Deployment Guide

Apache Web Server 2.2

TABLE OF CONTENTS

1 INTRODUCTION

Apache HTTP web server has been by far the most popular web server on the Internet today. Apache HTTP web server tops the list of the most used web server applications in the world, surpassing 100 million web sites. Apache HTTP web servers can run on multiple variants of Linux, Unix and Windows platforms.

2 DEPLOYMENT GUIDE OVERVIEW

This deployment guide shows how to install and configure the AX Series with Apache 2.2 HTTP web server. The AX Series Application Delivery Controller (ADC) offers additional security, reliability and optimization; namely: HTTP Compression, RAM Caching, SSL Offload and HTTP Connection Reuse.

3 DEPLOYMENT GUIDE PREREQUISITES

This deployment guide has the following prerequisites:

AX Series Requirement

The A10 Networks AX Series ADC must be running version 2.4.x or higher.

Apache HTTP Web Server Requirements

For Apache HTTP web server requirements, please see
http://httpd.apache.org/docs/2.0/platform/windows.html

Tested environment:

- Apache HTTP web server
  - ♦ Windows 2008 (64-bit) Enterprise Edition Server Operating System (OS)
  - ♦ Apache 2.2 HTTP Server ("Apache" and "httpd")
- Client Access (tested)
  - ♦ Microsoft Internet Explorer Version 8.0
  - ♦ Google Chrome Version 10.0
  - ♦ Mozilla Firefox Version 8

Note: *Generally, if the Virtual IP (VIP) is accessed from an external client, the AX device would be deployed in a routed mode. If the web site services are accessed internally, the AX device would be*

deployed in one-arm mode. If the web server applications are accessed from both internal and external clients, the AX device would be deployed in one-arm mode.

Note: *For additional deployment modes the AX Series device can support, please visit the following URL:*

http://www.a10networks.com/products/axseries-load-balancing101.php

4 ACCESSING THE AX SERIES LOAD BALANCER

This section describes how to access the AX Series device. The AX device can be accessed either from a Command Line Interface (CLI) or Graphical User Interface (GUI):

- CLI – Text-based interface in which you type commands on a command line. You can access the CLI directly through the serial console or over the network using either of the following protocols:
  - Secure protocol – Secure Shell (SSH) version 2
  - Unsecure protocol – Telnet (if enabled)
- GUI – web-based interface in which you click to access configuration or management pages and type or select values to configure or manage the device. You can access the GUI using the following protocol:
  - Secure protocol – Hypertext Transfer Protocol over Secure Socket Layer (HTTPS)

Note: *HTTP requests are redirected to HTTPS by default on the AX device.*

Default Access Information:

- Default Username: “admin”
- Default password: “a10”
- Default IP Address of the device: “172.31.31.31”

(For detailed information on how to access the AX Series device, refer to the *A10 Networks AX Series System Configuration and Administration Guide*.)

5 APACHE HTTP WEB SERVER INSTALLATION PROCEDURES

This deployment guide is based on Windows 2008 Server Apache installation. This deployment guide is not intended to provide full instructions for installing the Apache HTTP web server. If you need Apache installation procedures for Linux or Unix, please refer to the following Unix/Linux Apache Installation Guide: http://httpd.apache.org/docs/2.0/install.html

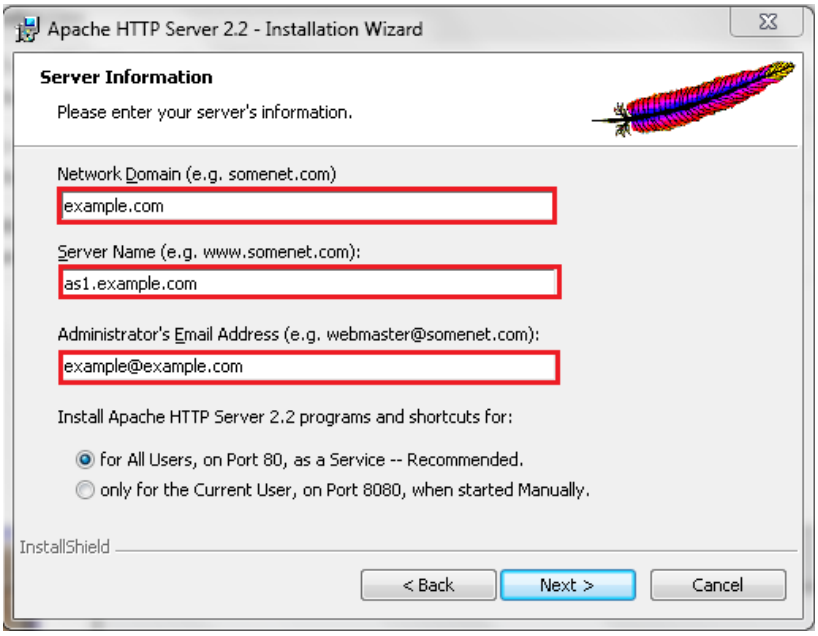
Installation Procedure

To download the latest Apache Server for Windows, download the Microsoft Installer (.msi) package at this site:

http://www.apache.org/dist/httpd/binaries/win32/

Install the Microsoft Installer on the intended web servers. During the installation, you will be prompted to enter the following information:

1. **Network Domain:** example.com
2. **Server Name:** as1.example.com (unique name for every server)
3. **Administrator Email Address:** example@example.com
4. **Select** "for all Users, on Port 80, as a Service--Recommended".



Install Apache software as a typical setup and select all default settings during installation. After the Apache HTTP web server has been installed, the web service process will start automatically. You can start/stop/restart services from your system tray.

To make sure that the Apache HTTP web server is running, open a browser and navigate to http://localhost or http://127.0.0.1. If the Apache server responds back with a page, then your Apache

HTTP web Server is working properly. Finally, set up your "documentroot" location. Documentroot is where your site and HTML files are located.

Note: *Apache also can run on an alternate port, so you need to explicitly include the port number (:8080) in the URL.*

Example: http://localhost:8080

6 APACHE REQUIRED CONFIGURATION

The httpd.conf directory contains the central configuration files for the Apache HTTP web server. The conf file is where the various functions of the HTTP servers are configured, including logging, timeout, keepalive and other configuration items. The httpd file is located within the /etc/httpd/conf/httpd.conf directory.

Source NAT

If you plan to use Source NAT, it is recommended to configure the Apache HTTP web servers with custom logs that include the "X-forwarded-for header". This configuration can be edited directly in the server's httpd.conf.

```
<IfModule mod_log_config.c>
     LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
   combined
```

For Source NAT, other httpd.conf configuration items such as KeepAlive, MaxKeepAliveRequests and KeepAliveTimeout should be configured to their lowest values.

