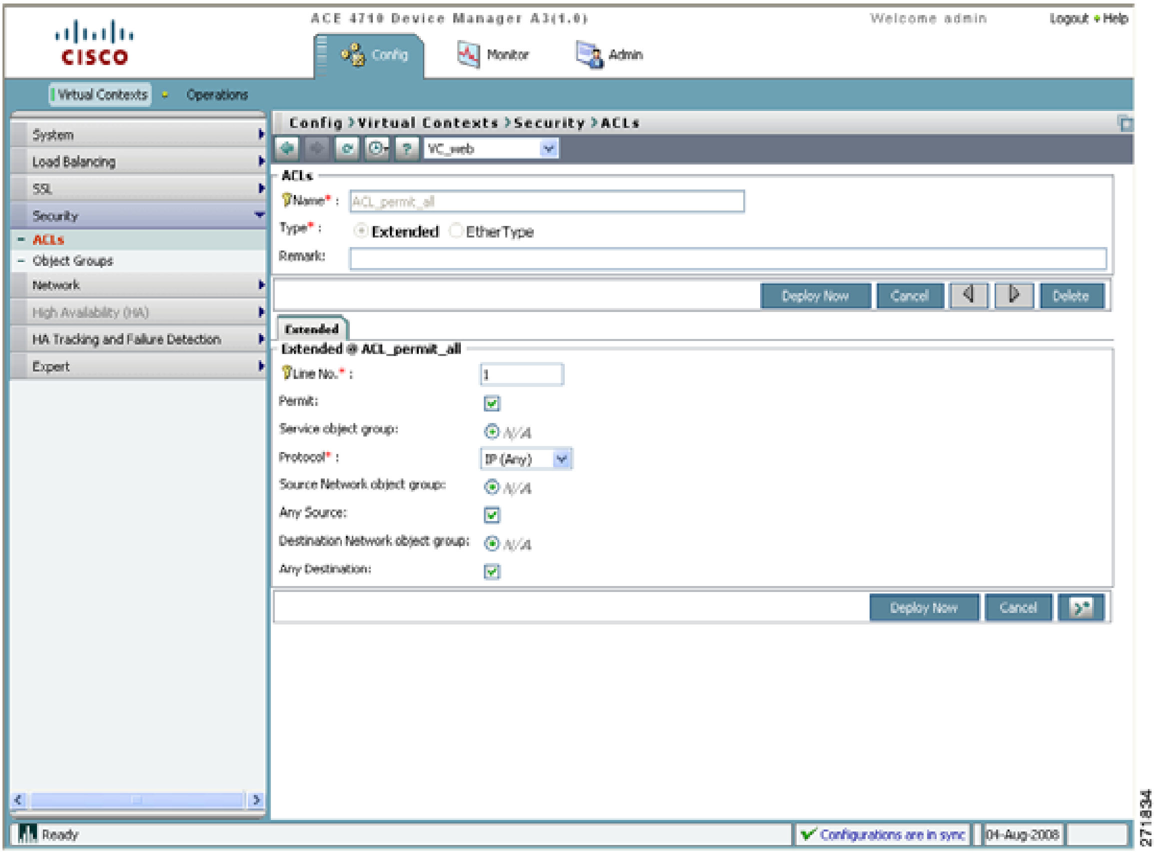
Step 4 Enter the following ACL properties. Leave the remaining properties blank or with the default values.

- Name: ACL_permit_all
- Type: Extended
  - Extended—Control network access for IP traffic
  - EtherType—Control network access for non-IP traffic

Step 5 Click **Deploy Now**. The Extended pane appears.

Step 6 Click **Add** to create an ACL entry. The ACL entry configuration window appears (Figure 4-3).

Figure 4-3 ACL Entry Configuration Window



Step 7 Create an ACL entry with the following attributes. Leave the remaining attributes blank or with the default values.

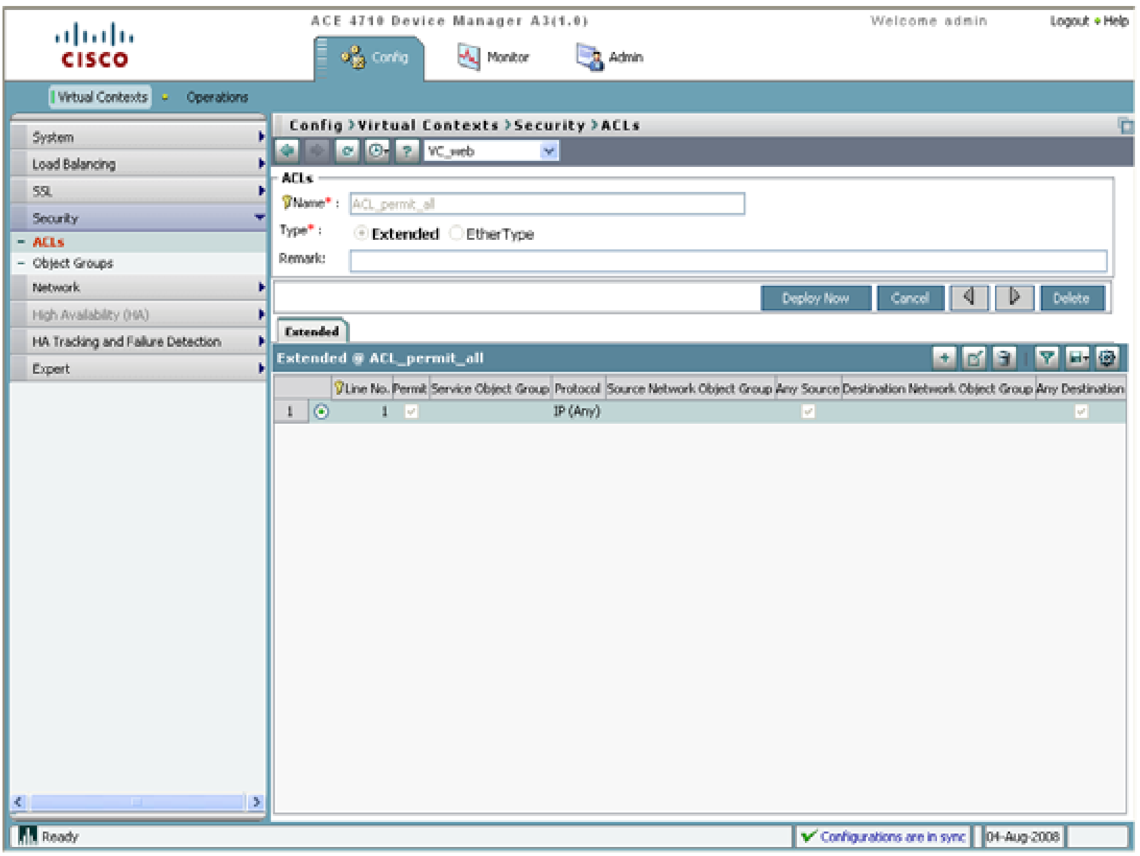
- Line No.: 1

Note For easier insertion of additional ACL entries later, you can enter non-sequential line numbers such as 10, 20, and so on.

- Permit: (Checked)
- Protocol: IP (Any)
- Any Source: (Checked)
- Any Destination: (Checked)

Step 8 Click **Deploy Now** to save the ACL entry on the virtual context. The ACL entry is added to the Extended @ ACL_permit_all pane (Figure 4-4).

Figure 4-4 ACL Entry is Added

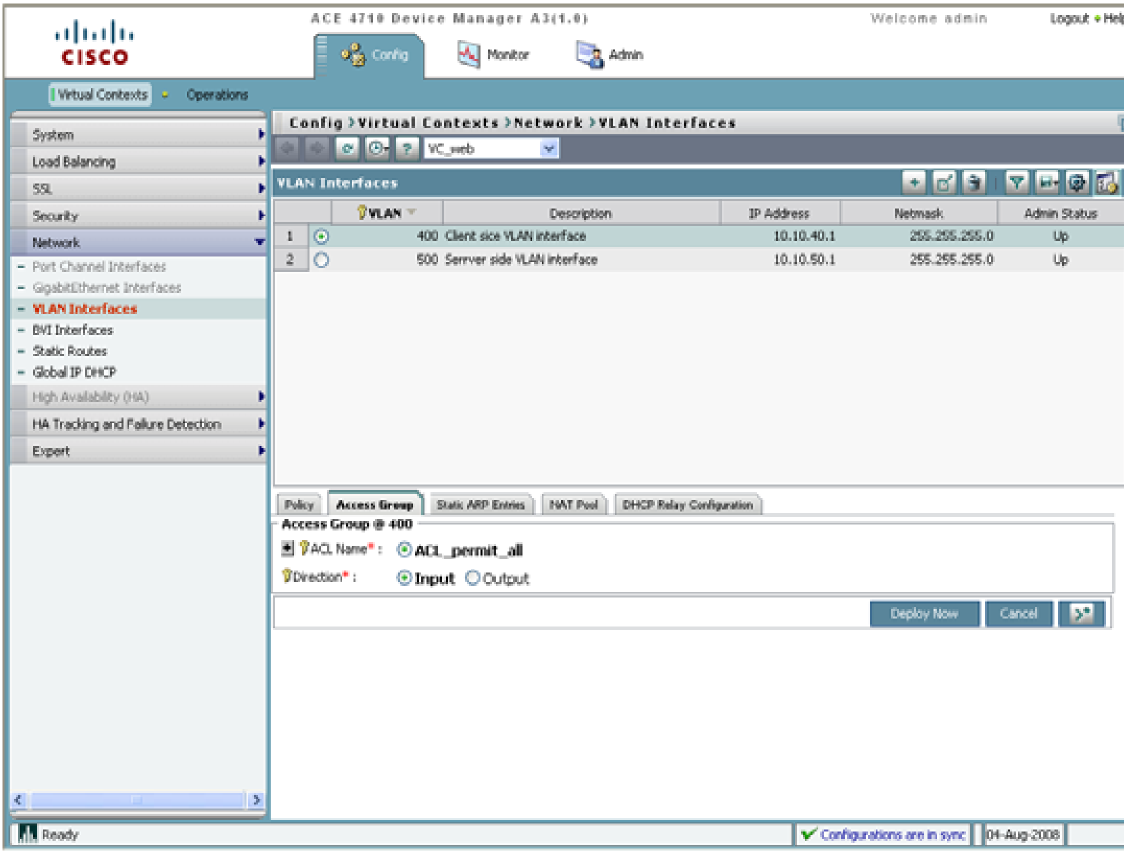


Step 9 Choose **Network > VLAN Interfaces**. The VLAN Interfaces pane appears.

Step 10 Choose the **Access Group** tab.

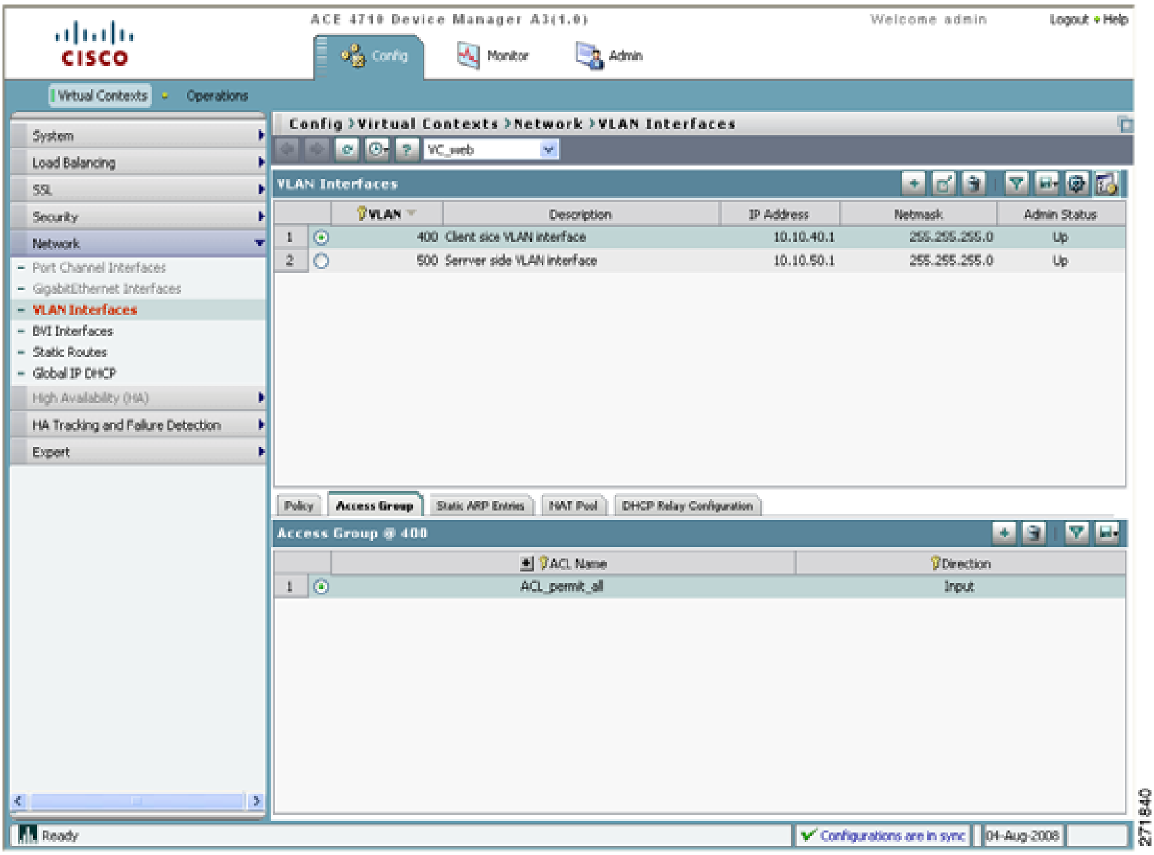
Step 11 Click **Add** above the pane (Figure 4-5).

Figure 4-5 Adding an ACL to an Interface



Step 12 Click **Deploy Now** to accept the defaults and add an ACL to the interface. The ACL is added in the Access Group pane (Figure 4-6).

Figure 4-6 ACL is Added to an Interface



Configuring an ACL Using the CLI

You can configure an ACL using the command-line interface (CLI) by following these steps:

Step 1 Check the CLI prompt to verify that you are operating in the desired context; change to the correct context if necessary.

```
host1/Admin# changeto VC_web
host1/VC_web#
```

Step 2 Enter configuration mode.

```
host1/VC_web# Config
host1/VC_web(config)#
```

Step 3 Create an ACL.

```
host1/VC_web(config)# access-list INBOUND extended permit ip any any
```

Step 4 Apply the ACL to an interface.

```
host1/VC_web(config)# interface vlan 400
host1/VC_web(config-if)# access-group input INBOUND
host1/VC_web(config-if)# exit
```

Step 5 Display the ACL configuration information.

```
host1/VC_web(config)# exit
host1/VC_web# show running-config access-list
```

In this chapter, you have created an ACL entry to permit all traffic to the network. Next, you will create a user who is allowed to perform a subset of the ACE management functions on part of your network resources.

CHAPTER 5

Configuring Role-Based Access Control

This chapter describes how to configure role-based access control (RBAC) on the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following sections:

- Overview
- Configuring RBAC Using the Device Manager GUI
- Configuring RBAC Using the CLI

Overview

After reading this chapter, you should have a basic understanding of how the ACE appliance provides security administration by using RBAC and how to configure a server maintenance user with permission to access a subset of your network.

One of the most challenging problems in managing large networks is the complexity of security administration. The ACE appliance allows you to determine the commands and resources available to each user through RBAC. In RBAC, users are associated with domains and roles.

A domain is a collection of physical and virtual network resources such as real servers and virtual servers.

User roles determine a user's privileges, such as the commands that the user can enter and the actions the user can perform in a particular context. The ACE provides a number of predefined roles. In addition, administrators in any context can define new roles.

The ACE provides the following predefined roles, which you cannot delete or modify:

- Admin—If created in the Admin context, has complete access to, and control over, all contexts, domains, roles, users, resources, and objects in the entire ACE. If created in a user context, gives a user complete access to and control over all policies, roles, domains, server farms, real servers, and other objects in that context.
- Network Admin—Has complete access to and control over the following features:
  - Interfaces
  - Routing
  - Connection parameters
  - Network Address Translation (NAT)
  - VIPs
  - Copy configurations
  - **changeto** command
- Network-Monitor—Has access to all **show** commands and to the **changeto** command. If you do not explicitly assign a role to a user with the **username** command, this is the default role.
- Security-Admin—Has complete access to and control over the following security-related features within a context:
  - ACLs
  - Application inspection
  - Connection parameters
  - Interfaces
  - Authentication, authorization, and accounting (AAA)
  - NAT

  - Copy configurations
  - **changeto** command
- Server-Appln-Maintenance—Has complete access to and control over the following features:
  - Real servers
  - Server farms
  - Load balancing
  - Copy configurations
  - **changeto** command
- Server-Maintenance—Can perform real server maintenance, monitoring, and debugging for the following features:
  - Real servers—Modify permission
  - Server farms—Debug permission
  - VIPs—Debug permission
  - Probes—Debug permission
  - Load balancing—Debug permission
  - **changeto** command—Create permission
- SLB-Admin—Has complete access to and control over the following ACE features within a context:
  - Real servers
  - Server farms
  - VIPs
  - Probes
  - Load balancing (Layer 3/4 and Layer 7)
  - NAT
  - Interfaces
  - Copy configurations
  - **changeto** command

- SSL-Admin—Can administer all SSL features:
  - SSL—Create permission
  - PKI—Create permission
  - Interfaces—Modify permission
  - Copy configurations—Create permission
  - **changeto** command—Create permission

You can create a user and assign them privileges through RBAC as follows:

Step 1 Create a domain and choose network resources for the domain.

