



# Apache HTTP

## Deployment Guide

UPDATED: 27 July 2023

**© 2022 Progress Software Corporation and/or one of its subsidiaries or affiliates. All rights reserved.**

These materials and all Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. The information in these materials is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear therein. The references in these materials to specific platforms supported are subject to change.

#1 Load Balancer in Price/Performance, 360 Central, 360 Vision, Chef, Chef (and design), Chef Habitat, Chef Infra, Code Can (and design), Compliance at Velocity, Corticon, Corticon.js, DataDirect (and design), DataDirect Cloud, DataDirect Connect, DataDirect Connect64, DataDirect XML Converters, DataDirect XQuery, DataRPM, Defrag This, Deliver More Than Expected, DevReach (and design), Driving Network Visibility, Flowmon, Inspec, Ipswitch, iMacros, K (stylized), Kemp, Kemp (and design), Kendo UI, Kinvey, LoadMaster, MessageWay, MOVEit, NativeChat, OpenEdge, Powered by Chef, Powered by Progress, Progress, Progress Software Developers Network, SequeLink, Sitefinity (and Design), Sitefinity, Sitefinity (and design), Sitefinity Insight, SpeedScript, Stylized Design (Arrow/3D Box logo), Stylized Design (C Chef logo), Stylized Design of Samurai, TeamPulse, Telerik, Telerik (and design), Test Studio, WebSpeed, WhatsConfigured, WhatsConnected, WhatsUp, and WS\_FTP are registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and/or other countries.

Analytics360, AppServer, BusinessEdge, Chef Automate, Chef Compliance, Chef Desktop, Chef Workstation, Corticon Rules, Data Access, DataDirect Autonomous REST Connector, DataDirect Spy, DevCraft, Fiddler, Fiddler Classic, Fiddler Everywhere, Fiddler Jam, FiddlerCap, FiddlerCore, FiddlerScript, Hybrid Data Pipeline, iMail, InstaRelinker, JustAssembly, JustDecompile, JustMock, KendoReact, OpenAccess, PASOE, Pro2, ProDataSet, Progress Results, Progress Software, ProVision, PSE Pro, Push Jobs, SafeSpaceVR, Sitefinity Cloud, Sitefinity CMS, Sitefinity Digital Experience Cloud, Sitefinity Feather, Sitefinity Thunder, SmartBrowser, SmartComponent, SmartDataBrowser, SmartDataObjects, SmartDataView, SmartDialog, SmartFolder, SmartFrame, SmartObjects, SmartPanel, SmartQuery, SmartViewer, SmartWindow, Supermarket, SupportLink, Unite UX, and WebClient are trademarks or service marks of Progress Software Corporation and/or its subsidiaries or affiliates in the U.S. and other countries. Java is a registered trademark of Oracle and/or its affiliates. Any other marks contained herein may be trademarks of their respective owners.

Please refer to the NOTICE.txt or Release Notes – Third-Party Acknowledgements file applicable to a particular Progress product/hosted service offering release for any related required third-party acknowledgements.

# Table of Contents

---

<b>1 Introduction</b>	<b>4</b>
1.1 Document Purpose	4
1.2 Intended Audience	4
<b>2 Apache HTTP Template</b>	<b>5</b>
<b>3 Architecture</b>	<b>6</b>
<b>4 Configure the LoadMaster</b>	<b>7</b>
4.1 Enable Subnet Originating Requests Globally	7
4.2 Enable Check Persist Globally	8
4.3 Create the Apache HTTP Virtual Services	9
4.3.1 Create an Apache HTTP Virtual Service	9
4.3.2 Create an Apache HTTPS Virtual Service	10
4.3.3 Create an Apache HTTPS Offloaded Virtual Service	11
4.3.4 Create an Apache HTTPS Re-encrypt Virtual Service	13
<b>References</b>	<b>16</b>
<b>Last Updated Date</b>	<b>17</b>

# 1 Introduction

The Apache HTTP Server is the world's most used web server software. Apache development began in 1995 and played a key role in the initial growth of the World Wide Web and quickly became the dominant HTTP server. It has remained most popular since 1996 and in 2009 it became the first web server software to serve more than 100 million websites.