The MaxKeepAliveRequest parameter is the number of requests allowed per persistent connection. The default MaxKeepAliveRequest is 100 and this is recommended to be set to a high value to improve server performance. KeepAliveTimeouts specifies the number of seconds that a server waits before it closes a connection. It is recommended to set the KeepAliveTime to at least 2 seconds, minimum.

```
KeepAlive On
   MaxKeepAliveRequests 100
   KeepAliveTimeout 2
```

7 ARCHITECTURE OVERVIEW

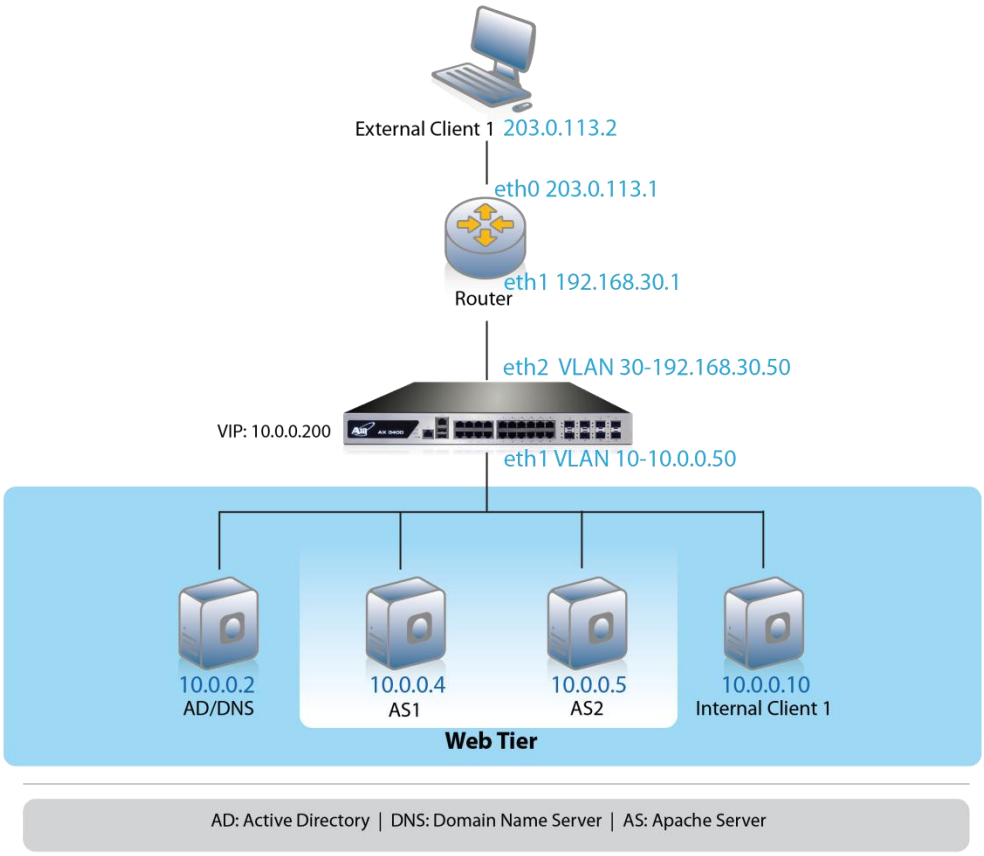


Figure 5: Configuration overview

Note: *In a typical network topology, the Apache HTTP web servers are installed within the demilitarized zone (DMZ) and the AD/DNS server is deployed within the internal network.*

8 BASIC CONFIGURATION

This section explains how the AX Series is configured with Apache 2.2 HTTP web server. This section contains detailed instructions on for installing the real servers, service group, virtual services, and virtual services in a basic Apache HTTP web server.

Ha health monitor is required for the basic configuration to work. If your network topology is based on “one-arm” deployment and internal clients reside on the same subnet as the Virtual Server for the Apache HTTP web server, IP Source Network Address Translation (SNAT) also is required.

Note: *The Virtual Server is also known as the "Virtual IP" ("VIP") that a client accesses during an initial request.*

9 HEALTH MONITOR CONFIGURATION

The AX Series can automatically initiate health status checks for real servers and service ports. Health checks assure that all requests go to functional and available servers. If a server or a port does not respond appropriately to a health check, the server is temporarily removed from the list of available servers. Once the server is restored and starts responding appropriately to the health checks, the server is automatically added back to the list of available servers.

1. Navigate to **Config Mode > SLB > Service**
2. Select **Add** from the **Health Monitor** drop-down list. In the **Name** field, enter "AS HC".
3. Select **Method** "HTTP".
4. Click **OK**, and then see the next section to continue with the Service Group configuration.

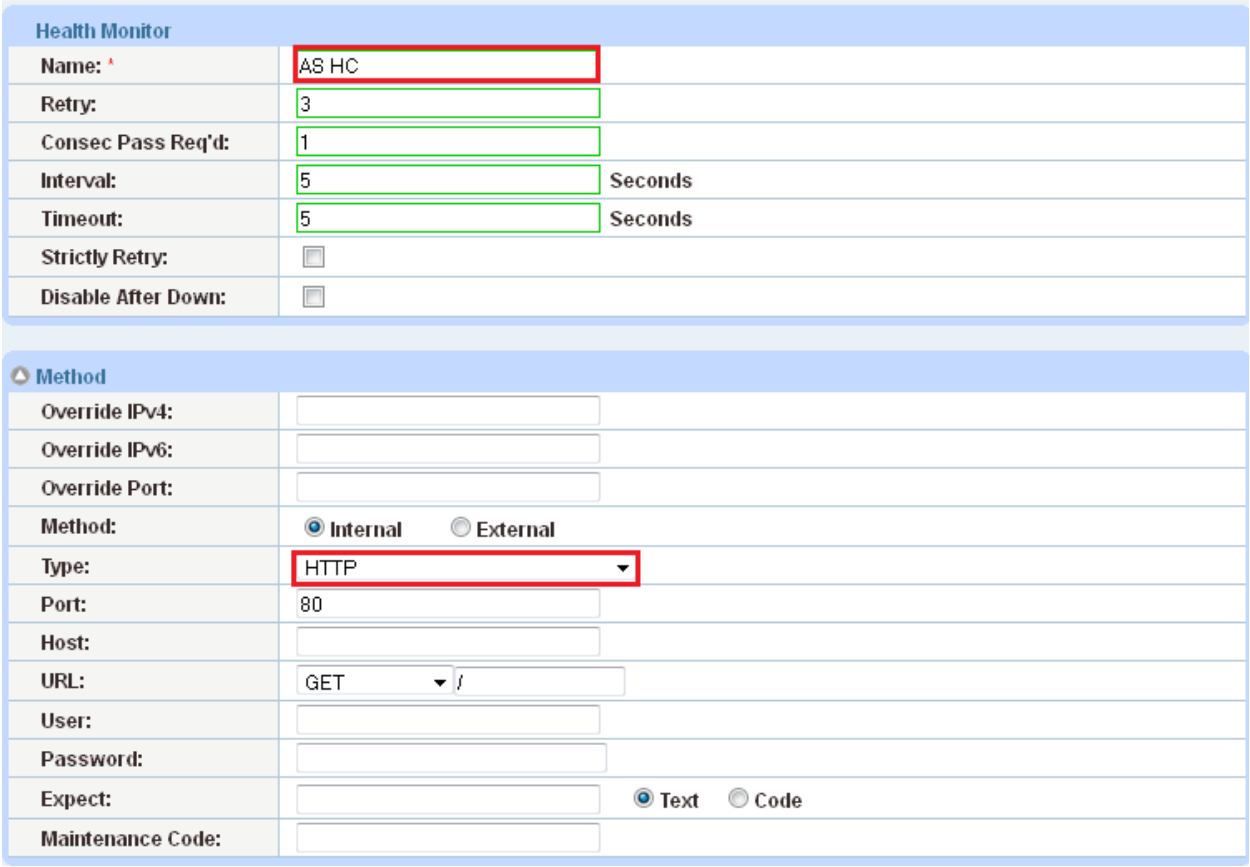


Figure 6: Health monitor configuration

10 SOURCE NAT CONFIGURATION

This section configures the IP Address pool to be used for IP Source Network Address Translation (SNAT). When incoming traffic from a client accesses the VIP address (for example: 10.0.0.200), the client requests are "source NAT-ed", which means that the AX device replaces the client's source IP address with an address from a pool of source NAT addresses. SNAT is required when your network topology is based on "one-arm" deployment and if you have internal clients that reside on the same subnet as the VIP.