Step 2 Create a user and associate the user with the following:

- A role (predefined or custom)
- A domain

This chapter describes how to create a domain and a user, and how to associate the user with a predefined role and the new domain. For more information on predefined roles and how to define a custom role, see the *Cisco 4700 Series Application Control Engine Appliance Virtualization Configuration Guide.*

To create a domain and a user, you can use either the ACE Device Manager GUI or the CLI.

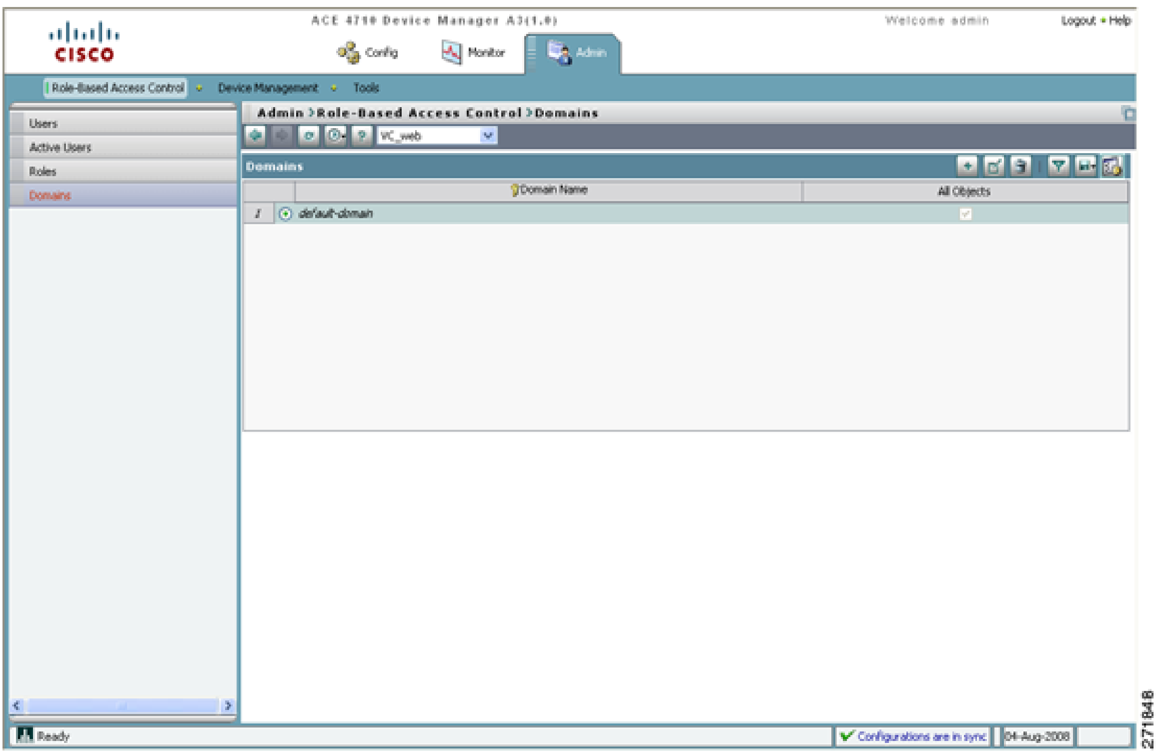
Configuring RBAC Using the Device Manager GUI

In this procedure, you use the GUI to create a domain that includes the user context that you created in Chapter 3, "Creating a Virtual Context," and then create a server maintenance user, user1, to manage those servers. Configure this RBAC setup using the GUI by following these steps:

Step 1 Choose **VC_web**.

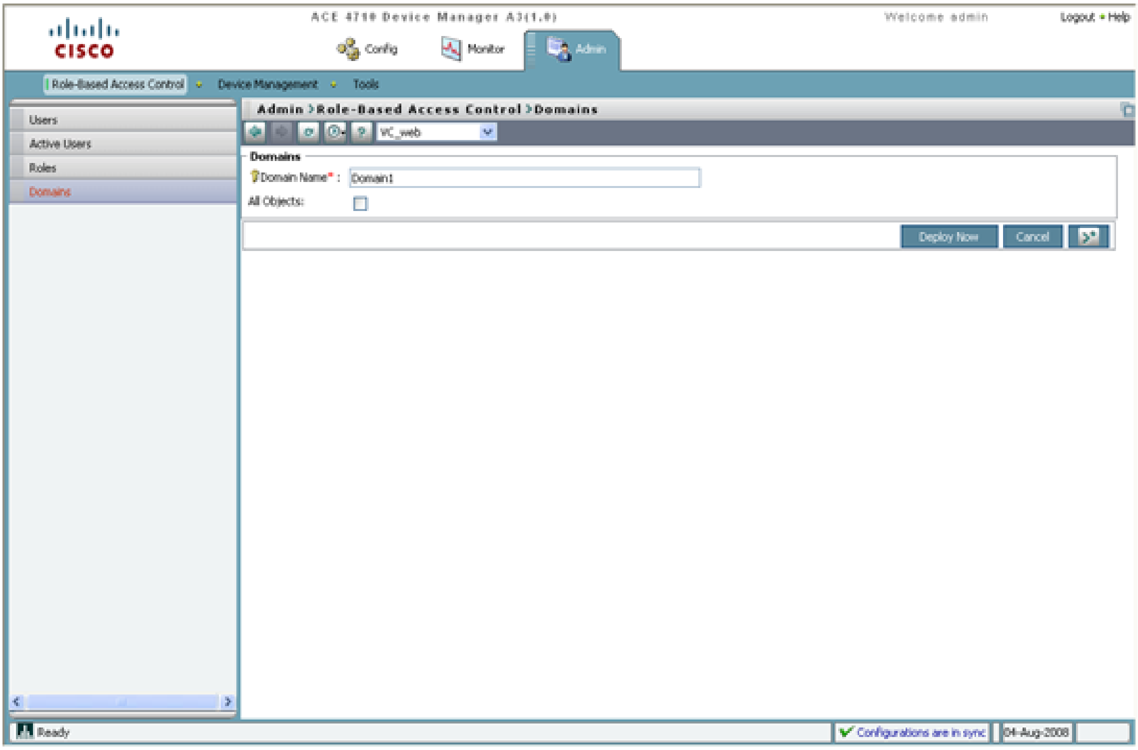
Step 2 Choose **Admin** > **Role-Based Access Control** > **Domains**. The Domains pane appears (Figure 5-1).

Figure 5-1 Domains Pane



Step 3 Click **Add** to add a new domain. The New Domain window appears (Figure 5-2).

Figure 5-2 Domains Window



271847

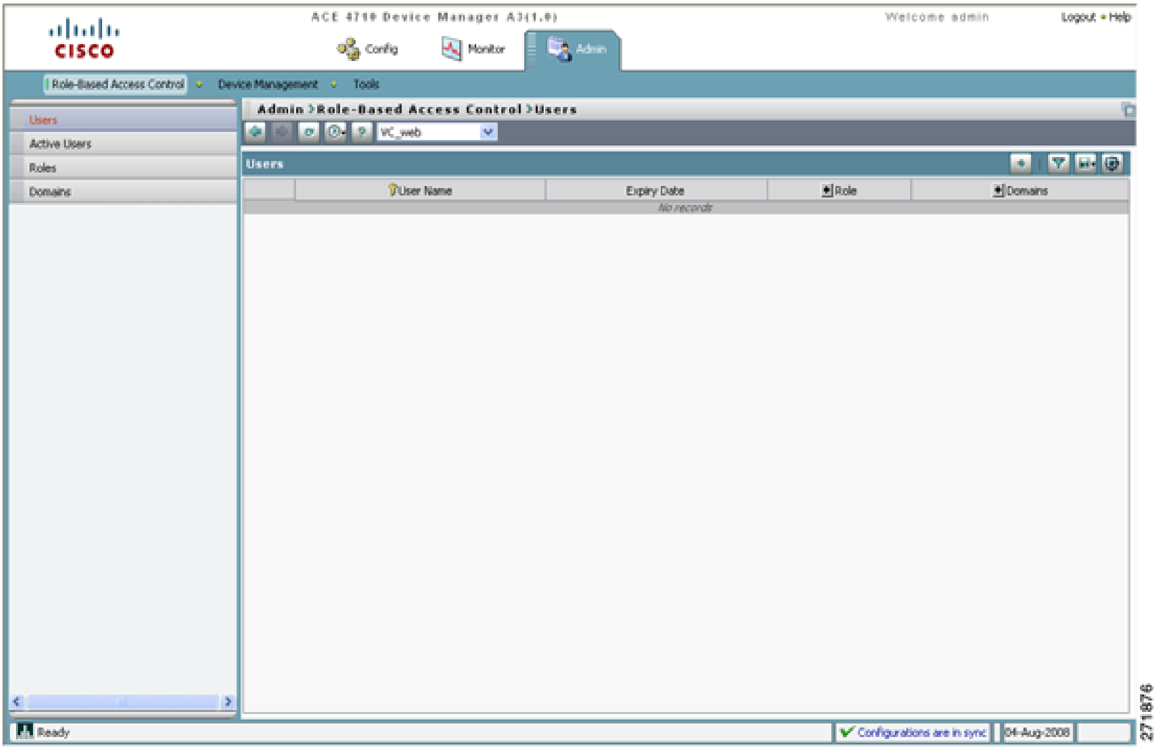
Step 4 Enter **Domain1** for the Domain Name.

Step 5 Select **All Objects**.

Step 6 Click **Deploy Now** to create a domain that includes all objects in context VC_web.

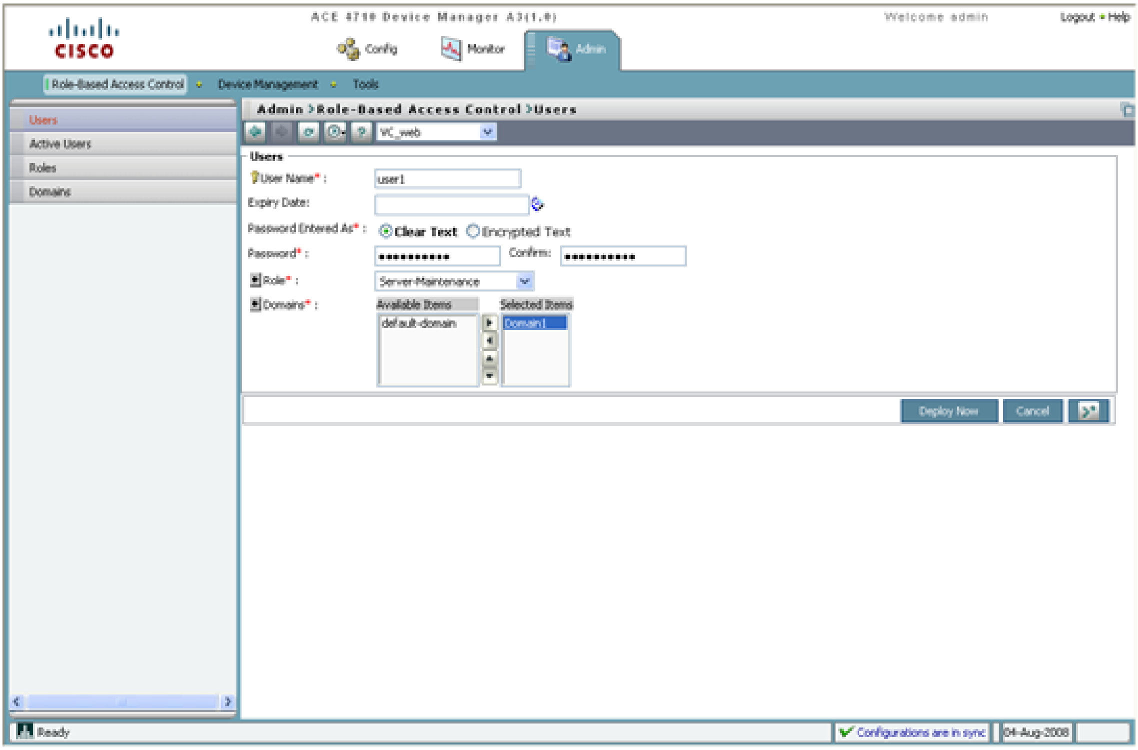
Step 7 Choose **Role-Based Access Control > Users** to create a user. The Users pane appears (Figure 5-3).

Figure 5-3 Users Pane



Step 8 Click **Add**. The User window appears (Figure 5-4).

Figure 5-4   Users Window



271874

Step 9 Enter the following user attributes. Leave the remaining attributes blank or with the default values.

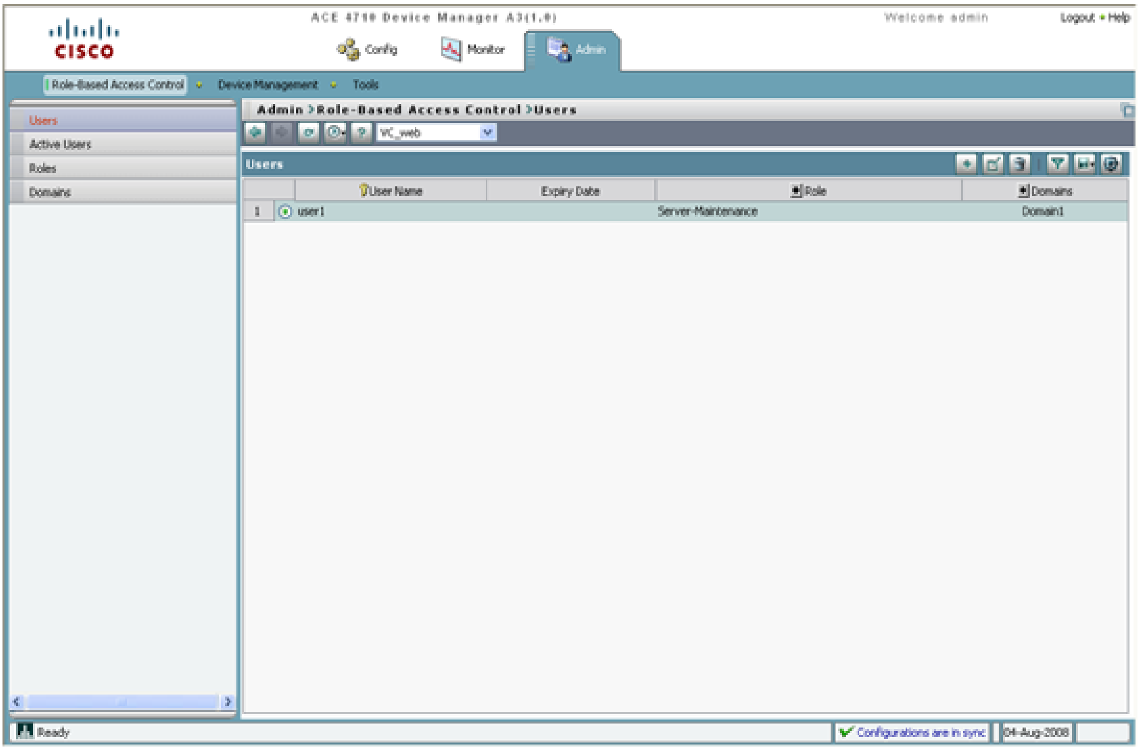
- User Name: user1
- Password: MYPASSWORD
- Confirm: MYPASSWORD
- Role: Server-Maintenance

Step 10 Choose **Domain1** and click the **right-arrow** button. Domain1 is moved to the Selected Items list.

Step 11 Choose **default-domain** and click the **left-arrow** button. Default-domain is removed from the Selected Items list.

Step 12 Associate the new user user1 with the role Server-Maintenance and the domain Domain1 by clicking **Deploy Now**. The new user is added to the Users pane (Figure 5-5).

