



# CHAPTER 1

## Overview

---

This guide provides the following information:

- An overview of the major functions and features of the Cisco Application Control Engine (ACE) module
- 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



### Note

All configuration examples in this guide are based on IPv4. IPv6 is supported on the ACE module in software releases A5(1.0) and later. For information about configuring and using IPv6 with your ACE module, see the *Routing and Bridging Guide, Cisco ACE Application Control Engine*.

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

This chapter contains the following sections:

- [ACE Technologies](#)
- [Setting Up an ACE](#)
- [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](#)
- [Configuring Route Health Injection](#)
- [Configuring Redundancy](#)
- [Configuring Bridged Mode](#)
- [Configuring One-Armed Mode](#)
- [Where to Go Next](#)

# 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, 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

Level	Availability	Scalability	Security
<b>Device</b>	Device Setup	Virtual Contexts	Access Control Lists
	Redundancy	Role-Based Access Control	
<b>Network Services</b>	Virtual Servers	Load Balancing Predictors	SSL
	Health Probes	Server Persistence Using Stickiness	Access Control Lists
		Role-Based Access Control	

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

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

At the network service level, the ACE provides advance services by supporting:

- High services availability—High-performance server load balancing, which allows you to distribute client requests among physical servers and server farms, and provide health monitoring at the server and server farm levels through implicit and explicit health probes.
- Scalability—Virtualization, which allows you to use 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—Allows you to establish and maintain 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

To set up an ACE, you first establish a connection to the ACE and perform the initial device setup to prepare the ACE for providing application networking services. For more information, see [Chapter 2, Setting Up an ACE](#).

## Creating Virtual Contexts

You partition the ACE device into multiple virtual contexts, each with its own resource allocation. For more information, see [Chapter 3, Configuring Virtualization](#).

## Configuring Access Control Lists

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 ACLs at the network services level, or how to configure more granular access control security, see the *Security Guide, Cisco ACE Application Control Engine*.

## 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

Physical Device	Virtual Device	Virtual Server	Server Farm	Real Server
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

You can distribute incoming client requests among the servers in a server farm by defining load-balancing rules called predictors using IP address and port information.

When a client requests 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). This server persistence feature is called 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

You can provide authentication, encryption, and data integrity in a Public Key Infrastructure (PKI) by using the SSL security protocol in your network.

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 module periodically sends messages to the server to determine the 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](#).

## Configuring Route Health Injection

Route Health Injection (RHI) allows you to send traffic to a preferred route by instructing the ACE to advertise a VLAN or a VIP throughout the network. By default, the ACE advertises the VLAN of the VIP interface for RHI. You can also configure the ACE to:

- Advertise the VIP of the virtual server
- Advertise a VLAN for RHI that is different from the VIP VLAN when there is an intervening device (for example, a Cisco Firewall Services Module) between the ACE and the supervisor engine

When the ACE advertises a VIP or a VLAN, the associated route is added to the routing table of the devices on the network. For more information, see [Chapter 11, Configuring Route Health Injection](#).

## Configuring Redundancy

Redundancy, sometimes referred to as high availability (HA) or fault tolerance (FT), uses a maximum of two ACEs in the same Catalyst 6500 series switch or Cisco 7600 series router or in separate switches or routers to ensure that your network remains operational even if one of the modules becomes unresponsive. Redundancy ensures that your network services and applications are always available.

**Note**

---

Redundancy is not supported between an ACE module and an ACE appliance operating as peers. Redundancy must be of the same ACE device type and software release.

---

Redundancy provides seamless switchover of flows in case an ACE becomes unresponsive or a critical host, interface, or a Hot Standby Router Protocol (HSRP) group fails. Redundancy supports the following network applications that require fault tolerance:

- Mission-critical enterprise applications
- Banking and financial services
- E-commerce
- Long-lived flows such as FTP and HTTP file transfers

For more information, see [Chapter 12, Configuring Redundant ACE Modules](#).

## Configuring Bridged Mode

When you configure the ACE in bridge or transparent mode, the ACE bridges the traffic between two VLANs. Each pair of VLANs is a subnet. A router is required to route traffic between these bridged subnets. When deployed in redundant pairs, there is no loop because only one device is forwarding at any given time.

In bridge mode, the ACE is said to be transparent because it acts as a bump in the wire instead of a next hop. The ACE operates at Layer 2 and is not dependent on IP addresses to pass traffic as it is in routed mode.

When operating an ACE in transparent mode, it is desirable to pass Bridge Protocol Data Units (BPDUs) to prevent bridge loops in case a redundant pair becomes active. By default BPDUs do not flow through an ACE. You have to explicitly enable this capability.

For more information, see [Chapter 13, Configuring Bridged Mode](#).

# Configuring One-Armed Mode

You typically configure the ACE with a client-side VLAN and a server-side VLAN. This configuration is referred to as two-armed mode.

You can also configure the ACE so that the clients, the ACE, and the servers are all in the same VLAN. This configuration is referred to as one-armed mode.

For more information, see [Chapter 14, Configuring One-Arm Mode](#).

## Where to Go Next

This chapter has provided you with an overview of the ACE module, its technologies, and its major features. In the next chapter, you will learn how to set up an ACE and to configure it for remote access using a management policy.

