

This section describes how to configure a load-balancing predictor on the Cisco 4700 Series Application Control Engine (ACE) appliance.

Guide Contents
Overview
Setting Up an ACE Appliance
Creating a Virtual Context
Configuring Access Control Lists
Configuring Role-Based Access Control
Configuring Server Load Balancing
Configuring a Load-Balancing Predictor (this section)
Configuring Server Persistence Using Stickiness
Configuring SSL Security
Configuring Health Monitoring Using Health Probes

Contents

- [1 Overview](#)
- [2 Configuring a Hash Header Predictor Using the Device Manager GUI](#)
- [3 Configuring a Hash Header Predictor Using the CLI](#)

Overview

After reading this section, 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:

1. Choose a server farm.
2. Choose a predictor type and its parameters.
3. Deploy the configuration.

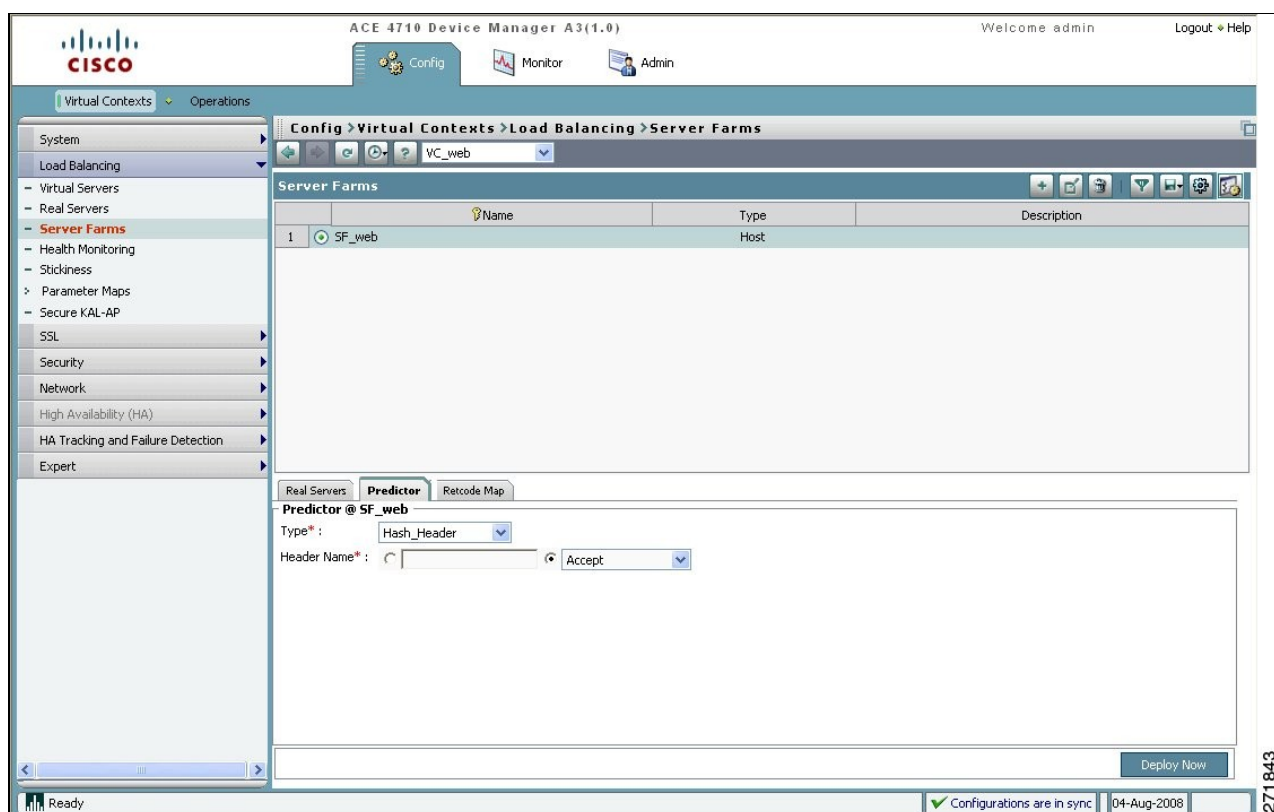
This section describes how to configure a hash header predictor for the server farm that was created in [Configuring Server Load Balancing](#) (as illustrated in Figure 6-1). You can use either the ACE Device Manager GUI or the CLI.

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:

1. Choose **Config > Virtual Contexts**. Choose context **VC_web**.
2. Choose **Load Balancing > Server Farms**. The Server Farms pane appears (Figure 1).

Figure 1 Configuring a Predictor



3. Choose **SF_web**.
4. Choose the **Predictor** tab.
5. Choose **Hash_Header** for the predictor Type.
6. Choose **Accept** for the Header Name.
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:

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#
```

2. Enter configuration mode for SF_web.

```
host1/VC_web# config
```

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

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

3. Configure a hash header predictor.

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

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