Figure 5-5 Users Pane with user1 Added



271875

Configuring RBAC Using the CLI

Configure RBAC using the CLI by following these steps:

Step 1 Verify that you are operating in the desired context by checking the CLI prompt. If necessary, change to the correct context.

```
host1/Admin# changeto VC_web
host1/VC_web#
```

Step 2 Enter configuration mode.

```
host1/VC_web# Config
host1/VC_web(config)#
```

Step 3 Create a domain for the context.

```
host1/VC_web(config)# domain Domain1
host1/VC_web(config-domain)#
```

Step 4 Allocate all objects in the VC_web context to the domain.

```
host1/VC_web(config-domain)# add-object all
host1/VC_web(config-domain)# exit
host1/VC_web(config)#
```

Step 5 Configure new user user1, and assign the predefined role TECHNICIAN and the domain Domain1 to the user.

```
host1/VC_web(config)# username user1 password 5 MYPASSWORD role
TECHNICIAN domain Domain1
```

Note The parameter 5 for password is for an MD5-hashed strong encryption password. Use 0 for a clear text password.

```
host1/VC_web(config)# exit
```

Step 6 Display the user and domain configurations.

```
host1/VC_web# show running-config role
host1/VC_web# show running-config domain
```

In this chapter, you have created a user to perform a limited number of functions on a subset of your network. Next, you will create a virtual server for server load balancing.

CHAPTER 6

Configuring Server Load Balancing

This chapter describes how to configure server load balancing on the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following sections:

- Overview
- Configuring Layer 7 Server Load Balancing Using the Device Manager GUI
- Configuring Layer 7 Server Load Balancing Using the CLI

Overview

After reading this chapter, you should have an understanding of the basic server load-balancing capabilities provided by the ACE appliance. You should also be able to configure a virtual server for Layer 7 load-balancing purposes.

When there is a client request for web services, a load-balancing device decides to which server it should send the request. For example, a client request may consist of an HTTP GET for a web page or an FTP GET to download a file. The ACE, as a server load balancer, selects a server that can successfully fulfill the client request in the shortest amount of time without overloading either the server or the server farm as a whole.

The ACE uses a virtual server to intercept web traffic to a website. A virtual server allows multiple real servers to appear as one for load-balancing purposes. A virtual server, also called a Virtual IP (VIP), is defined by its IP address, the protocol used (for example, UDP or TEC), and the port address.

Multiple servers grouped together in server farms are assigned to each virtual server and the ACE appliance carries out load balancing across them. Real servers are dedicated servers that provide services to clients—for example, delivery of HTTP or XML content. Server farms contain the same content and typically reside in the same physical location in a data center.

You can configure the ACE for server load balancing by following these steps:

Step 1 Create a virtual server.

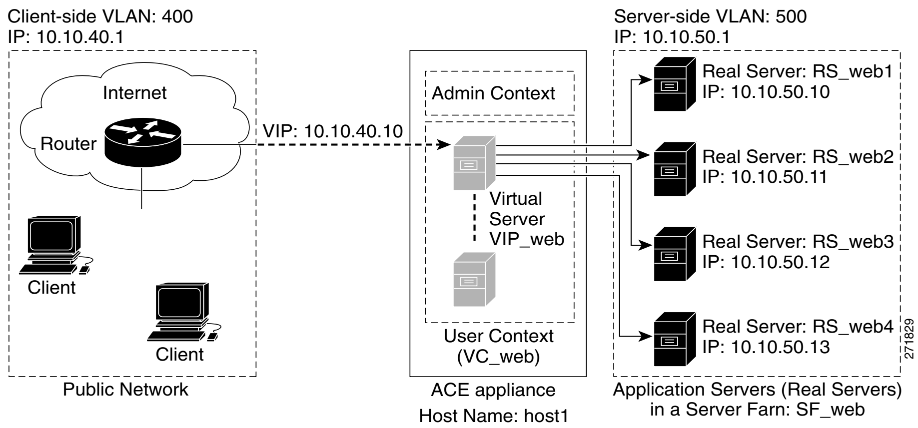
Step 2 Configure the real servers and associate them with a server farm.

Step 3 Assign the server farm to the virtual server.

Step 4 Deploy the configuration.

This chapter describes how to configure a virtual server using either the Device Manager GUI or the CLI, using the network setup example illustrated in Figure 6-1.

Figure 6-1 Example Server Load-Balancing Setup



The configuration of the example setup is as follows:

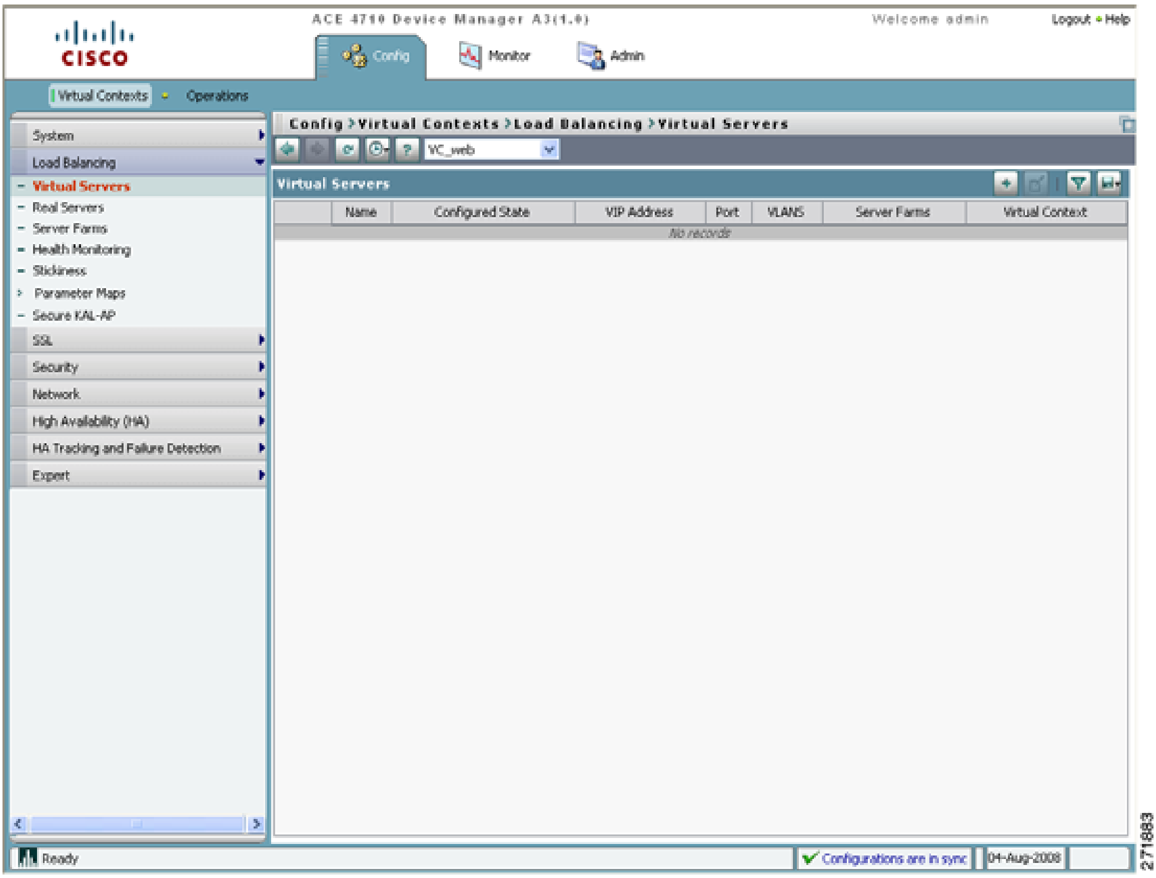
- A virtual server VS_web is created with a virtual IP address 10.10.40.10 to forward the client traffic from VLAN 400 to the application servers in VLAN 500.
- There are four real servers grouped into the server farm SF_web.
- The virtual server uses a round-robin predictor to forward the client requests to one of the real servers in the server farm.

Configuring Layer 7 Server Load Balancing Using the Device Manager GUI

You can configure Layer 7 server load balancing using the Device Manager GUI by following these steps:

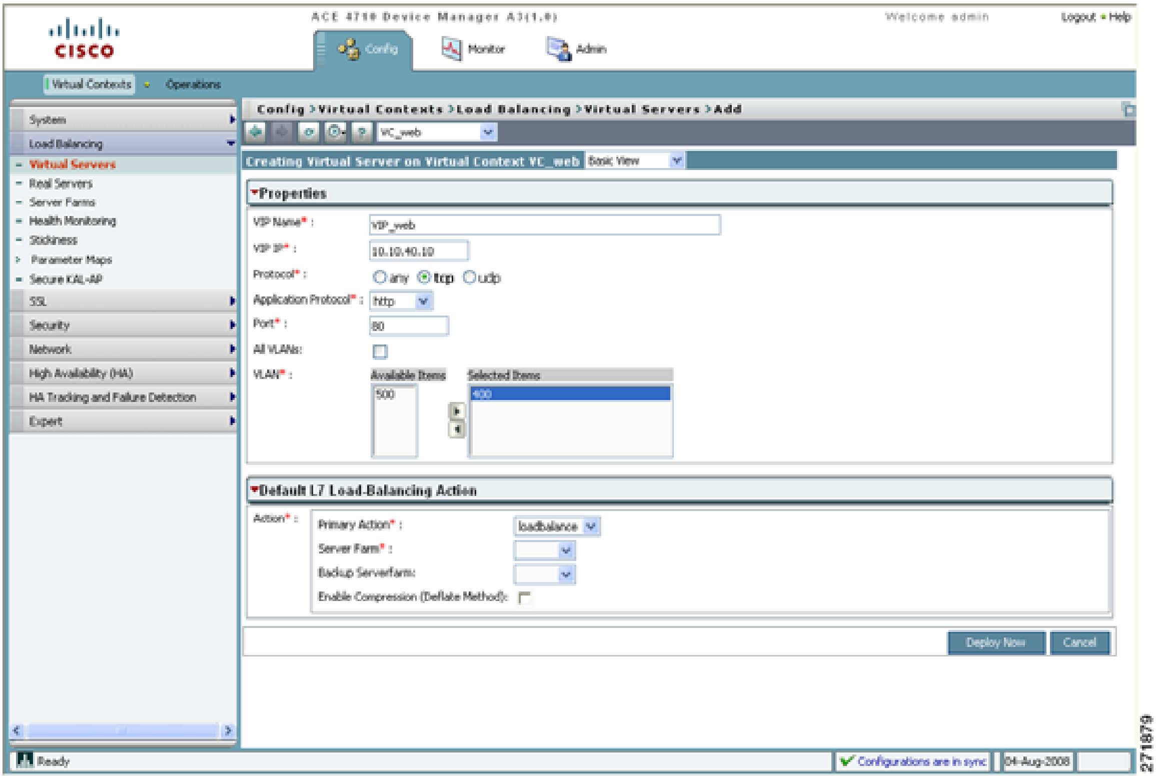
Step 1 Choose **Load Balancing > Virtual Servers**. The Virtual Servers pane appears (Figure 6-2). Choose the user context **VC_web**.

Figure 6-2 Virtual Servers Pane



Step 2 Click **Add** to add a new virtual server. The Virtual Server configuration window appears (Figure 6-3).

Figure 6-3 Properties in the Virtual Server Configuration Window



By default, the Basic View configuration option is selected and the Properties section is open.

Step 3 In Properties, enter the following virtual server attributes. Leave the remaining attributes blank or with their default values.

- VIP Name: VS_web
- VIP IP: 10.10.40.10

Note A client request targeted at a website (a URL) is translated to an IP address according to the Domain Name System (DNS). A virtual IP address assigned to a virtual server is the IP address that corresponds to the URL of the website from which the client requests services.

- Protocol: TCP
- Application Protocol: HTTP
- Port: 80
- VLAN: 400

Step 4 In the Default L7 Load-Balancing Action section, choose **loadbalance** from the Primary Action drop-down list.

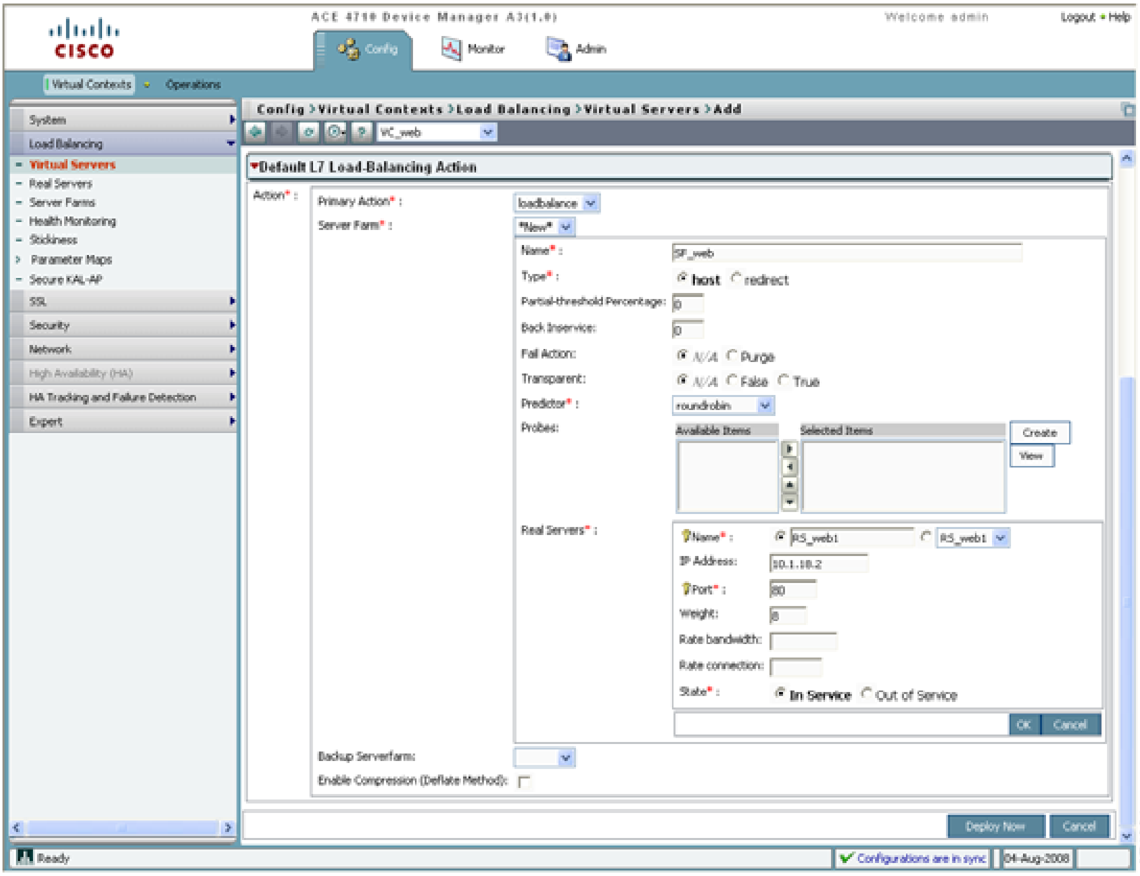
Step 5 Choose ***New*** from the Server Farm drop-down list to configure a new server farm.

Step 6 Enter the following server farm attributes. Leave the remaining attributes blank or with their default values.

- Name: SF_web
- Type: host
- Predictor: roundrobin

Step 7 Click **Add** to add a new entry to the Real Servers pane. A new entry appears in the Real Servers pane (Figure 6-4).

Figure 6-4 ***Real Servers Pane in the Virtual Server Configuration Window***



Step 8 Enter the following attributes for the first real server to be configured. Leave the remaining attributes blank or with their default values.

- Name: RS_web1
- IP Address: 10.10.50.10
- Port: 80
- Weight: 8
- State: In Service

Click **OK** to save the attributes of the first real server.

Note For information on how to configure a health probe, see Chapter 10, "Configuring Health Monitoring Using Health Probes."

Step 9 Add three more entries to the Real Servers pane by repeating Steps 7 and 8 with the following real server names and corresponding IP addresses. Leave the remaining attributes with their default values.