To configure Source NAT, use this section to configure the address pool. Then, later in this document, a procedure shows how to apply the pool to the VIP.

1. Navigate to **Config Mode> Service > IP Source NAT > IPv4 Pool**.
2. Click **Add**.
3. Enter the following:
   - **NAT:** "Source NAT"
   - **Start IP Address:** "10.0.0.50"
   - **End IP Address:** "10.0.0.50"
   - **Netmask:** "255.255.255.0"

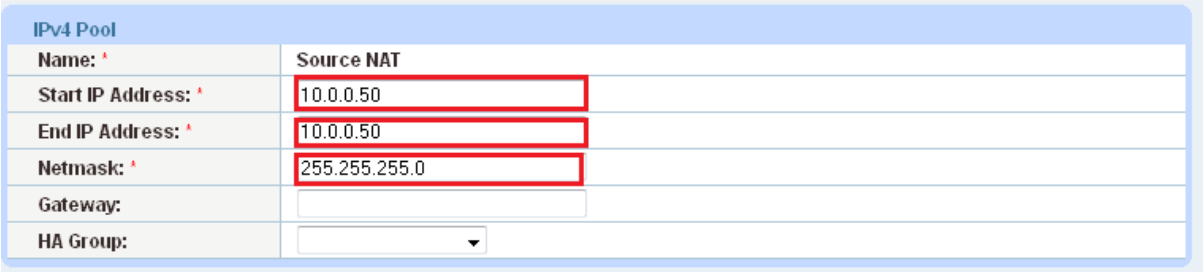


Figure 7: Source NAT pool configuration

4. Click **OK**, then click **Save** to save the configuration.

Note: *When you are in the Virtual Service configuration section, you can apply the Source NAT* pool to the VIP.

Note: *When using the AX device in a High Availability (HA) configuration, an HA Group must be selected. This will prevent duplicate IP addresses from occurring in the Source NAT Pool.*

11 SERVER CONFIGURATION

This section demonstrates how to configure the Apache HTTP web servers on the AX Series.

1. Navigate to **Config Mode > Service > SLB > Server**.
2. Click **Add** to add a new server.
3. Within the Server section, enter the following required information:
   - **Name:** "AS1"
   - **IP address /Host:** "10.0.0.4"

   Note: *Enter additional servers if necessary.*

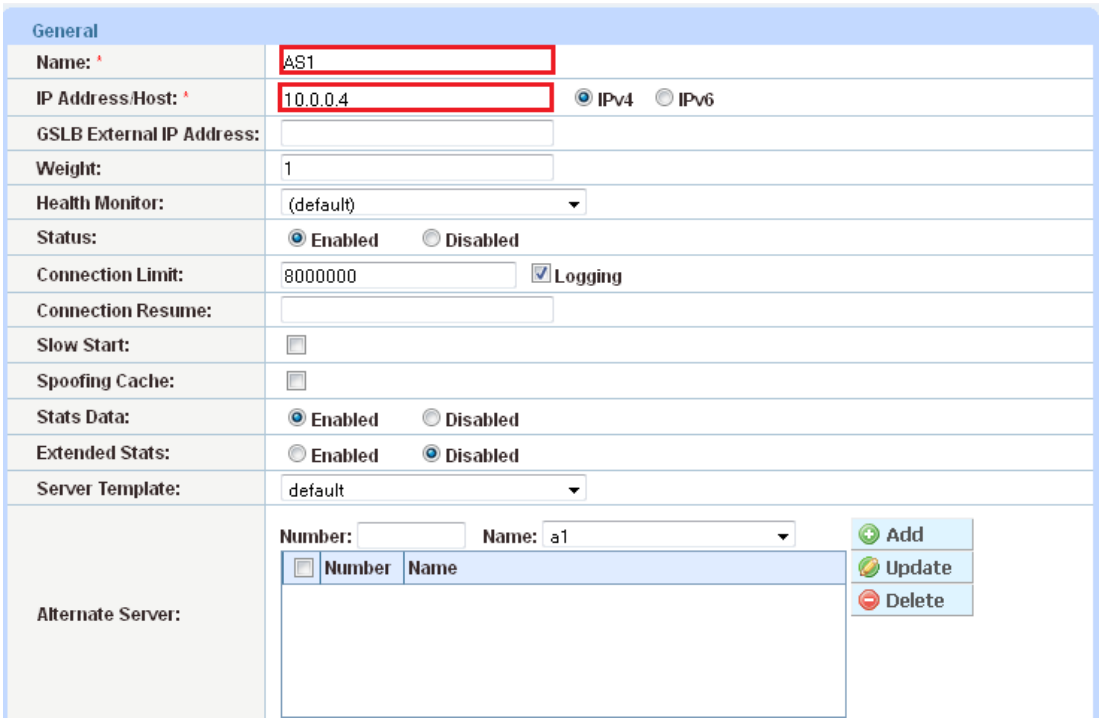


Figure 8: Server configuration

4. To add a port to the server configuration:
   a. Enter the port number in the **Port** field.

b. Select the **Protocol**.

c. Click **Add**.

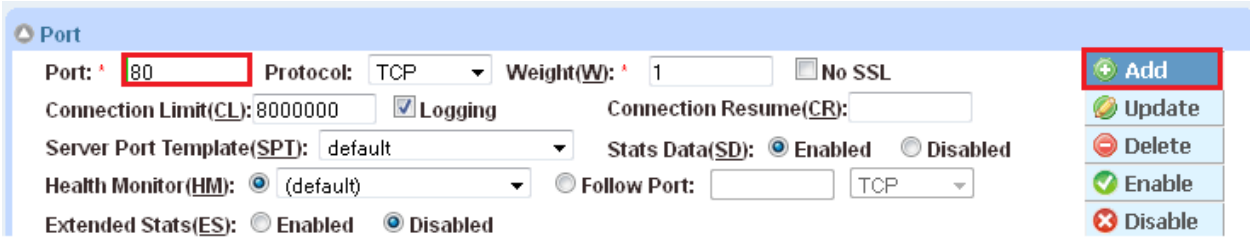


Figure 9: Server port configuration

5. Click **OK**, then click **Save** to save the configuration.

12 SERVICE GROUP CONFIGURATION

This section contains the basic configuration as to how to configure a service group.

1. Navigate to **Config Mode > Service > SLB > Service Group**.
2. Click **Add**.
3. Enter or select the following values:
   - **Name:** "SG80"
   - **Type:** "TCP"
   - **Algorithm:** "Round Robin"
   - **Health Monitor:** "AS HC"
4. In the Server section, select a server from the Server drop-down list and enter "80" in the **Port** field.
5. Click **Add**. Repeat for each server.

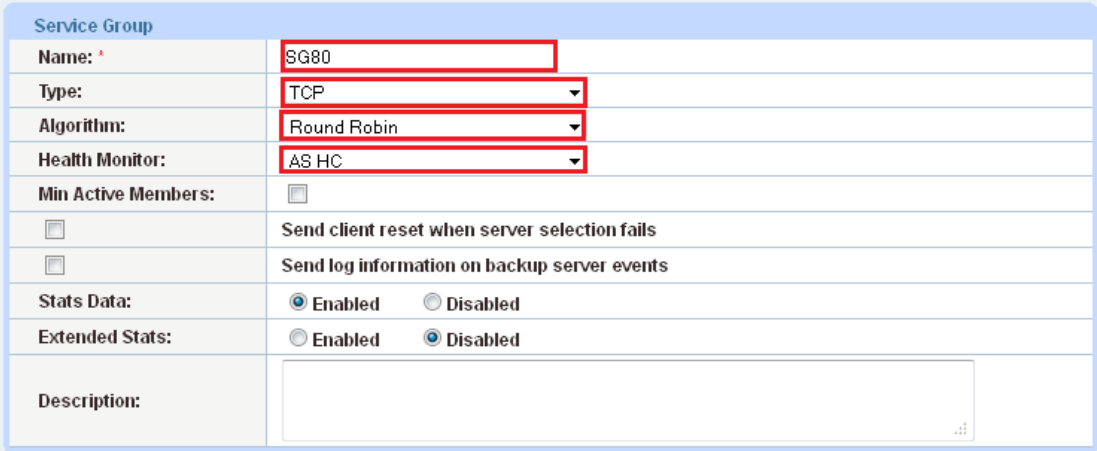


Figure 10: Service group configuration

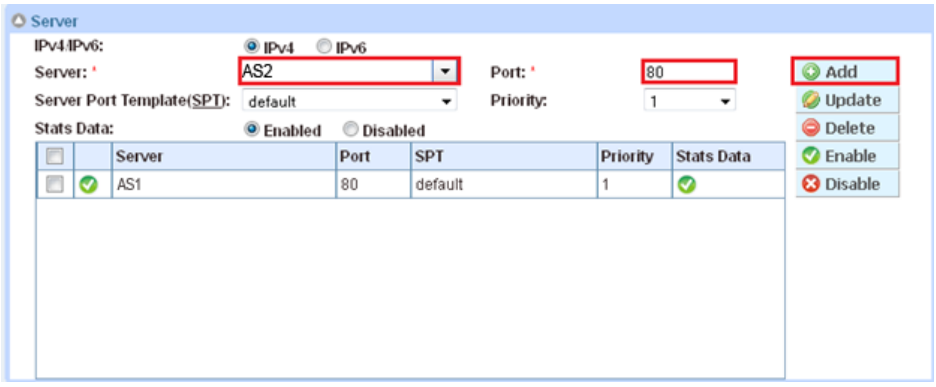


Figure 11: Server configuration

6. Click **OK**, then click **Save** to save the configuration.

13 VIRTUAL SERVER CONFIGURATION

This section contains the basic configuration for a Virtual Server. The Virtual Server is also known as the "Virtual IP" (“VIP”) that a client accesses during an initial request.

1. Navigate to **Config Mode > Service > SLB > Virtual Service**.
2. In the General section, enter the name of the VIP and its IP address:
   - **Name:** “AS VIP”
   - **IP Address:** “10.0.0.200”

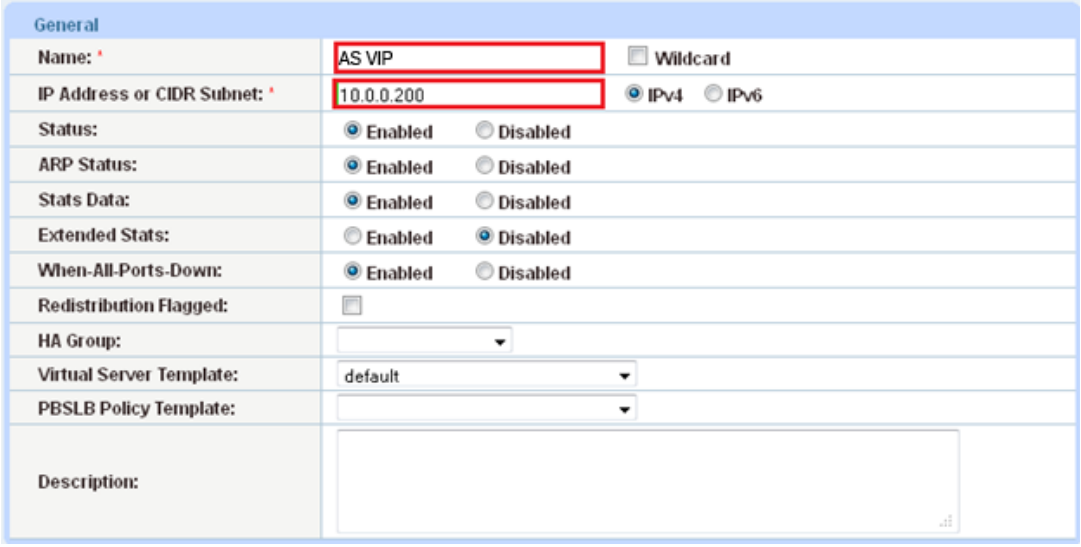


Figure 12: Virtual server configuration

3. In the Port section, click **Add**.

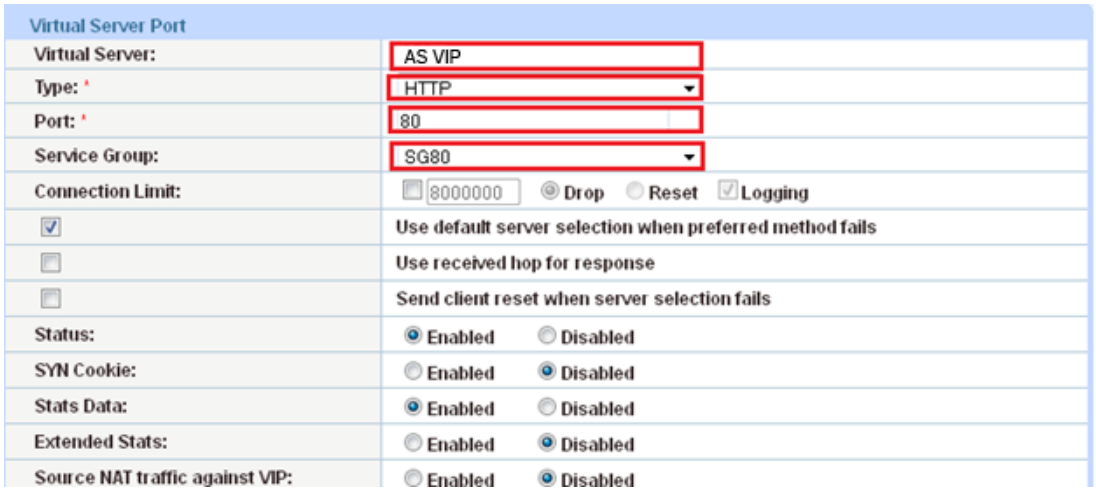


Figure 13: Virtual-server port configuration

4. Select the following values:

   - **Virtual Server:** "HTTP"

     Note: *The Port number will be pre-selected after selecting the protocol type.*

   - **Service Group:** "SG80"

5. Click **OK**, then click **Save** to save the configuration.

13.1 VALIDATING THE CONFIGURATION

This concludes the basic configuration of Apache HTTP web server configuration. Using a client within the network, you can access the VIP with a browser and type the URL as http://10.0.0.200.

Within the AX GUI, you can validate that the Apache HTTP web server application is working and functional by navigating to the configuration lists for servers, virtual servers, and health monitors.