For RS_web2, enter:

- Name: RS_web2
- IP Address: 10.10.50.11
- Port: 80

For RS_web3, enter:

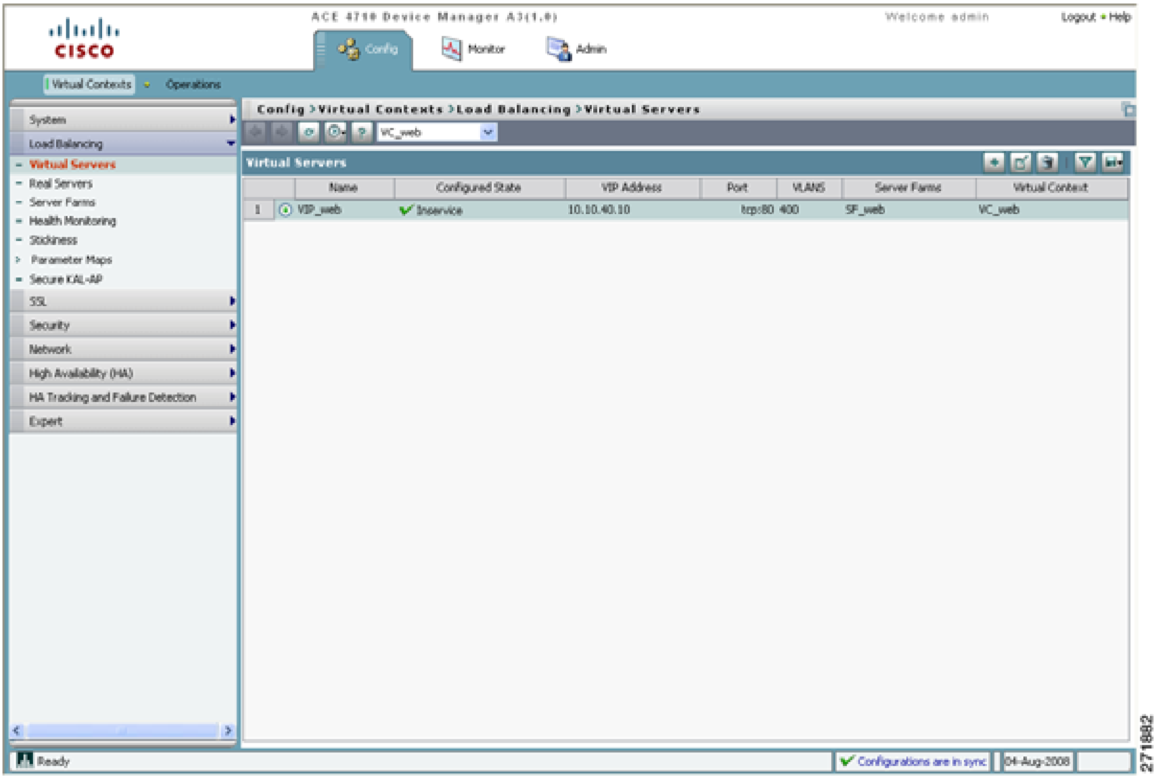
- Name: RS_web3
- IP Address: 10.10.50.12
- Port: 80

For RS_web4, enter:

- Name: RS_web4
- IP Address: 10.10.50.13
- Port: 80

Step 10 Click **Deploy Now** at the bottom of the window to save your settings for the virtual server. The Virtual Servers pane reappears (Figure 6-5). The newly configured virtual server appears in the pane and is in the Inservice state, which means that the virtual server is in use as a destination for server load balancing.

Figure 6-5 Virtual Servers Pane with a Virtual Server Created



Configuring Layer 7 Server Load Balancing Using the CLI

You can configure Layer 7 server load balancing using the command-line interface (CLI). This section contains the following topics:

- Configuring Real Servers
- Creating a Server Farm
- Creating a Virtual Server Traffic Policy

Configuring Real Servers

Configure real servers on the ACE using the CLI by following these steps:

Step 1 Verify that you are operating in the desired context by checking the CLI prompt. If necessary, change to the correct context.

```
host1/Admin# changeto VC_web
host1/VC_web#
```

Step 2 Enter configuration mode.

```
host1/VC_web# config
```

Step 3 Create a real server named RS_web1 as type host (the default).

```
host1/VC_web(config)# rserver RS_web1
host1/VC_web(config-rserver-host)#
```

Step 4 Enter a description of the real server.

```
host1/VC_web(config-rserver-host)# description content server web-one
```

Step 5 Assign the real server with an IP address of 10.10.50.10.

```
host1/VC_web(config-rserver-host)# ip address 10.10.50.10
```

Step 6 Place the real server in service and exit configuration mode.

```
host1/VC_web(config-rserver-host)# inservice
host1/VC_web(config-rserver-host)# exit
host1/VC_web(config)#
```

Step 7 Add three more real servers by repeating Steps 3 through 6, using the following real server names, descriptions, and IP addresses.

For RS_web2, enter:

- Name: RS_web2
- Description: content server web-two
- IP Address: 10.10.50.11

For RS_web3, enter:

- Name: RS_web3
- Description: content server web-three
- IP Address: 10.10.50.12

For RS_web4, enter:

- Name: RS_web4
- Description: content server web-four
- IP Address: 10.10.50.13

Step 8 Display the configuration of the real servers.

```
host1/VC_web(config)# do show running-config rserver
```

Creating a Server Farm

After you create and configure the real servers, you can create a server farm and associate the real servers with it. Create a server farm by following these steps:

Step 1 Create a server farm of type host (the default) named SF_web.

```
host1/VC_web(config)# serverfarm SF_web
host1/VC_web(config-sfarm-host)#
```

Step 2 Associate real server RS_web1 to the server farm through port 80.

```
host1/VC_web(config-sfarm-host)# rserver RS_web1 80
host1/VC_web(config-sfarm-host-rs)#
```

Step 3 Place the real server in service within the server farm and exit configuration mode.

```
host1/VC_web(config-sfarm-host-rs)# inservice
host1/VC_web(config-sfarm-host-rs)# exit
host1/VC_web(config-sfarm-host)#
```

> **Note** Before you can start sending connections to a real server in a server farm, you must place it in service. Otherwise, the ACE considers it out of service and the server farm cannot receive or respond to client requests.

Step 4 Similarly, associate the RS_web2, RS_web3, and RS_web4 real servers with the SF_web server farm.

```
host1/VC_web(config-sfarm-host)# rserver RS_web2 80
host1/VC_web(config-sfarm-host-rs)# inservice
host1/VC_web(config-sfarm-host-rs)# exit
host1/VC_web(config-sfarm-host)# rserver RS_web3 80
host1/VC_web(config-sfarm-host-rs)# inservice
host1/VC_web(config-sfarm-host-rs)# exit
host1/VC_web(config-sfarm-host)# rserver RS_web4 80
host1/VC_web(config-sfarm-host-rs)# inservice
host1/VC_web(config-sfarm-host-rs)# exit
```

Step 5 Exit server farm configuration mode.

```
host1/VC_web(config-sfarm-host)# exit
host1/VC_web(config)#
```

Step 6 Display the information for the real servers and verify that the real servers appear as operational (even though network connectivity has not been established).

```
host1/VC_web(config)# do show rserver RS_web1
host1/VC_web(config)# do show rserver RS_web2
host1/VC_web(config)# do show rserver RS_web3
host1/VC_web(config)# do show rserver RS_web4
```

Step 7 Display how the ACE populates the ARP table with the real servers.

```
host1/VC_web(config)# do show arp
```

Creating a Virtual Server Traffic Policy

You can create a virtual server traffic policy on the ACE by following these steps:

Step 1 Create a Layer 7 server load-balancing policy map named PM_LB to match the class maps in the order in which they occur for load balancing.

```
host1/VC_web(config)# policy-map type loadbalance first-match PM_LB
host1/VC_web(config-pmap-lb)#
```

Note The ACE uses a class map to specify a series of flow match criteria (traffic classifications). The ACE uses a policy map to define a series of actions (functions) that you want applied to a set of classified inbound traffic.

Step 2 For a simple load-balancing policy, assign the ACE default class map which contains an implicit match any statement to match any traffic classification.

```
host1/VC_web(config-pmap-lb)# class class-default
host1/VC_web(config-pmap-lb-c)#
```

Step 3 Add the server farm SF_web to the Layer 7 server load-balancing policy map and exit configuration mode.

```
host1/VC_web(config-pmap-lb-c)# serverfarm SF_web
host1/VC_web(config-pmap-c)# exit
host1/VC_web(config-pmap)# exit
host1/VC_web(config)#
```

Step 4 Create a Layer 3 and Layer 4 load-balancing class map VS_web.

```
host1/VC_web(config)# class-map VS_web
host1/VC_web(config-cmap)#
```

Step 5 Define a match statement for the IP address 10.10.40.10 for any IP protocol and exit configuration mode.

```
host1/VC_web(config-cmap)# match virtual-address 10.10.40.10
255.255.255.0 tcp eq 80
host1/VC_web(config-cmap)# exit
host1/VC_web(config)#
```

Step 6 Create a Layer 3 and Layer 4 multi-match policy map to direct classified incoming requests to the load-balancing policy map.

```
host1/VC_web(config)# policy-map multi-match PM_multi_match
host1/VC_web(config-pmap)#
```

Step 7 Associate the Layer 3 and Layer 4 class map VS_web with the policy map.

```
host1/VC_web(config-pmap)# class VS_web
host1/VC_web(config-pmap-c)#
```

Step 8 Associate the Layer 7 load-balancing policy map PM_LB with the Layer 3 and Layer 4 policy map.

```
host1/VC_web(config-pmap-c)# loadbalance policy PM_LB
host1/VC_web(config-pmap-lb-c)#
```

Step 9 Enable a VIP for load-balancing operations and exit configuration mode.

```
host1/VC_web(config-pmap-lb-c)# loadbalance vip inservice
host1/VC_web(config-pmap-c)# exit
host1/VC_web(config-pmap)# exit
host1/VC_web(config)#
```

Step 10 Access the interface to which you want to apply the multi-match policy map.

```
host1/VC_web(config)# interface vlan 400
host1/VC_web(config-if)#
```

Step 11 Apply the multi-match policy map PM_multi_match.

```
host1/VC_web(config-if)# service-policy input PM_multi_match
host1/VC_web(config-if)# exit
host1/VC_web(config)#
```

Step 12 Save the running configuration to the startup configuration.

```
host1/VC_web(config)# do copy running-config startup-config
```

Step 13 Display the service policy state for the PM_multi_match policy map.

```
host1/VC_web(config)# do show service-policy PM_multi_match
```

In this chapter, you have configured a virtual server for load-balancing HTTP traffic. In the next chapter, you will configure a load-balancing predictor to forward client requests to the appropriate real servers.

CHAPTER 7

Configuring a Load-Balancing Predictor

This chapter describes how to configure a load-balancing predictor on the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following sections:

- Overview
- Configuring a Hash Header Predictor Using the Device Manager GUI
- Configuring a Hash Header Predictor Using the CLI

Overview

After reading this chapter, you should have a basic understanding of how the ACE appliance selects a real server for a client request using a predictor and how to configure a hash header predictor as an example.

When there is a client request for web services, the ACE selects a server that can successfully fulfill the client request in the shortest amount of time without overloading either the individual server or the server farm.

The ACE makes load-balancing choices using a predictor. When you configure a predictor, you define the series of checks and calculations that the ACE will perform to determine which real server can best service a client request.

For each server farm, you can configure one of several predictor types to allow the ACE to select an appropriate server. Two common predictor types include the following:

- Round-robin—Selects a server from the list of real servers based on weighted server capacity. A weight can be assigned to each real server based on its connection capacity in relation to the other servers in a server farm. Servers with higher weight values receive a proportionally higher number of connections than servers with lower weight values. For example, a server with a weight of 5 would receive five connections for every one connection received by a server with a weight of 1. Also known as weighted round-robin, this is the default predictor.
- Hash header—Selects a server using a hash value based on the HTTP header name.

For a complete list of predictor types that the ACE supports and how to configure them, see the *Cisco 4700 Series Application Control Engine Appliance Administration Guide*.

You can configure a server load-balancing predictor by following these steps:

Step 1 Choose a server farm.

Step 2 Choose a predictor type and its parameters.

Step 3 Deploy the configuration.

This chapter describes how to configure a hash header predictor for the server farm that was created in Chapter 6, "Configuring Server Load Balancing," as illustrated in Figure 6-1. You can use either the ACE Device Manager GUI or the CLI.

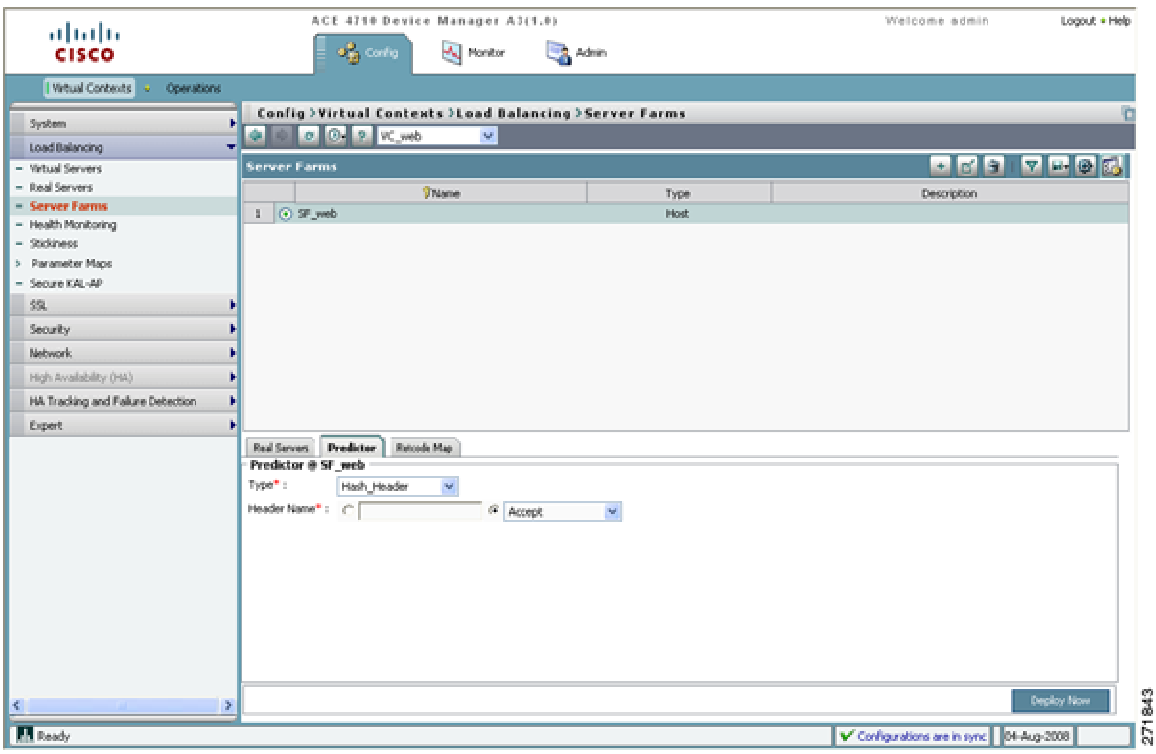
Configuring a Hash Header Predictor Using the Device Manager GUI

You can configure a hash header predictor using the ACE Device Manager GUI by following these steps:

Step 1 Choose **Config** > **Virtual Contexts**. Choose context **VC_web**.

Step 2 Choose **Load Balancing** > **Server Farms**. The Server Farms pane appears (Figure 7-1).

Figure 7-1 Configuring a Predictor



Step 3 Choose **SF_web**.

Step 4 Choose the **Predictor** tab.

Step 5 Choose **Hash_Header** for the predictor Type.

Step 6 Choose **Accept** for the Header Name.

Step 7 Assign the hash header predictor to server farm SF_web by clicking **Deploy Now**.

Configuring a Hash Header Predictor Using the CLI

You can configure a hash header predictor using the CLI by following these steps:

Step 1 Verify that you are operating in the desired context by checking the CLI prompt. If necessary, change to the correct context.

```
host1/Admin# changeto VC_web
host1/VC_web#
```

Step 2 Enter configuration mode for SF_web.

```
host1/VC_web# config
host1/VC_web(config)# serverfarm SF_web
host1/VC_web(config-sfarm-host)#
```

Step 3 Configure a hash header predictor.

```
host1/VC_web(config-sfarm-host)# predictor hash header Accept
```

Step 4 Display the predictor configuration information.

```
host1/VC_web(config-sfarm-host)# exit
host1/VC_web(config)# exit
host1/VC_web# show running-config serverfarm
```

In this chapter, you have configured a hash header predictor for your server load balancing. Next, you will configure server persistence by using the stickiness feature.

CHAPTER 8

Configuring Server Persistence Using Stickiness

This chapter describes how to configure server persistence using stickiness on the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following sections:

- Overview
- Configuring HTTP Cookie Stickiness Using the Device Manager GUI
- Configuring HTTP Cookie Stickiness Using the CLI

Overview

After reading this chapter, you should have a basic understanding of how the ACE appliance provides server persistence using stickiness, and how to configure HTTP cookie stickiness.

When customers visit an e-commerce site, they usually start by browsing the site. Depending on the application, the site may require that the client become persisted (stuck) to one server as soon as the initial connection is established, or the application may require this action only when the client starts to create a transaction, such as when building a shopping cart.

For example, after the client adds items to a shopping cart, it is important that all subsequent client requests are directed to the same real server so that all the items are contained in one shopping cart on one server. An instance of a customer's shopping cart is typically local to a particular server rather than duplicated across multiple servers.

E-commerce applications are not the only types of applications that require a sequence of client requests to be directed to the same real server. Any web applications that maintain client information may require stickiness, such as banking and online trading applications, or FTP and HTTP file transfers.

The ACE can be configured so that the same client can maintain multiple, simultaneous, or subsequent TCP or IP connections with the same real server for the duration of a session. This session persistence capability of the ACE is called stickiness. A session is defined as a series of transactions between a client and a server over some finite period of time (from several minutes to several hours).

Depending on the configured server load-balancing policy, the ACE sticks a client to an appropriate server after the ACE determines which load-balancing method to use. If the ACE determines that a client is already stuck to a particular server, then the ACE sends that client request to that server, regardless of the load-balancing criteria. If the ACE determines that the client is not stuck to a particular server, it applies the normal load-balancing rules to the request.

To determine how a particular client is stuck to a specific web server and how an application distinguishes each client or a group of clients, the ACE supports the following sticky methods:

- Source and/or destination IP address—For stickiness, you can use the source IP address, the destination IP address, or both to uniquely identify individual clients and their requests based on their IP net masks. However, if an enterprise or service provider uses a mega-proxy (a free, anonymous web proxy service) to establish client connections to the Internet, the source IP address is not a reliable indicator of the true source of the request. In this case, you can use another sticky method to ensure session persistence.
- Cookie—Client cookies uniquely identify clients to the ACE and to the servers that provide content. A cookie is a small data structure within the HTTP header that a server uses to deliver data to a web client, with the request that the client store the information. This information might include items that users have added to their shopping carts or travel dates that they have chosen. When the ACE examines a request for content and determines that the

content is sticky, it examines any cookie or URL present in the content request. The ACE uses the information in the cookie or URL to direct the content request to the appropriate server.

- Hypertext Transfer Protocol (HTTP) header—You can specify a header offset to provide stickiness based on a unique portion of the HTTP header.

The e-commerce application often dictates which of these methods is appropriate for a particular e-commerce application.

The ACE uses sticky groups for stickiness attributes. These attributes include the sticky method, timeout, replication, and attributes related to a particular sticky method.

To track sticky connections, the ACE uses a sticky table with information about sticky groups, sticky methods, sticky connections, and real servers. The ACE uses a configurable timeout mechanism to age out sticky table entries. When an entry times out, it becomes eligible for reuse. High connection rates may cause the premature aging out of sticky entries. In this case, the ACE reuses the entries that are closest to expiration first.

Entries in the sticky table can be either dynamic (generated by the ACE as needed) or static (configured). When you create a static sticky entry, the ACE places the entry in the sticky table immediately, and it remains in the sticky database until you remove it from the configuration.

You can configure stickiness by following these steps:

Step 1 Ensure that resources are allocated for stickiness.

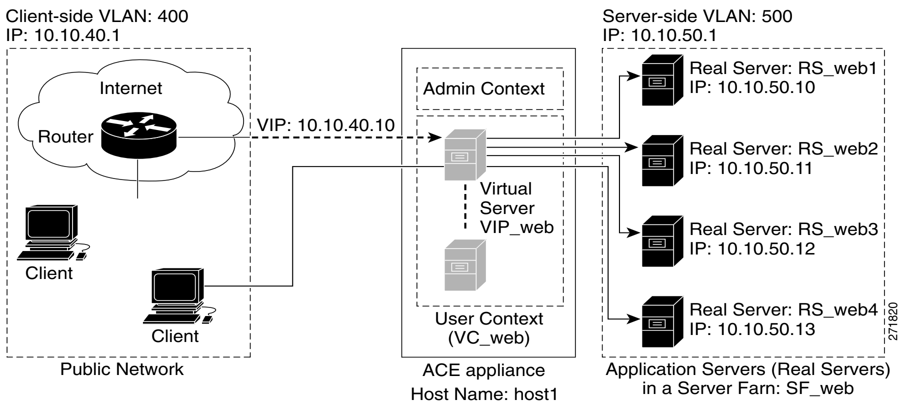
Step 2 Create a sticky group.

Step 3 Associate the sticky group with a Layer 7 server load-balancing action of a virtual server.

Step 4 Deploy the configuration.

Figure 8-1 illustrates that in a server load-balancing environment, requests from a client are stuck to real server RS_web4 in a session.

Figure 8-1 Client Requests Stuck to a Server



This chapter describes how to configure stickiness using the HTTP cookie sticky method. For information on how to configure stickiness using the IP address and HTTP header methods, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide.*

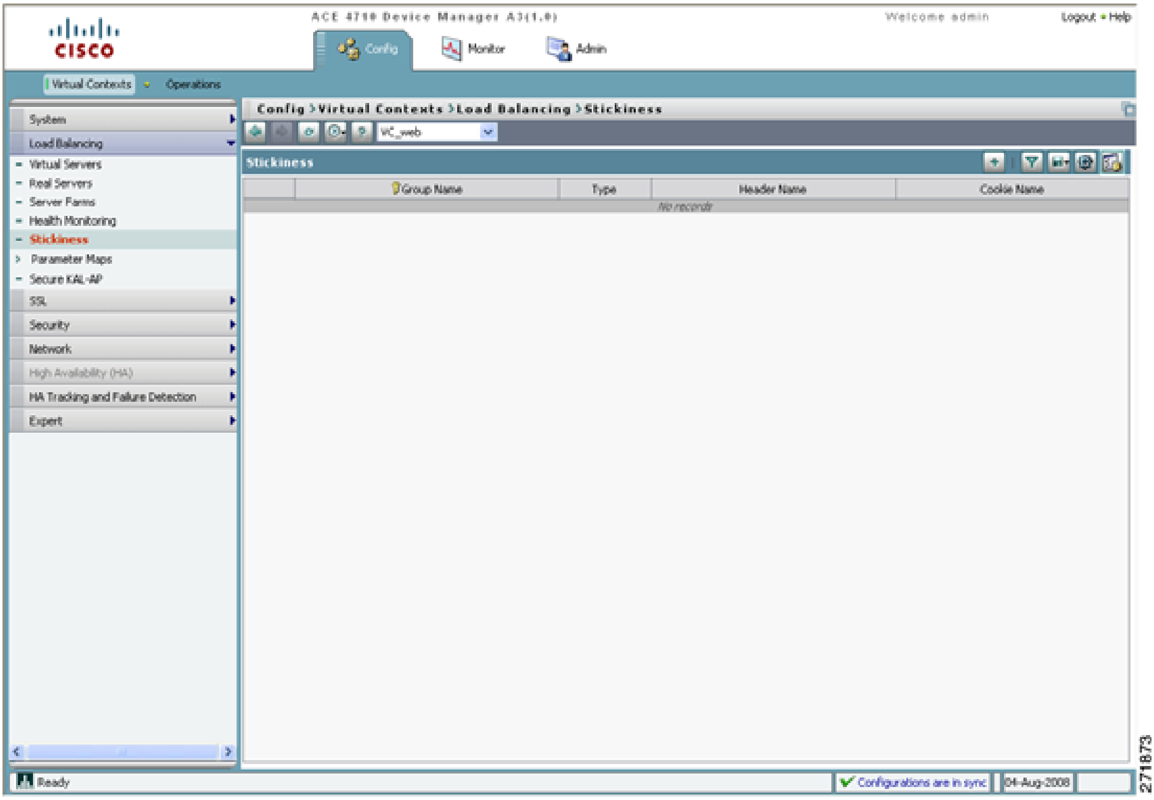
Configuring HTTP Cookie Stickiness Using the Device Manager GUI

You can configure HTTP cookie stickiness using the GUI by following these steps:

Step 1 Make sure that the context in which you are configuring the sticky group is associated with a resource class that allocates resources to stickiness. See the "Creating a Resource Class" section in Chapter 3.

Step 2 Choose **Load Balancing** > **Stickiness**. The Stickiness pane appears (Figure 8-2).

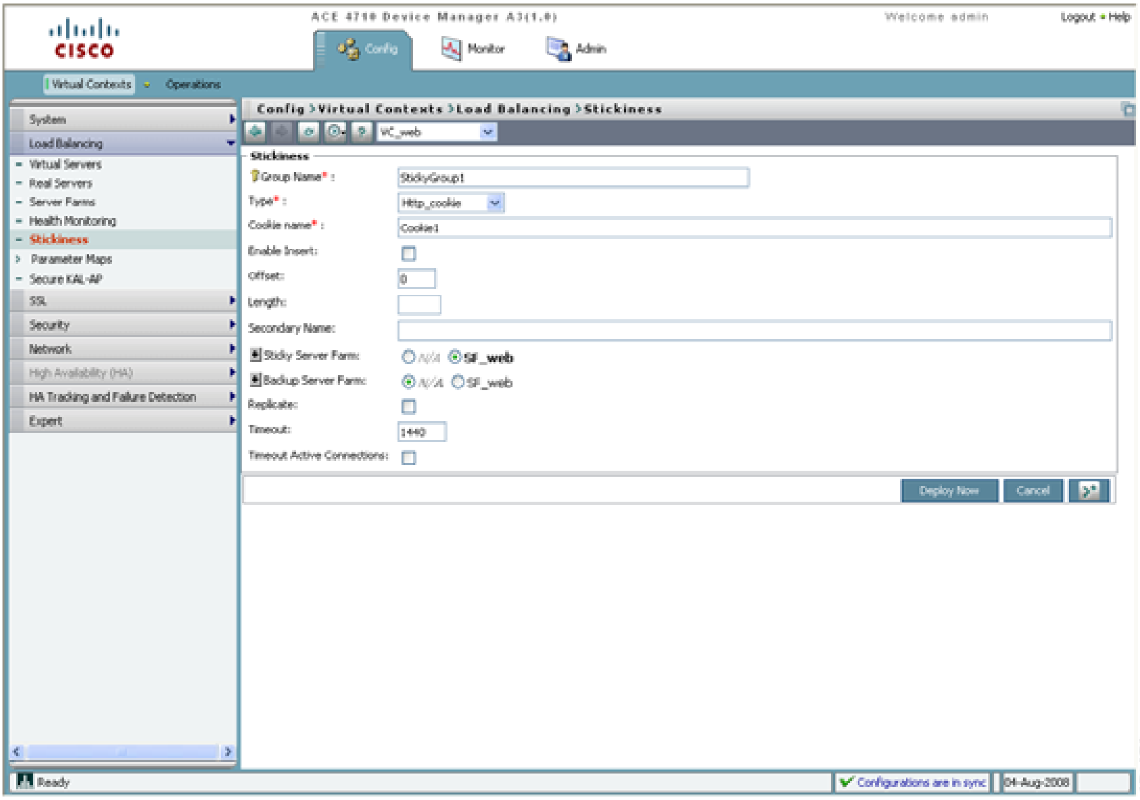
Figure 8-2 Stickiness Pane



Step 3 Choose the **VC_web** context.

Step 4 Add a new sticky group by clicking **Add**. The Stickiness configuration window appears (Figure 8-3).

Figure 8-3 Stickiness Configuration Window



Step 5 Enter the following attributes for the new sticky group. Leave the remaining attributes blank or with their default values.

- Group Name: StickyGroup1
- Type: Http_cookie
- Cookie name: Cookie1
- Sticky Server Farm: SF_web

Step 6 Add the new sticky group to the Stickiness pane by clicking **Deploy Now.**

Configuring HTTP Cookie Stickiness Using the CLI

You can configure HTTP cookie stickiness using the CLI by following these steps:

Step 1 Verify that you are operating in the desired context by checking the CLI prompt. If necessary, change to the correct context.

```
host1/Admin# changeto VC_web
host1/VC_web#
```

Step 2 Enter configuration mode.

```
host1/VC_web# config
host1/VC_web(config)#
```

Step 3 Create an HTTP-cookie-type sticky group and enter the cookie configuration mode.

```
host1/VC_web(config)# sticky http-cookie Cookie1 StickyGroup1
host1/VC_web(config-sticky-cookie)#
```

Step 4 Configure a timeout for HTTP cookie stickiness.

```
host1/VC_web(config-sticky-cookie)# timeout 1440
```

Step 5 Associate a server farm with the sticky group and exit configuration mode.

```
host1/VC_web(config-sticky-cookie)# serverfarm SF_web
host1/VC_web(config-sticky-cookie)# exit
host1/VC_web(config)# exit
host1/VC_web#
```

Step 6 Display the HTTP cookie configuration.

```
host1/VC_web# show running-config sticky
```

In this chapter, you have configured a sticky group using the HTTP-cookie method. In the next chapter, you will configure SSL security.

CHAPTER 9

Configuring SSL Security

This chapter describes how to configure SSL on the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following sections:

Overview

After reading this chapter, you should have a basic understanding of how the ACE appliance provides SSL security for your network and how to configure SSL termination, in which the ACE operates as an SSL server.

SSL configuration in an ACE establishes and maintains a SSL session between the ACE and another device. It provides for secure data transactions between a client and a server. SSL provides authentication, encryption, and data integrity in a Public Key Infrastructure (PKI), a set of policies and procedures that establishes a secure information exchange between devices.

In SSL, data is encrypted using one or more symmetric keys that are known only by the two endpoints in the transaction. In a key exchange, one device generates the symmetric key and then encrypts it using an asymmetric encryption scheme before transmitting the key to the other device.

Asymmetric encryption requires each device to have a unique key pair consisting of a public key and a private key. A private key is an encryption/decryption key known only to the parties exchanging the messages. A public key is a value provided by some designated authority as an encryption key that, combined with a private key derived from the public key, can be used to effectively encrypt messages and digital signatures. The two keys are mathematically related; data that is encrypted using the public key can only be decrypted using the corresponding private key, and vice versa.

SSL facilitates client and server authentication through the use of digital certificates. Digital certificates are a form of digital identification to prove the identity of the server to the client, or optionally, the client to the server. A certificate ensures that the identification information is correct and the public key embedded in it actually belongs to the client or server.

A Certificate Authority (CA) issues digital certificates in the context of a PKI. CAs are trusted authorities that sign certificates to verify their authenticity. As the certificate issuer, the CA uses its private key to sign the certificate. Upon receiving a certificate, a client uses the issuer's public key to decrypt and verify the certificate signature to ensure that the certificate was actually issued and signed by an authorized entity.

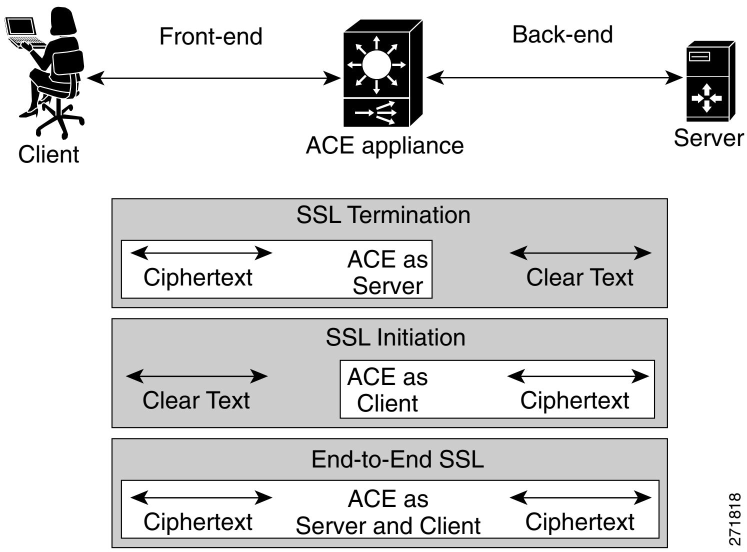
If you do not have a certificate and the corresponding key pair, you can use the ACE to generate a key pair and a certificate signing request (CSR) to apply for a certificate from a CA. The CA signs the CSR and returns the authorized digital certificate to you. The ACE supports import, export, and other management functions to manage the various certificates and key pair files within each context.

The client and server use the SSL handshake protocol to establish an SSL session between the two devices. During the handshake, the client and server negotiate the SSL parameters that they will use during the secure session. During the SSL handshake, the ACE uses an SSL proxy service, which includes the configuration of SSL session parameters, an RSA key pair, and a matching certificate.

The ACE applies SSL session parameters to an SSL proxy service. Creating an SSL parameter map allows you to apply the same SSL session parameters to different proxy services. The SSL session parameters include timeouts, close protocol behavior, and SSL version—SSL 3 and/or Transport Layer Security (TLS) 1. For more information on these parameters, see the *Cisco 4700 Series Application Control Engine Appliance SSL Configuration Guide*.

You can configure the ACE to act as a client or a server during an SSL session by defining operational attributes such as SSL session parameters, SSL key pairs and certificates, and traffic characteristics. When the traffic characteristics match the settings specified in the operational attributes, the ACE executes the actions associated with the SSL proxy service. Figure 9-1 shows the three basic SSL configurations in which the ACE is used to encrypt and decrypt data between the client and the server: SSL termination, SSL initiation, and end-to-end SSL.

Figure 9-1 ACE SSL Configurations



In SSL termination, an ACE context is configured for a front-end application in which the ACE operates as an SSL server that communicates with a client. When you define the flow between an ACE and a client, the ACE operates as a virtual SSL server by adding security services between a web browser (the client) and the HTTP connection (the server).

All inbound SSL flows that come from a client terminate at the ACE. After the connection is terminated, the ACE decrypts the ciphertext (encrypted content) from the client and sends the data as clear text (unencrypted content) to an HTTP server. For information about configuring the ACE for SSL termination, see the "Configuring SSL Termination" section.