Figure 14: Server list

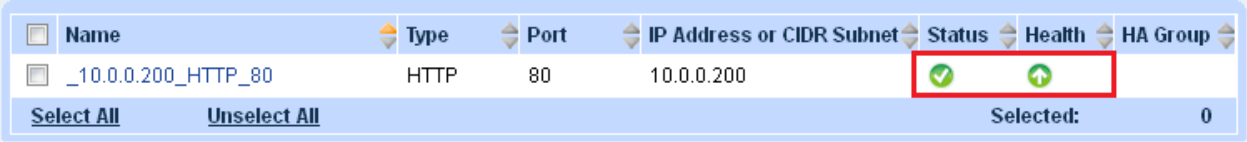


Figure 15: Virtual service list

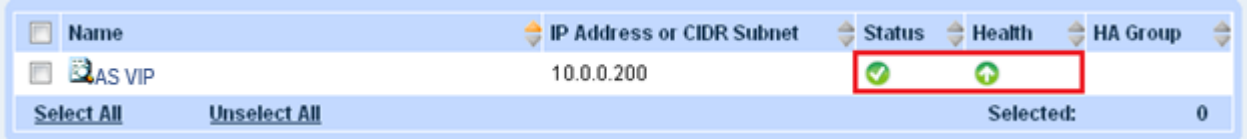


Figure 16: Virtual server list

14 ADVANCED CONFIGURATION

This section contains the advanced configuration of the AX Series with Apache HTTP web server. The advanced configuration increases server performance with features such as SSL Offload, HTTP Compression, HTTP Connection Reuse, Cookie Persistence, and RAM Caching.

The first step in the advanced configuration is to predefine all the optimization and performance features in configuration templates. Once all the performance features are defined in the templates, you can bind the features to the VIP.

Note: *With the assumption that you already understand basic configuration of the server, service group, virtual service and virtual server, this section will move directly to advanced configuration with minimal changes from the basic configuration.*

15 SSL OFFLOAD

SSL Offload mitigates the performance impact that encrypting and decrypting SSL traffic sent via secure SSL can have on a web server application or web server farm. SSL Offload is a performance optimization feature that enables a server to offload the SSL traffic to the AX Series.

To configure AX SSL Offload for the Apache HTTP web server, navigate to the Apache virtual service on the AX device, and change the virtual service type from 80 (HTTP) to 443 (HTTPS).

1. Navigate to **Config Mode > Service > SLB > Virtual Service**.
2. Click on the service name.
3. Select “HTTPS” from the **Port** drop-down list.

Note: *You also may want to change the name to correlate with the virtual port name. (As an example, the "_10.0.0.200_HTTP_80" service should be renamed “_10.0.0.200_HTTPS_443” if the virtual port is updated to use the HTTPS service type.)*

Note: *Leave the port 80 configuration in the service group and server. SSL offload is configured as HTTPS (443) from the front end but is HTTP (80) to the backend servers/server pool.*

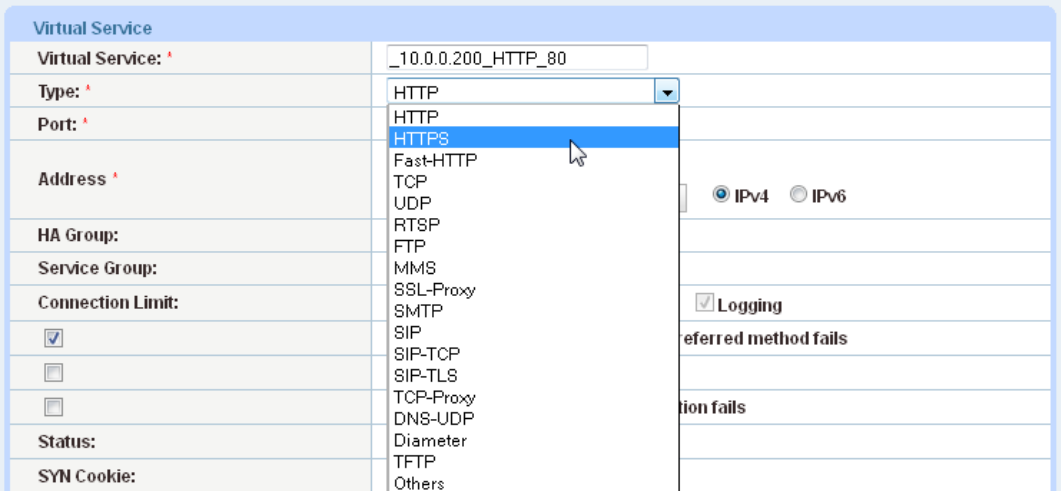


Figure 17: Virtual service configuration

15.1 IMPORT OR GENERATE THE SERVER CERTIFICATE

Since the AX device will act as an HTTPS proxy for the Apache HTTP web servers, the server certificate for each server must be imported onto or generated on the AX device.

There are two options to configure when installing an SSL template from the AX Series:

- **Option 1:** Generate a self-signed certificate on the AX device.
- **Option 2:** Import an SSL certificate and key signed by a Certificate Authority (CA).

15.1.1 OPTION 1: GENERATE A SELF-SIGNED CERTIFICATE

1. Navigate to **Config Mode > Service > SSL Management > Certificate**.
2. Click **Create**.
3. Enter the **File Name** of the certificate, “WS”.
4. From the **Issuer** drop-down list, select “Self”.
5. Enter the following values:
    - **Common Name:** “AS”
    - **Division:** “A10”
    - **Organization:** “A10”
    - **Locality:** San Jose
    - **State or Province:** “CA”
    - **Country:** “USA”
    - **Email Address:** “ASadmin@example.com”
    - **Valid Days:** “730” (Default)
    - **Key Size (Bits):** “2048”

Note: *The AX Series can support 512-bit, 1028-bit, 2048-bit, and 4096-bit keys. The higher the bit size, the more CPU processing that will be required on the AX device.*

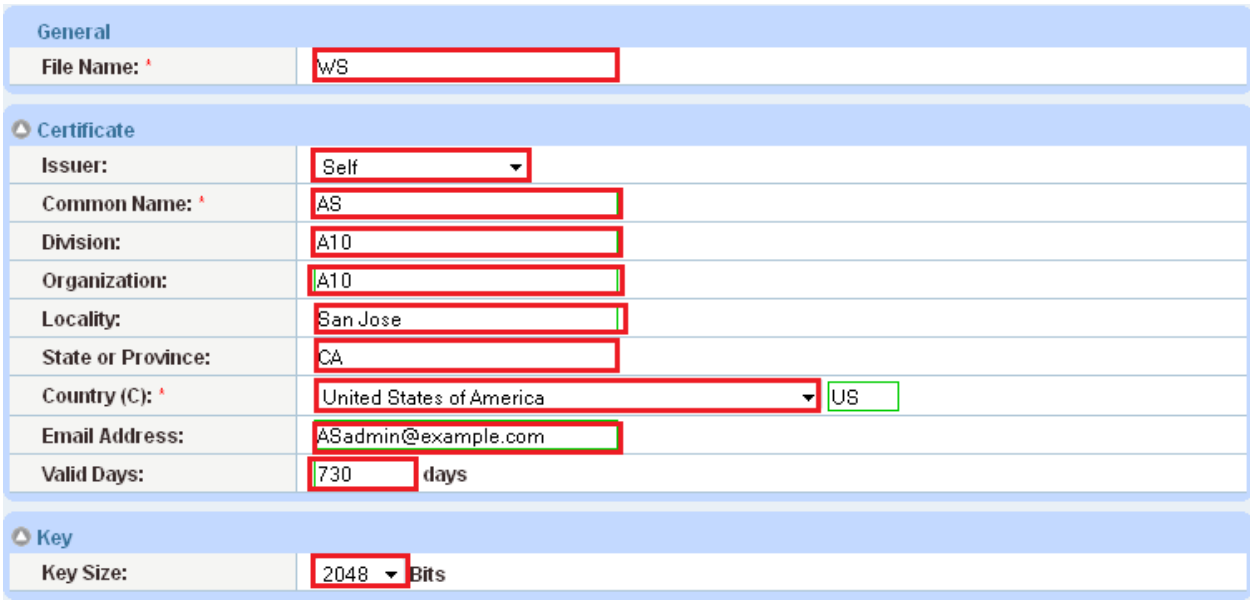


Figure 18: Self-signed certificate configuration

6. Click **OK**, then click **Save** to save the configuration.

15.1.2 OPTION 2: IMPORT THE CERTIFICATE AND KEY

1. Navigate to **Config Mode > Service > SSL Management > Certificate**.
2. Click **Import**.
3. Enter the **Name**, “AS”.
4. Select “Local” or “Remote”, depending on the file location.
5. Enter the certificate **Password** (if applicable).
6. Enter or select file location and access settings.
7. Click **OK**.