In SSL initiation, an ACE context is configured for a back-end application in which the ACE operates as a client that communicates with an SSL server. When you define the flow between an ACE and an SSL server, the ACE operates as a client and initiates the SSL session. SSL initiation enables the ACE to receive clear text from a client and then establish an SSL session with an SSL server, joining the client and SSL server connections.

The ACE encrypts the clear text that it receives from the client and sends the data as ciphertext to an SSL server. The SSL server can either be an ACE configured for SSL termination (a virtual SSL server) or a real SSL server (web server). On the outbound flow from the SSL server, the ACE decrypts the ciphertext from the server and sends clear text back to the client. For more information on configuring the ACE for SSL initiation, see the *Cisco 4700 Series Application Control Engine Appliance SSL Configuration Guide*.

In end-to-end SSL, an ACE context is configured for both SSL termination and SSL initiation. You configure the ACE for end-to-end SSL when you have an application that requires secure SSL channels between the client and the ACE, and between the ACE and the SSL server.

For example, a transaction between banks requires end-to-end SSL to protect all financial information exchanged. End-to-end SSL also allows the ACE to insert load-balancing and security information into the data. The ACE decrypts the ciphertext that it receives and inserts load-balancing and firewall information into the clear text. The ACE then re-encrypts the data and passes the ciphertext to its intended destination. For more information on configuring the ACE for end-to-end SSL initiation, see the *Cisco 4700 Application Control Engine Series Appliance SSL Configuration Guide*.

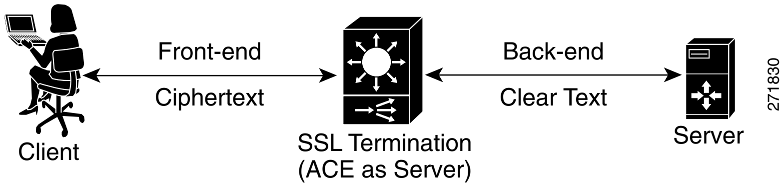
Configuring SSL Termination

SSL termination occurs when the ACE, acting as an SSL proxy server, terminates an SSL connection from a client and then establishes a TCP connection to an HTTP server. When the ACE terminates the SSL connection, it decrypts the ciphertext from the client and transmits the data as clear text to the HTTP server.

Figure 9-2 shows the following network connections in which the ACE terminates the SSL connection with the client:

- Client to ACE—An SSL connection exists between the client and the ACE acting as an SSL proxy server.
- ACE to Server—A TCP connection exists between the ACE and the HTTP server.

Figure 9-2 SSL Termination



Before configuring the ACE for an SSL operation, you must first configure it for server load balancing. To configure your ACE for server load balancing, see Chapter 6, "Configuring Server Load Balancing."

SSL termination is a Layer 3 and Layer 4 application because it is based on the destination IP address of the inbound traffic flow from the client. When configuring a policy map for SSL termination, you associate the following elements:

- The SSL proxy service, including SSL session parameters, certificate, and key pair.
- The virtual SSL server IP address that the destination IP address of the inbound traffic must match (a class map). When a match occurs, the ACE negotiates with the client to establish an SSL connection.

You can configure the ACE for SSL termination by following these steps:

Step 1 Import a key file with a key pair.

Step 2 Import a certificate that matches the imported key pair.

Step 3 Configure a parameter map.

Step 4 Configure an SSL proxy service using the key pair, certificate, and parameter map.

Step 5 Create a virtual server for SSL termination using the SSL proxy service.

Step 6 Deploy the configuration.

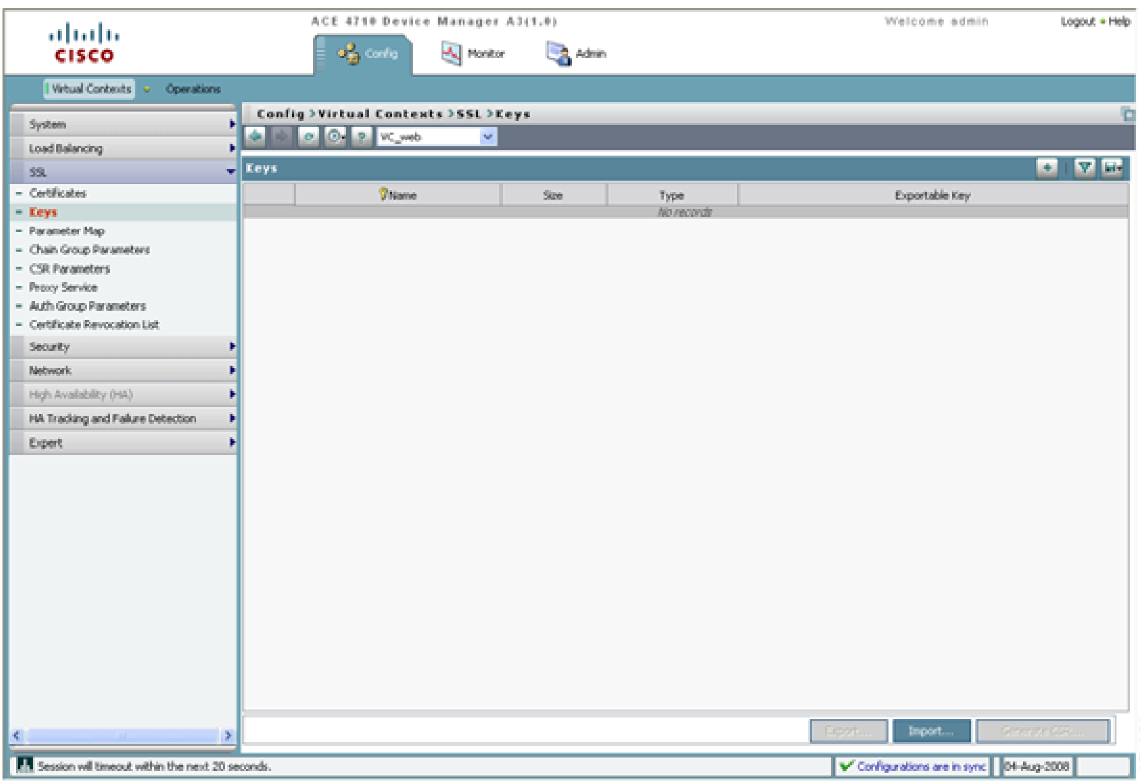
This chapter describes how to configure the ACE for SSL termination using either the Device Manager GUI or the CLI.

Configuring the ACE for SSL Termination Using the Device Manager GUI

You can configure the ACE for SSL termination using the Device Manager GUI by following these steps:

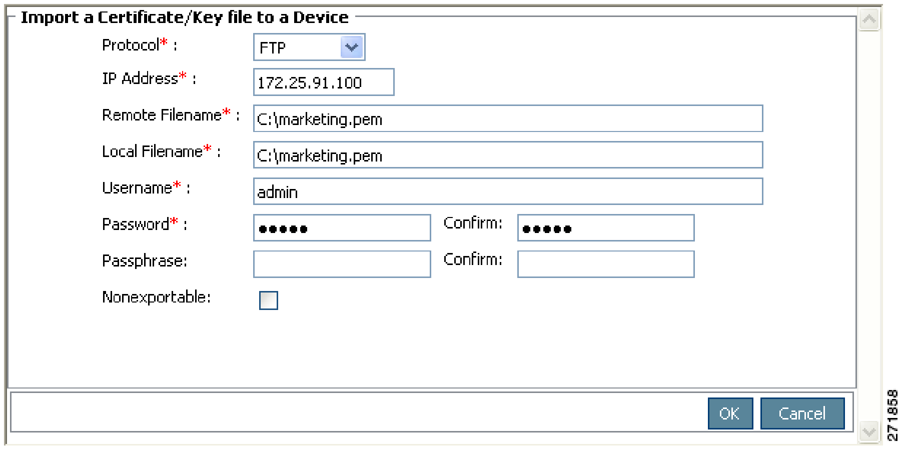
Step 1 Choose the user context **VC_web**, and then choose **SSL > Keys**. The Keys pane appears (Figure 9-3).

Figure 9-3 Keys Pane



Step 2 Click **Import…** to import a key file. The Import a Certificate/Key File to a Device window appears (Figure 9-4).

Figure 9-4 Import a Certificate/Key File to a Device Window



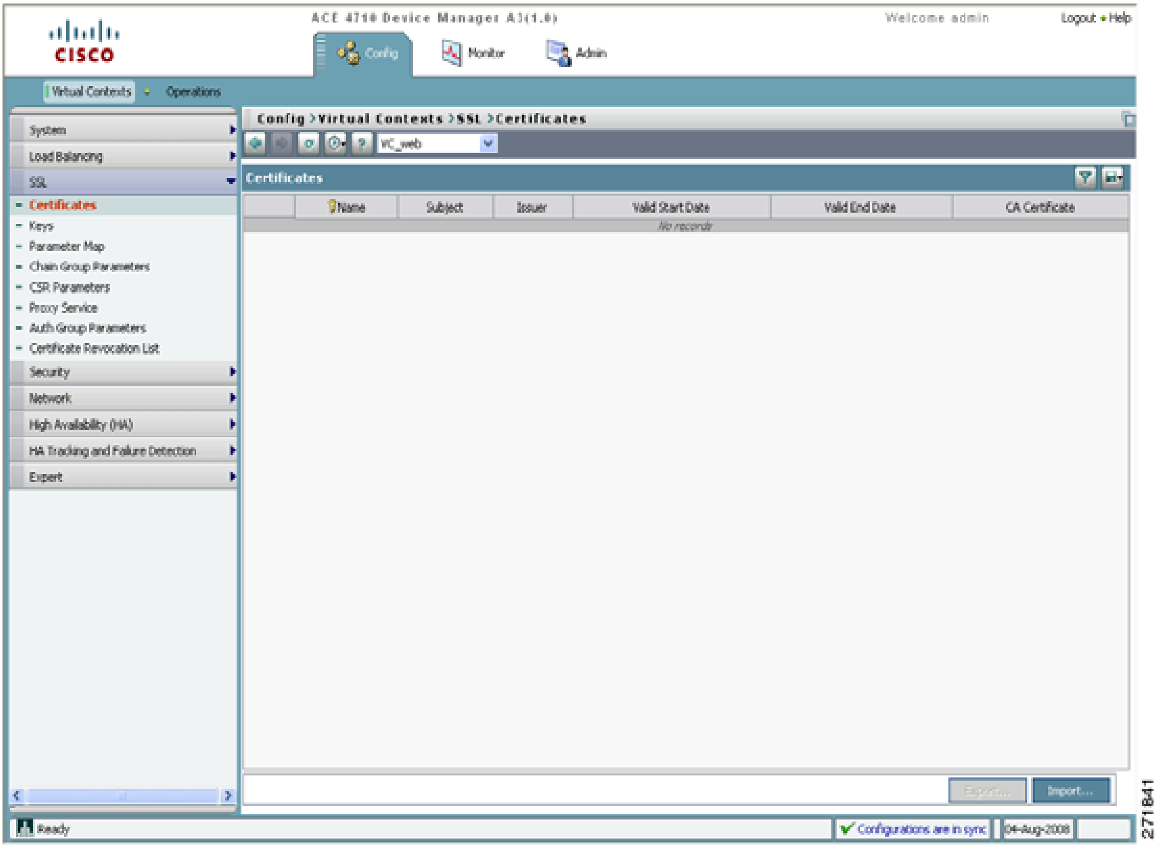
Enter the following parameters. Leave the remaining parameters blank or with their default values.

- Protocol: FTP
- IP Address: 172.25.91.100 (in order for this to work, you should use an IP address where you can access the remote key file)
- Remote Filename: C:\marketing.pem
- Local Filename: C:\marketing.pem
- Username: Admin
- Password: (password for your FTP server)
- Confirm: (retype the password for your FTP server)

Step 3 Click **OK** to import the key file.

Step 4 Choose **SSL** > **Certificates**. The Certificates pane appears (Figure 9-5).

Figure 9-5 Certificates Pane



Step 5 Click **Import...** to import a certificate file. The Import a Certificate/Key File to a Device window reappears. Enter the following parameters. Leave the remaining parameters blank or with their default values.

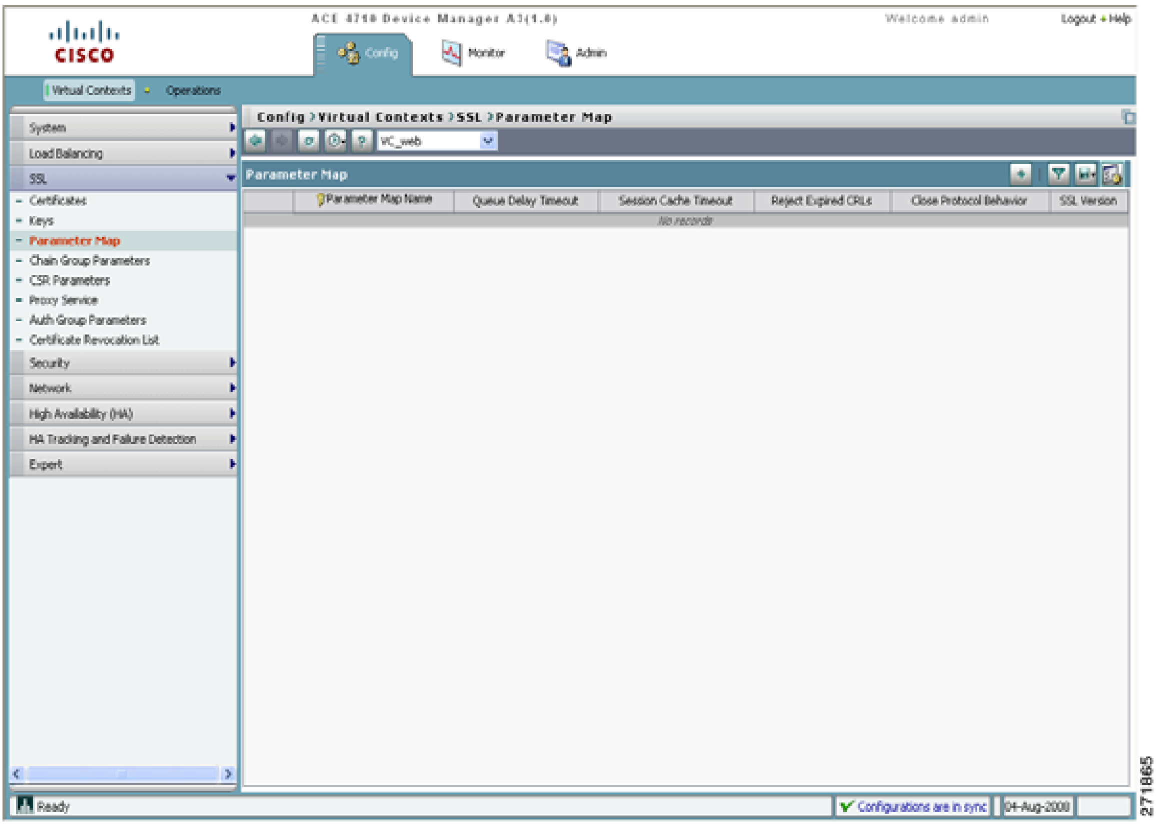
- Protocol: FTP
- IP Address: 172.25.91.100 (in order for this to work, you should use an IP address where you can access the certificate file)
- Remote Filename: C:\marketing_cert.pem
- Local Filename: C:\marketing_cert.pem
- Username: Admin

- Password: (password for your FTP server)
- Confirm: (retype the password for your FTP server)

Step 6 Click **OK** to import the certificate file.

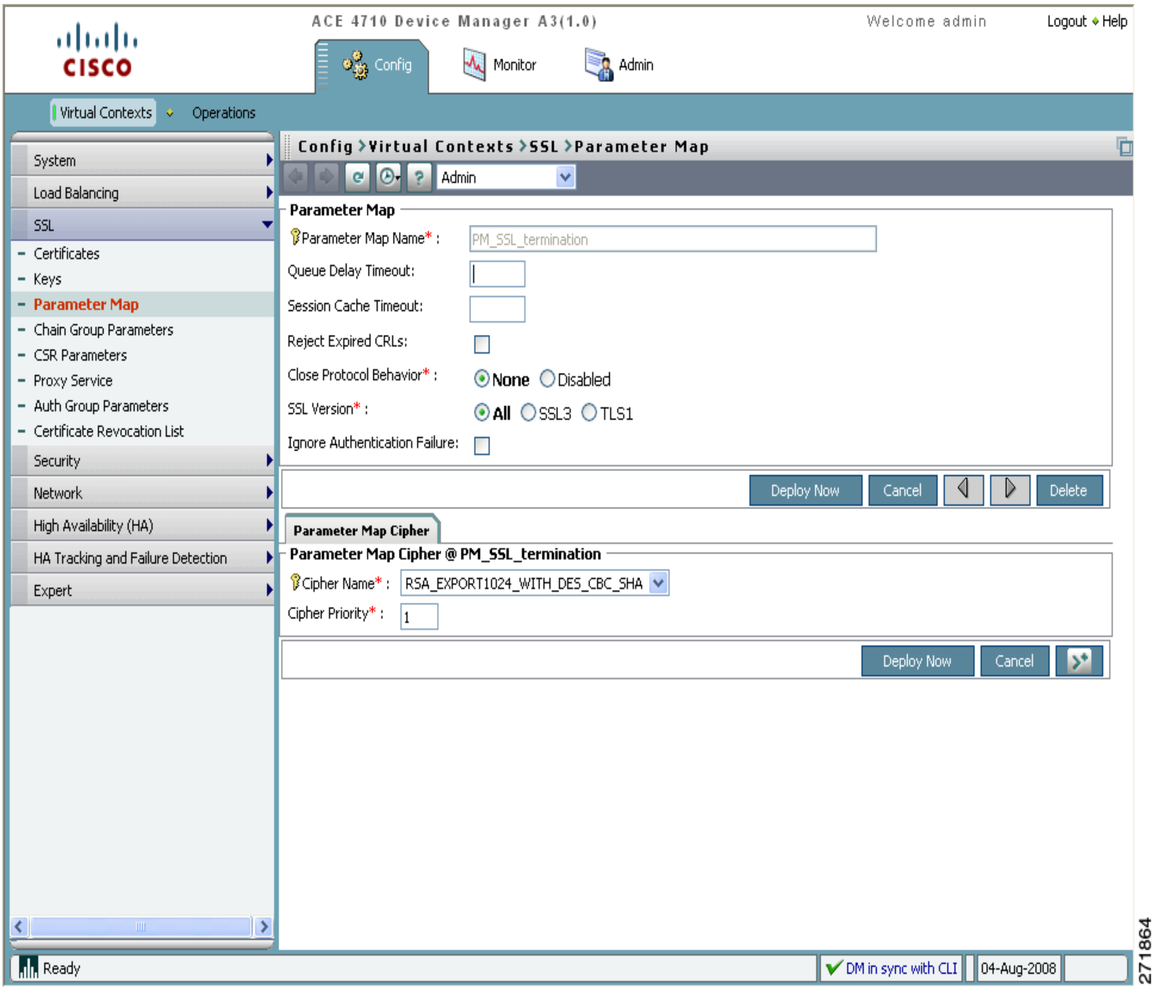
Step 7 Choose **SSL** > **Parameter Map**. The Parameter Map pane appears (Figure 9-6).

Figure 9-6 Parameter Map Pane



Step 8 Click **Add** to create a parameter map. The Parameter Map window appears (Figure 9-7).

Figure 9-7 Parameter Map Window



Step 9 Enter the following parameter. Leave the remaining parameters blank or with their default values.

- Parameter Map Name: PM_SSL_termination

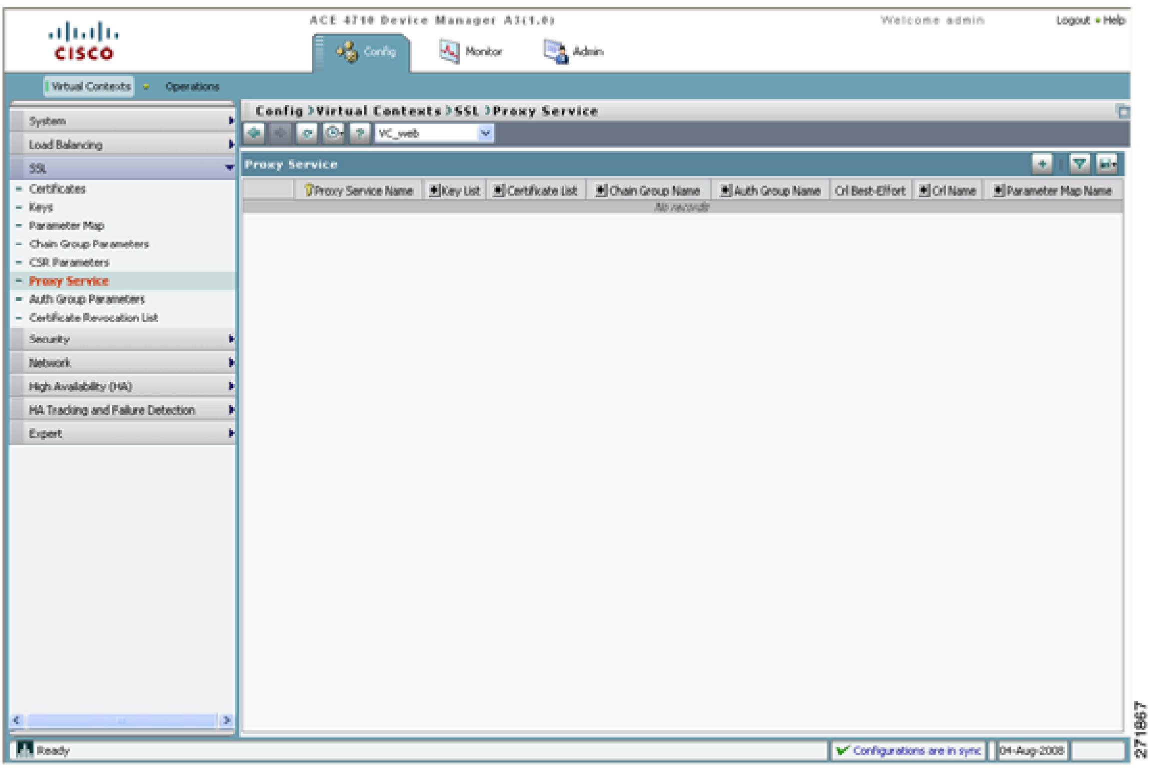
Step 10 Click **Deploy Now** to deploy the parameter map on the ACE appliance. The Parameter Map Cipher pane appears.

Step 11 Select **Add** in the Parameter Map Cipher pane (Figure 9-7).

Step 12 Accept the defaults and click **Deploy Now** in the Parameter Map Cipher pane to add a cipher to the parameter map.

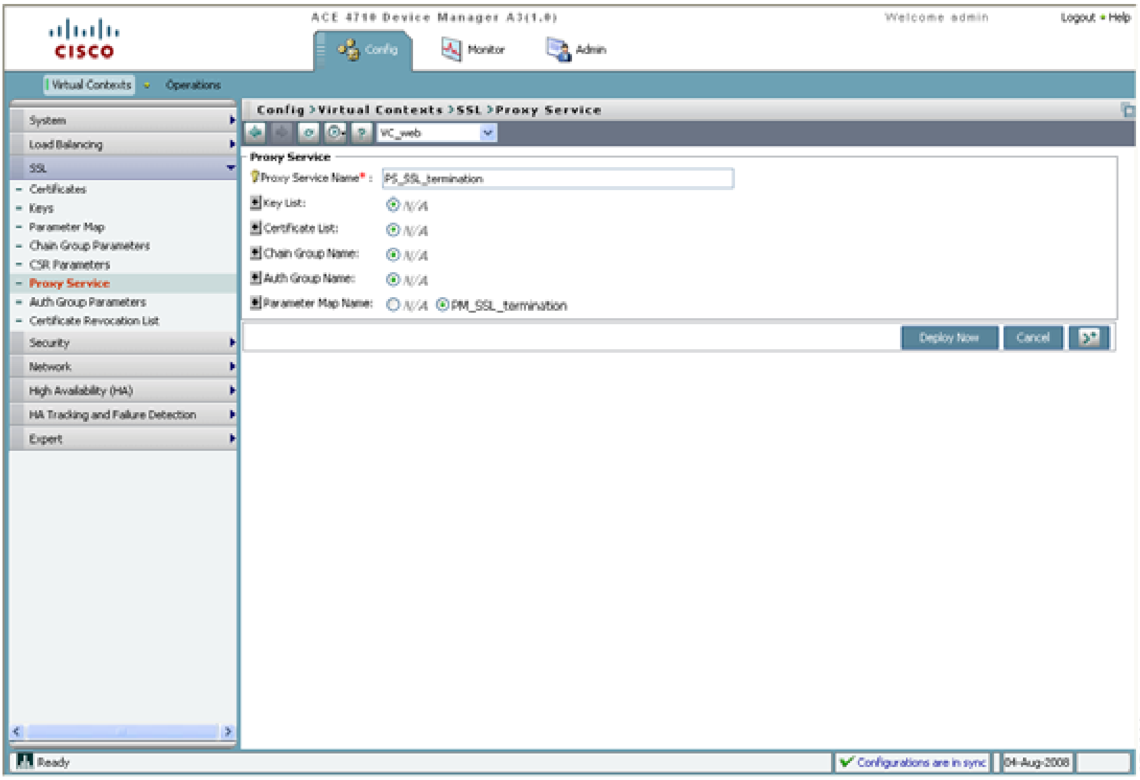
Step 13 Create an SSL proxy service by choosing **SSL > Proxy Service**. The Proxy Service pane appears (Figure 9-8).

Figure 9-8 Proxy Service Pane



Step 14 Click **Add** to create a proxy service. The Proxy Service window appears (Figure 9-9).

Figure 9-9 Proxy Service Window



Step 15 Enter the following parameters. Leave the remaining parameters blank or with their default values.

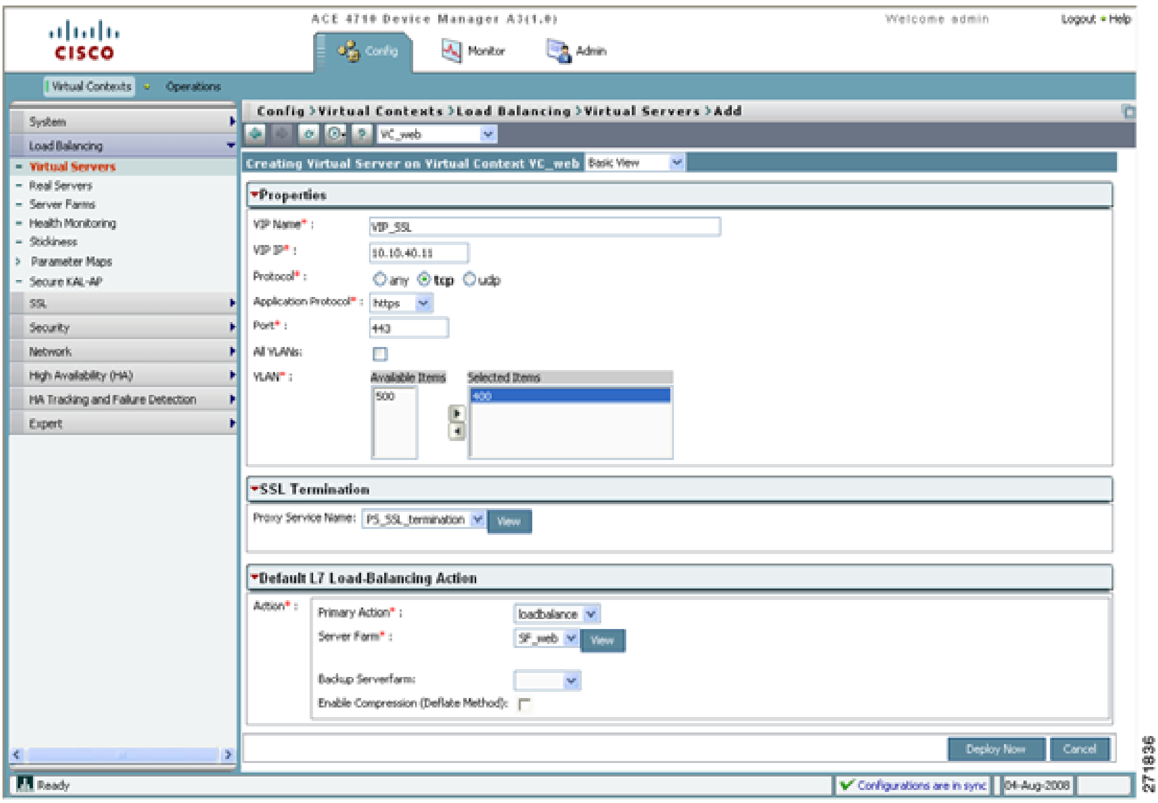
- Proxy Service Name: PS_SSL_termination
- Key List: (choose the key file that you imported earlier)
- Certificate List: (choose the certificate that you imported earlier)
- Parameter Map Name: PM_SSL_termination

Step 16 Click **Deploy Now** to deploy the proxy service on the ACE appliance.

Step 17 Configure a virtual server for SSL termination by choosing **Load Balancing > Virtual Servers**. The Virtual Servers pane appears.

Step 18 Click **Add** to create a virtual server. The Add virtual server window appears (Figure 9-10).

Figure 9-10 Add Virtual Server on Virtual Context Window



Step 19 Enter the following parameters. Leave the remaining parameters blank or with their default values.

- VIP Name: VIP_SSL
- VIP IP: 10.10.40.11
- Protocol: tcp

- Application Protocol: https
- Port: 443
- VLAN: 400
- Proxy Service Name: PS_SSL_termination
- Primary Action: loadbalance
- Server Farm: SF_web

Step 20 Click **Deploy Now** to deploy the virtual SSL server on the ACE appliance.

Configuring the ACE for SSL Termination Using the CLI

You can configure the ACE for SSL termination using the CLI by following these steps:

Step 1 Verify that you are operating in the desired context, by checking the CLI prompt. If necessary, change to the correct context.

```
host1/Admin# changeto VC_web
host1/VC_web#
```

Step 2 Import the key file marketing.pem from an FTP server.

```
host1/VC_web# crypto import ftp 172.25.91.100 Admin /marketing.pem
marketing.pem
Password: ****
Passive mode on.
Hash mark printing on (1024 bytes/hash mark).
#
Successfully imported file from remote server.
host1/VC_web#
```

Step 3 Copy the certificate information from the certificate you received from the CA, and paste it into a certificate file called marketing_cert.pem.

```
host1/VC_web# crypto import terminal marketing_cert.pem
```

```
Enter PEM formatted data ending with a blank line or "quit" on a line
by itself.

-----------BEGIN CERTIFICATE-----------------------------------------
MIIC1DCCAj2gAwIBAgIDCCQAMA0GCSqGSIb3DQEBAgUAMIHEMQswCQYDVQQGEwJa
QTEVMBMGA1UECBMMV2VzdGVybiBDYXBlMRIwEAYDVQQHEwlDYXBlIFRvd24xHTAb
BgNVBAoTFFRoYXd0ZSBDb25zdWx0aW5nIGNjMSgwJgYDVQQLEx9DZXJ0aWZpY2F0
aW9uIFNlcnZpY2VzIERpdmlzaW9uMRkwFwYDVQQDExBUaGF3dGUgU2VydmVyIENB
MSYwJAYJKoZIhvcNAQkBFhdzZXJ2ZXItY2VydHNAdGhhd3RlLmNvbTAeFw0wMTA3
-----------END CERTIFICATE-------------------------------------------
```

Step 4 Enter quit to close the file.

```
quit
host1/VC_web#
```

Step 5 Verify that the certificate matches the key pair.

```
host1/VC_web# crypto verify marketing.pem marketing_cert.pem
keypair in marketing.pem matches certificate in marketing_cert.pem
```

Step 6 Start configuring SSL termination by entering configuration mode.

```
host1/VC_web# config
host1/VC_web(config)#
```

Step 7 Create an SSL proxy service.

```
host1/VC_web(config)# ssl-proxy service PS_SSL_termination
host1/VC_web(config-ssl-proxy)#
```

Step 8 Configure the SSL proxy service by defining the key pair and corresponding certificate.

```
host1/VC_web(config-ssl-proxy)# key marketing
host1/VC_web(config-ssl-proxy)# cert marketing_cert
host1/VC_web(config-ssl-proxy)# exit
host1/VC_web(config)#
```

Step 9 Create a Layer 3 and Layer 4 class map and configure it with the input traffic match criteria.

```
host1/VC_web(config)# class-map CM_SSL
host1/VC_web(config-cmap)# match virtual-address 10.10.40.11 tcp any
host1/VC_web(config-cmap)# exit
host1/VC_web(config)#
```

Step 10 Create a policy map and associate with it the class map CM_SSL.

```
host1/VC_web(config)# policy-map multi-match PM_SSL
host1/VC_web(config-pmap)# class CM_SSL
host1/VC_web(config-pmap-c)#
```

Step 11 Associate the SSL proxy service PS_SSL_termination with the policy map.

```
host1/VC_web(config-pmap-c)# ssl-proxy server PS_SSL_termination
host1/VC_web(config-pmap-c)# exit
host1/VC_web(config-pmap)# exit
host1/VC_web(config)#
```

Step 12 Apply the policy map to the input traffic of the VLAN 400 interface.

```
host1/VC_web(config)# interface vlan 400
host1/VC_web(config-if)# service-policy input PM_SSL
```

Step 13 Display the running configuration to verify that the information that you just added is configured properly.

```
host1/VC_web(config-if)# do show running-config
```

In this chapter, you have configured a virtual server for SSL termination. In the next chapter, you will configure server health monitoring.