Note: *If you are importing a CA-signed certificate for which you used the AX device to generate the CSR, you do not need to import the key. The key is automatically generated on the AX device when you generate the CSR.*

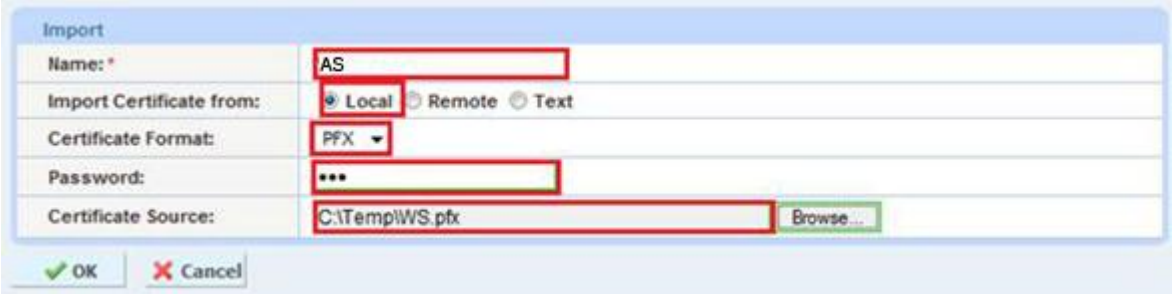


Figure 19: SSL certificate import

8. Click **OK**, then click **Save** to save the configuration.

16 CONFIGURE AND APPLY CLIENT SSL TEMPLATE

This section describes how to configure a client SSL template and apply it to the VIP.

1. Navigate to **Config Mode > Service > Template > SSL > Client SSL**.
2. Click **Add**.
3. Enter or select the following values:
   - **Name:** “Client SSL-AS”
   - **Certificate Name:** "AS”
   - **Key Name:** “AS”
   - **Pass Phrase:** “example”
   - **Confirm Pass Phrase:** “example”

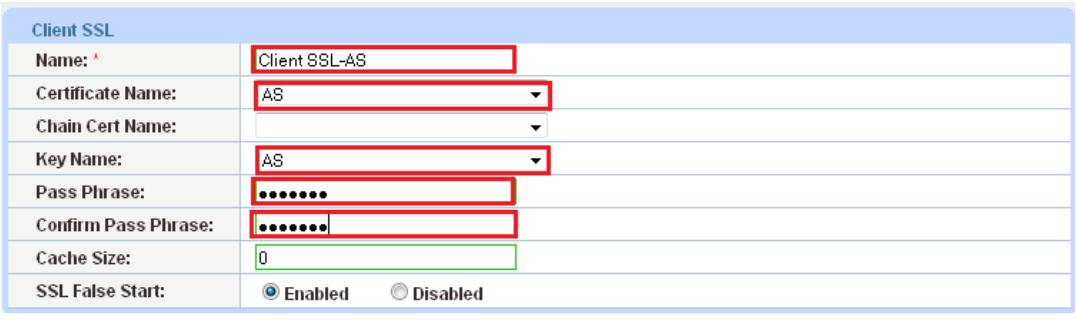


Figure 20: Client SSL template

Once the Client SSL template is completed, you must bind the template to the HTTPS VIP (port 443), as follows:

1. Navigate to **Config Mode > Service > SLB > Virtual Server**.
2. Click on the virtual server name.
3. Select “443” and click **Edit**.
4. Apply the Client SSL template created by selecting it from the **Client-SSL Template** drop-down list.

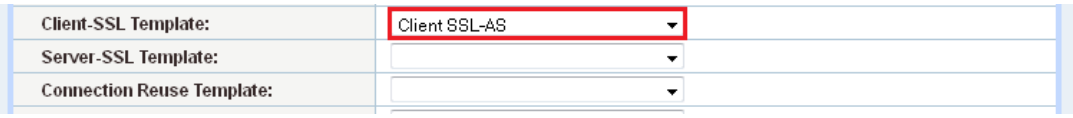


Figure 21: Client SSL template selection

5. Click **OK**, then click **Save** to save the configuration.

17 HTTP COMPRESSION

HTTP Compression is a bandwidth optimization feature that compresses the HTTP objects requested from a web server. If your web site uses lots of bandwidth, enabling HTTP Compression will provide faster transmission times between a client's browser and web servers. The purpose of compression is to transmit the requested data more efficiently and with faster response times to the client. HTTP Compression makes HTTP requests much faster by transmitting less data.

17.1 CREATE HTTP COMPRESSION TEMPLATE

1. Navigate to **Config Mode > Service > Template > Application > HTTP**.
2. Click **Add**.
3. Enter a **Name**, “HTTP Compression”.
4. Click **Compression** to display the compression configuration options.

Note: *Compression is disabled by default. When compression is enabled, the compression options will have the default values shown in following example*:

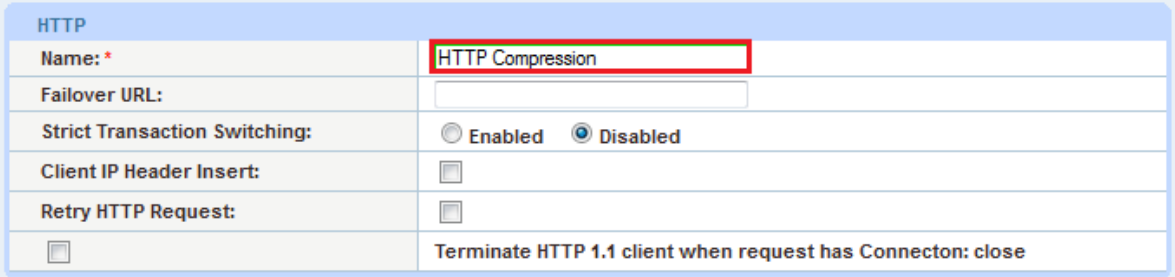


Figure 22: HTTP Compression template

5. Select **Enabled** next to **Compression**.

Note: *The AX* Series *offers various compression levels, ranging from levels 1 to 9. Level 1 is the recommended compression setting.*