CHAPTER 10

Configuring Health Monitoring Using Health Probes

This chapter describes how to configure a health probe on the Cisco 4700 Series Application Control Engine (ACE) appliance. This chapter contains the following sections:

- Overview
- Configuring an HTTP Health Probe Using the Device Manager GUI
- Configuring an HTTP Health Probe Using the CLI

Overview

After reading this chapter, you should have a basic understanding of how the ACE appliance supports server health monitoring using health probes, and how to configure an HTTP health probe.

To detect failures and make reliable load-balancing decisions, you can configure the ACE appliance to track the health of servers and server farms by periodically sending out health probes (sometimes referred to as keepalives). By default, the ACE implicitly checks for server failures.

You can configure probes on the ACE to make active connections and explicitly send traffic to servers. The ACE evaluates the server's response to determine the health of that server.

When the ACE determines the health of a server, the result is one of the following:

- Passed—The server returned a valid response.
- Failed—The server failed to provide a valid response to the ACE within a specified number of retries.

When a server fails in response to the probe, the ACE can check for network problems that prevent a client from accessing that server. The ACE can place the server out of service.

A probe can be any of several types, including TCP, UDP, ICMP, Telnet, and HTTP. You can also configure scripted probes using the TCL scripting language.

You can configure a probe by following these steps:

Step 1 Create the probe and specify its name, type, and attributes.

Step 2 Associate the probe with one of the following:

- A real server.
- A real server that is associated with a server farm. You can associate a single probe or multiple probes to a real server within a server farm.
- A server farm. All real servers in the server farm receive the probe.

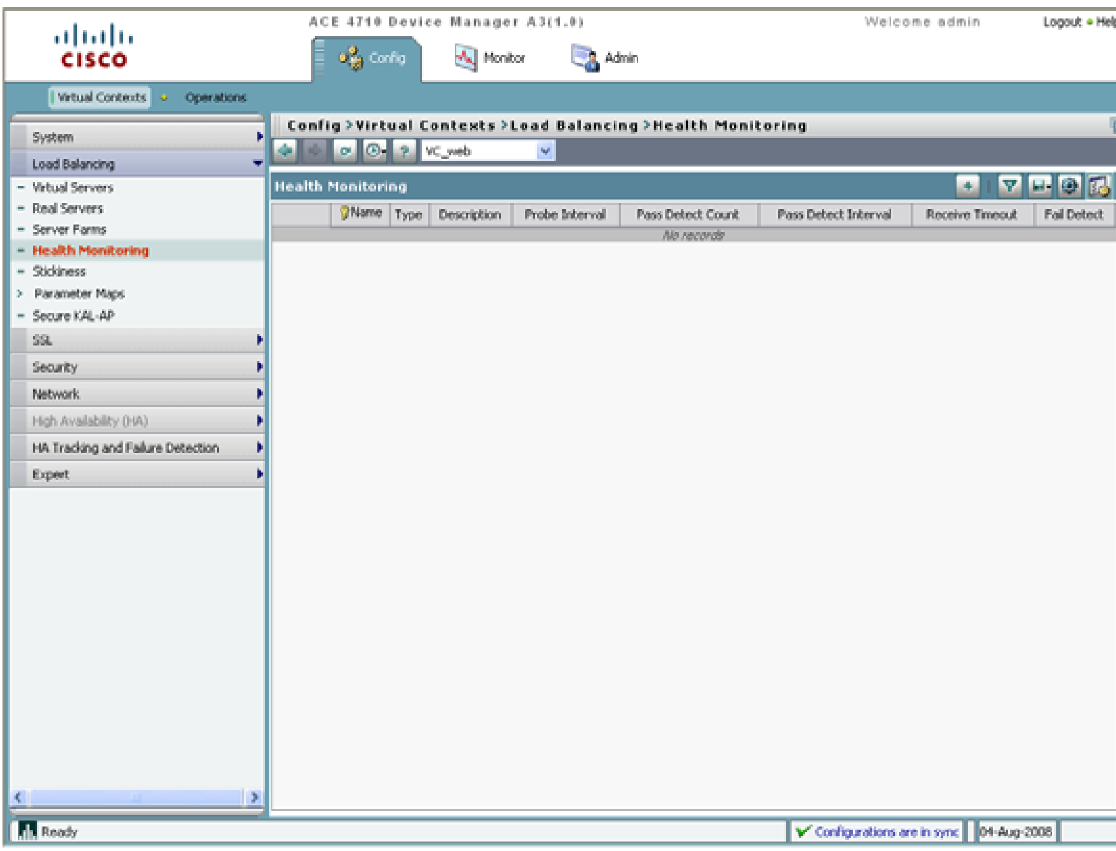
You can configure a probe by using either the ACE Device Manager GUI or the CLI. This chapter describes how to configure an HTTP probe. For information on how to configure other types of probes, see the *Cisco 4700 Series Application Control Engine Appliance Server Load-Balancing Configuration Guide.*

Configuring an HTTP Health Probe Using the Device Manager GUI

You can configure an HTTP health probe using the ACE Device Manager GUI by following these steps:

Step 1 Choose **Load Balancing** > **Health Monitoring**. The Health Monitoring pane appears (Figure 10-1).

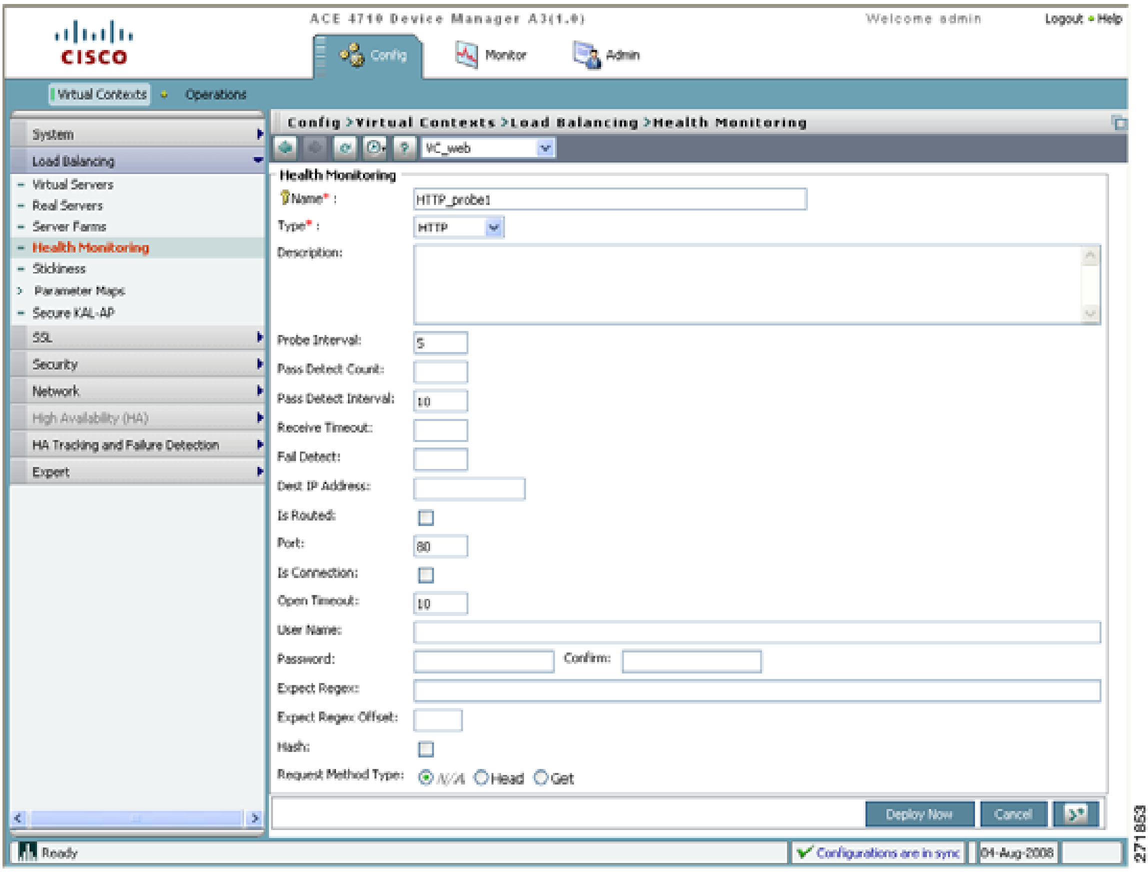
Figure 10-1 Health Monitoring Pane



271854

Step 2 Click **Add** to add a new health probe. The Health Monitoring window appears (Figure 10-2).

Figure 10-2 Health Monitoring Window



Step 3 Enter the following health probe attributes. Leave the remaining attributes blank or with their default values.

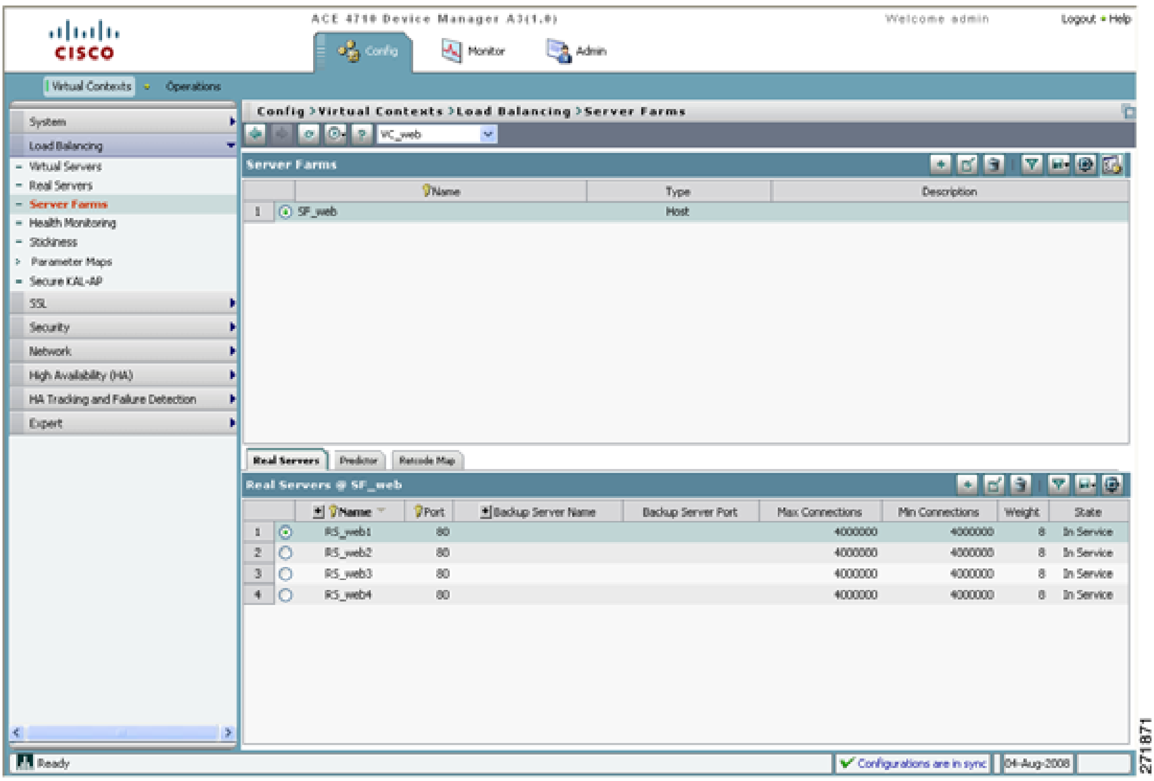
- Name: HTTP_probe1
- Type: HTTP
- Probe Interval: 5

- Pass Detect Interval: 10
- Port: 80

Step 4 Click **Deploy Now** to deploy this configuration on the ACE appliance.

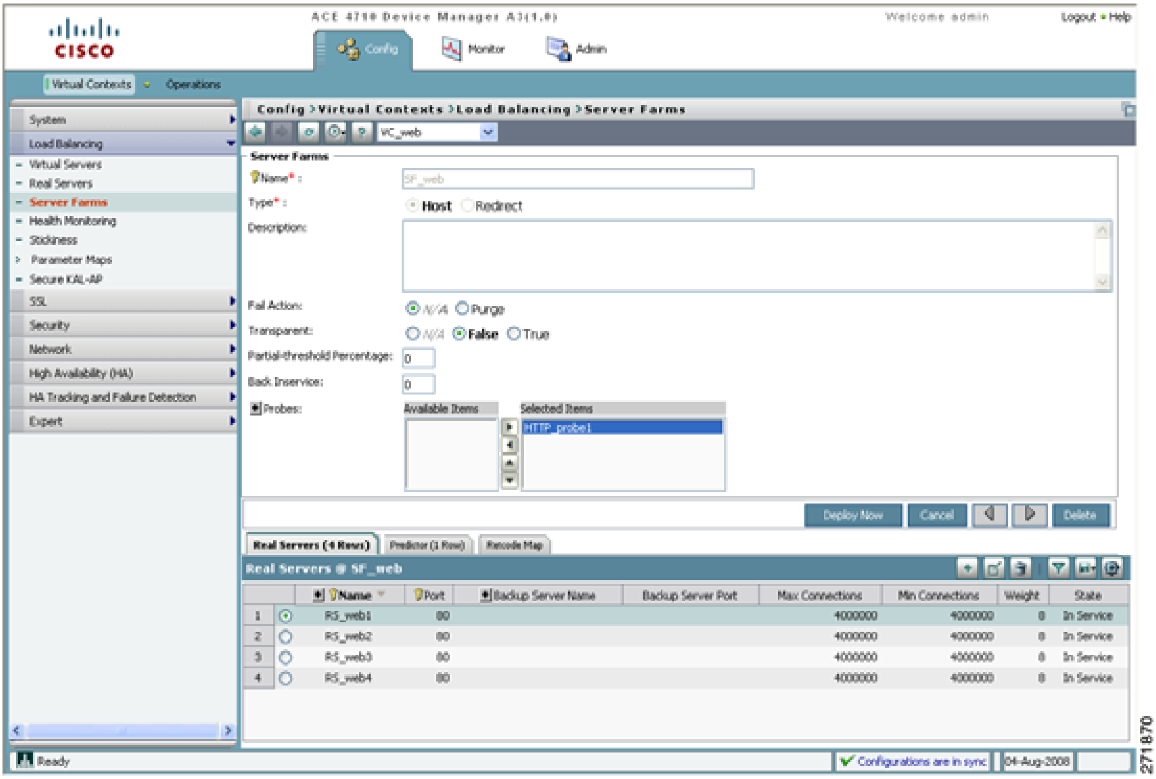
Step 5 Associate the health probe with a server farm by choosing **Load Balancing > Server Farms**. The Server Farms pane appears (Figure 10-3).

Figure 10-3 Server Farms Pane



Step 6 Choose the server farm **SF_web** and click **Edit**. The Server Farms window appears (Figure 10-4).

Figure 10-4 Server Farms Window



Step 7 For Probes, choose **HTTP_probe1** from the Available Items list, and click the **right-arrow** button to move the probe to the Selected Items list.

Step 8 Click **Deploy Now** to associate the health probe HTTP_probe1 with the server farm SF_web.

Configuring an HTTP Health Probe Using the CLI

You can configure an HTTP health probe using the CLI by following these steps:

Step 1 Verify that you are operating in the desired context by checking the CLI prompt. If necessary, change to the correct context.

```
host1/Admin# changeto VC_web
host1/VC_web#
```

Step 2 Enter configuration mode.

```
host1/VC_web# config
host1/VC_web(config)#
```

Step 3 Define an HTTP probe named HTPP_probe1 to access its configuration mode.

```
host1/VC_web(config)# probe http HTTP_probe1
host1/VC_web(config-probe-http)#
```

Step 4 Configure port number 80 for the HTTP probe.

```
host1/VC_web(config-probe-http)# port 80
```

Step 5 Configure a time interval of 5 seconds between probes.

```
host1/VC_web(config-probe-http)# interval 5
```

Step 6 Configure a pass detect interval of 10 seconds, after which the ACE will send another probe to a failed server.

```
host1/VC_web(config-probe-http)# passdetect interval 10
```

Step 7 Exit probe configuration mode.

```
host1/VC_web(config-probe-http)# exit
host1/VC_web(config)#
```

Step 8 Associate the probe HTTP_probe1 with the server farm SF_web, and exit configuration mode.

```
host1/VC_web(config)# serverfarm SF_web
host1/VC_web(config-sfarm-host)# probe HTTP_probe1
host1/VC_web(config-sfarm-host)# exit
host1/VC_web(config)# exit
host1/VC_web#
```

Step 9 Display the HTTP probe configuration.

```
host1/VC_web# show running-config probe
```

In this chapter, you have configured an HTTP health probe.

INDEX

H

L

M

N

P

R

S

T

U

V