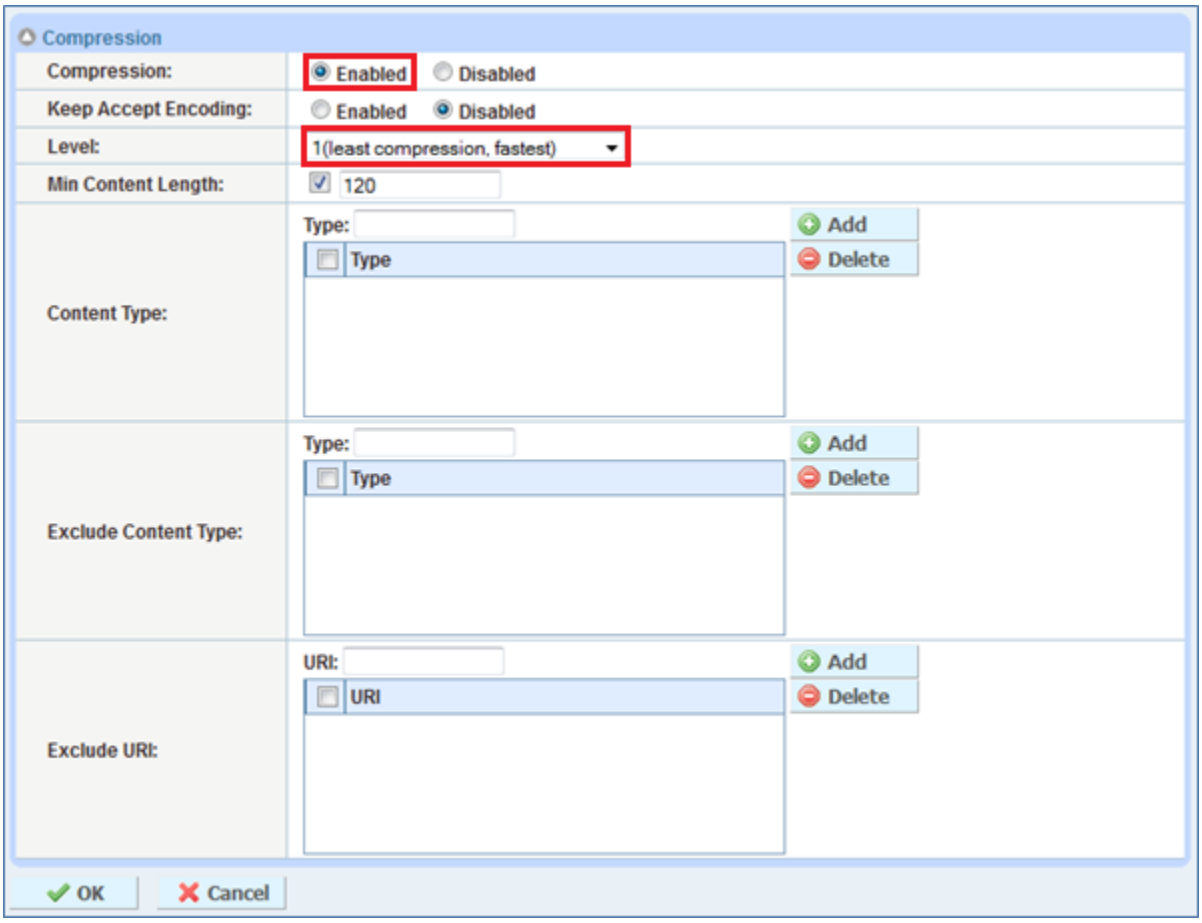


Figure 23: Compression configuration column

6. Click **OK**, then click **Save** to save the configuration.

18 COOKIE PERSISTENCE

To enable cookie persistence, the template must be created first, as follows:

1. Navigate to **Config Mode > Service > Template > Cookie Persistence**.
2. Click **Add** to add a new cookie persistence template.
3. Enter the **Name**, "AS".
4. Select the **Expiration** radio button and enter “86400” in the **Seconds** field.

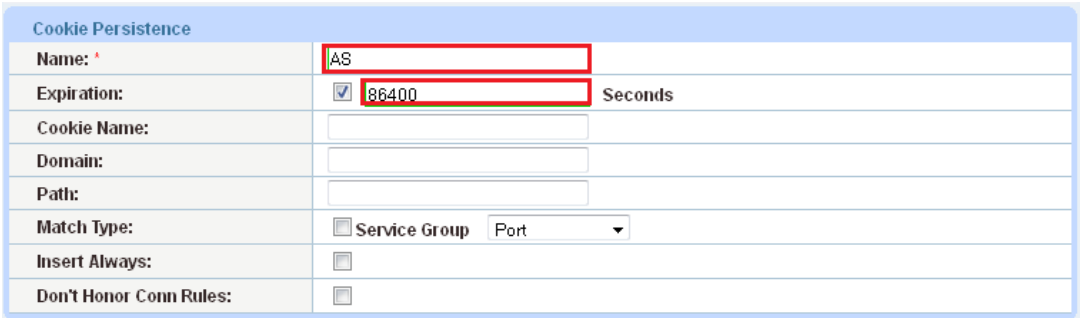


Figure 24: Cookie Persistence template

5. Click **OK**, then click **Save** to save the configuration.

19 TCP CONNECTION REUSE

1. Navigate to **Config Mode > Service > Template > Connection Reuse**.
2. Click **Add**.
3. Enter **Name**: “ASConnectionReuse”.

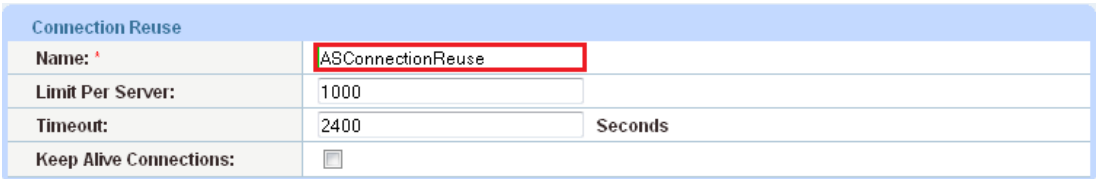


Figure 25: TCP Connection Reuse template

4. Click **OK**, then click **Save** to save the configuration.

Note: *For the best connection reuse results, these are the recommend Apache HTTP web server settings in the Apache httpd.conf file.*

- KeepAlive – On
- MaxKeepAliveRequests – 0 or a high number such as 800+. The value 0 = unlimited.
- KeepAlive Timeout – high value, 250+
- MaxRequestsPerChild – 5000-10000

20 RAM CACHING

RAM Caching allows cacheable data to be cached within the AX Series device itself, thus reducing overhead on the Apache HTTP web servers and increasing their capacity. RAM Caching reduces the number of connections and server requests that need to be processed.

1. Navigate to **Config Mode > Service > Template > Application > RAM Caching**.
2. Click **Add**.
3. Enter or select the following values:

- **Name:** "ASRC"
- **Age:** 3600 seconds
- **Max Cache Size:** 80 MB
- **Min Content Size:** 512 Bytes
- **Max Content Size:** 81920 Bytes
- **Replacement Policy**: "Least Frequently Used"

Note: *The RAM Caching policy option is not required unless you have specific data that requires caching, no caching, or invalidation. These policy options can be configured in the* Policy *section of the RAM Caching template. For additional information on RAM caching policies, please refer to the AX Series Application Delivery and Server Load Balancing Guide.*

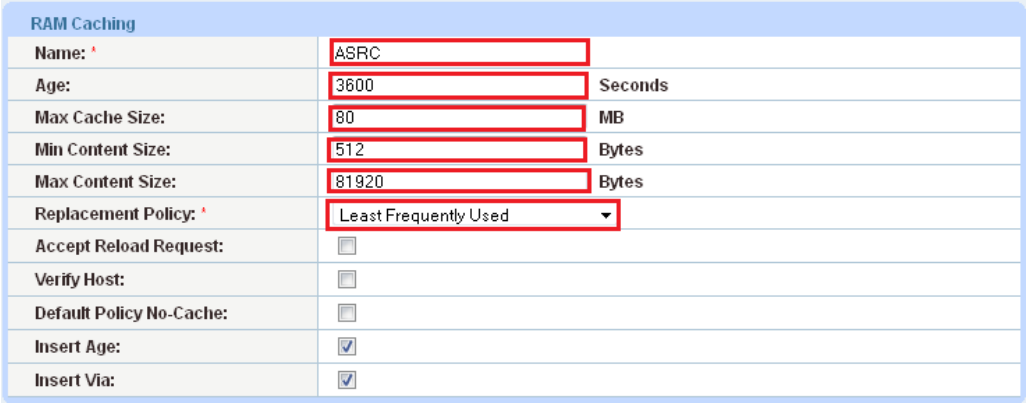


Figure 26: RAM Caching template

4. Click **OK**, then click **Save** to save the configuration.

21 HTTP-TO-HTTPS REDIRECT

This section explains how to redirect Apache web traffic that originates from HTTP to HTTPS using AX aFleX scripts. aFleX is based on a standard scripting language, TCL, and enables the AX device to perform Layer 7 deep-packet inspection (DPI). For examples of aFleX scripts, please refer to the following URL:

http://www.a10networks.com/products/axseries-aflex_advanced_scripting.php

As an example, one of the most commonly used aFleX scripts is the “HTTP redirect to HTTPS traffic” script. You can download additional aFleX script examples from the URL listed above.

To configure a transparent HTTPS redirect using aFleX:

1. Create the aFleX script.
2. Configure a VIP with virtual service HTTP (port 80).
3. Apply the aFleX script to the virtual port on the VIP.

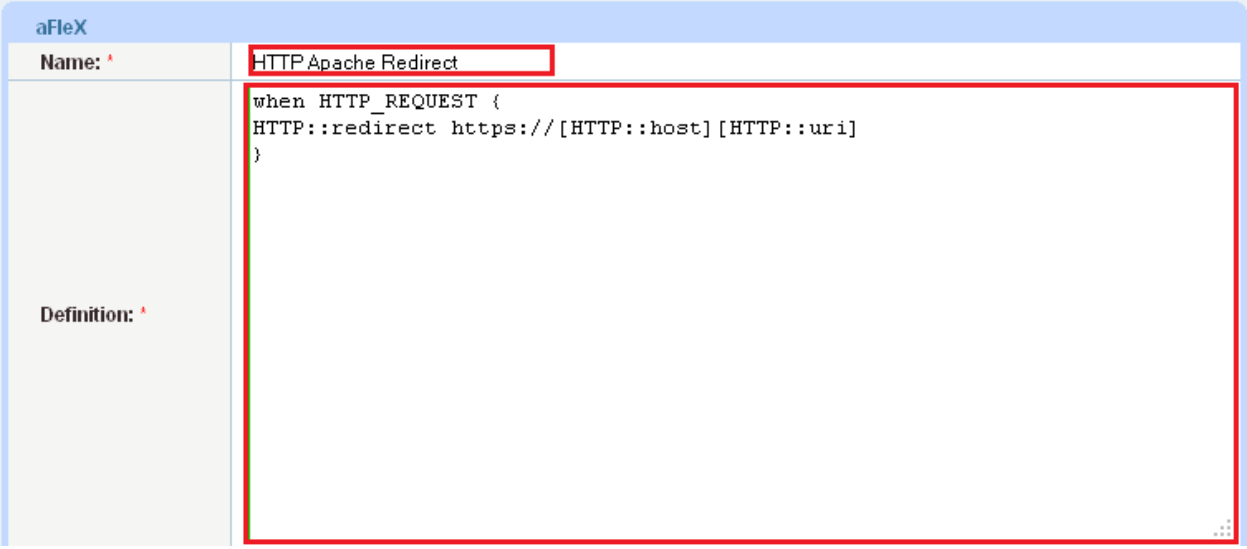


Figure 27: Redirect script

Redirect Script Copy and Paste:

```
when HTTP_REQUEST {

HTTP::redirect https://[HTTP::host][HTTP::uri]

}
```

Note: *The aFleX script must be bound to virtual-server port 80.*

22 APPLY OPTIMIZATION AND ACCELERATION FEATURE TEMPLATES ON VIP

After configuring the optimization and acceleration features, you must bind them to the virtual port on the VIP to place them into effect.

1. Navigate to **Config Mode > Service > SLB > Virtual Service**.
2. Click on the virtual service name.

3. Apply the features by selecting the templates from the applicable drop-down lists.

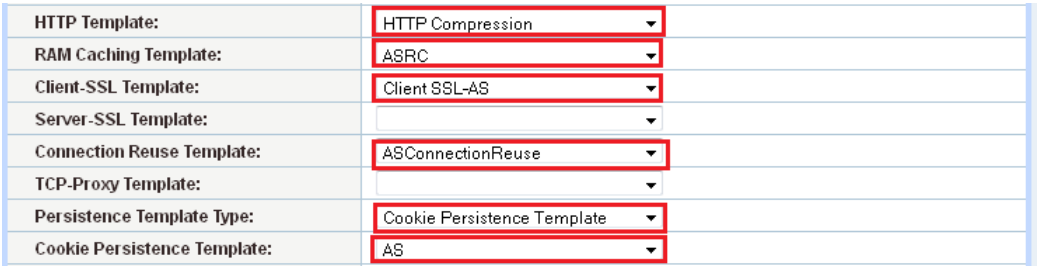


Figure 28: Applying features

4. Click **OK**, then click **Save** to save the configuration.

23 SUMMARY AND CONCLUSION

The sections above show how to deploy the AX device for optimization of Apache HTTP web servers. By using the AX device to load balance a pool of Apache HTTP web servers, the following key advantages are achieved:

- High availability for Apache HTTP web servers to prevent web site failure, with no adverse impact on user access to applications
- Seamless distribution of client traffic across multiple Apache HTTP web servers for site scalability
- Higher connection counts, faster end user responsiveness, and reduced Apache HTTP web server CPU utilization by initiating SSL Offload, HTTP Compression, RAM Caching and Connection Reuse
- Improved site performance and reliability to end users

By using the AX Series Advanced Traffic Manager, significant benefits are achieved for all Apache HTTP web application users. For more information about AX Series products, please refer to the following URLs:

http://www.a10networks.com/products/axseries.php

http://www.a10networks.com/resources/solutionsheets.php

http:/www.a10networks.com/resources/casestudies.php

A. CLI COMMANDS FOR SAMPLE BASIC CONFIGURATION

The following sections show the CLI commands for implementing the sample configurations described above.

```
ASAX#show running-config
hostname ASAX
clock timezone Europe/Dublin
slb server AS1 10.0.0.4
   port 80  tcp
slb server AS2 10.0.0.5
   port 80  tcp
slb service-group SG80 tcp
    member AS1:80
    member AS2:80
slb virtual-server "AS VIP" 10.0.0.200
   port 80  http
      name _10.0.0.200_HTTP_80
      source-nat pool "Source NAT"
      service-group SG80
end
ASAX#
```

B. CLI COMMANDS FOR SAMPLE ADVANCED CONFIGURATION

```
ASAX#show running-config
hostname ASAX
ip nat pool "Source NAT" 10.0.0.50 10.0.0.50 netmask /24
health monitor "AS HC"
```

```
 method http
slb server AS1 10.0.0.4
   port 80  tcp
slb server AS2 10.0.0.5
   port 80  tcp
slb service-group SG80 tcp
    member AS1:80
    member AS2:80
slb template connection-reuse ASConnectionReuse
slb template cache ASRC
slb template http "HTTP Compression"
slb template client-ssl "Client SSL-AS"
   cert AS
   key AS pass-phrase encrypted
KZlpUbp6Q888EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
slb template persist cookie AS
   expire 86400
   insert-always
slb template persist source-ip srcip
slb virtual-server "AS VIP" 10.0.0.200
   port 443  https
      name _10.0.0.200_HTTPS_443
      source-nat pool "Source NAT"
      service-group SG80
      template http "HTTP Compression"
      template cache ASRC
      template client-ssl "Client SSL-AS"
```

```
      template http
      template connection-reuse ASConnectionReuse
      template persist cookie AS
end
ASAX#
```