The Kemp LoadMaster is used to load balance the Apache HTTP workload. The LoadMaster offers advanced Layer 4 and Layer 7 server load balancing, SSL Acceleration and a multitude of other advanced Application Delivery Controller (ADC) features. The LoadMaster intelligently and efficiently distributes user traffic among the application servers so that users get the best experience possible.

## 1.1 Document Purpose

This document provides the recommended LoadMaster settings used when load balancing the Apache HTTP workload. The Kemp Support Team is available to provide solutions for scenarios not explicitly defined. The Kemp support site can be found at: <https://support.kemptechnologies.com>

## 1.2 Intended Audience

This document is intended to be read by anyone who is interested in configuring the LoadMaster to optimize the Apache HTTP server.

# 2 Apache HTTP Template

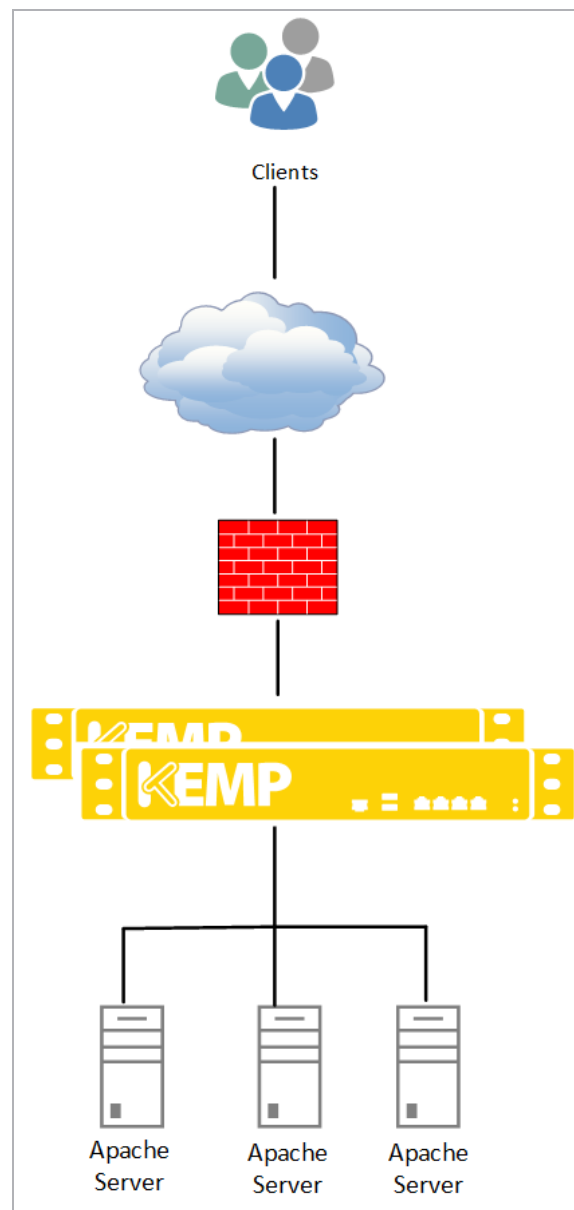
Kemp has developed a template containing our recommended settings for Apache HTTP. You can install this template on the LoadMaster and use it when creating Virtual Services. Using a template automatically populates the settings in the Virtual Services, which is quicker and easier than manually configuring each Virtual Service. If needed, you can make changes to any of the Virtual Service settings after using the template.

Download released templates from the following page: [LoadMaster Templates](#).

For more information and steps on how to import and use templates, refer to the [Virtual Services and Templates, Feature Description](#).

For steps on how to manually add and configure each of the Virtual Services, refer to the **Configure the LoadMaster** section of this document.

# 3 Architecture



# 4 Configure the LoadMaster

The deployed Apache HTTP environment determines which of the following setups is used.

## 4.1 Enable Subnet Originating Requests Globally

It is best practice to enable the **Subnet Originating Requests** option globally.

In a one-armed setup (where the Virtual Service and Real Servers are on the same network/subnet) **Subnet Originating Requests** is usually not needed. However, enabling **Subnet Originating Requests** should not affect the routing in a one-armed setup.

In a two-armed setup where the Virtual Service is on network/subnet A, for example, and the Real Servers are on network B, **Subnet Originating Requests** should be enabled on LoadMasters with firmware version 7.1-16 and above.

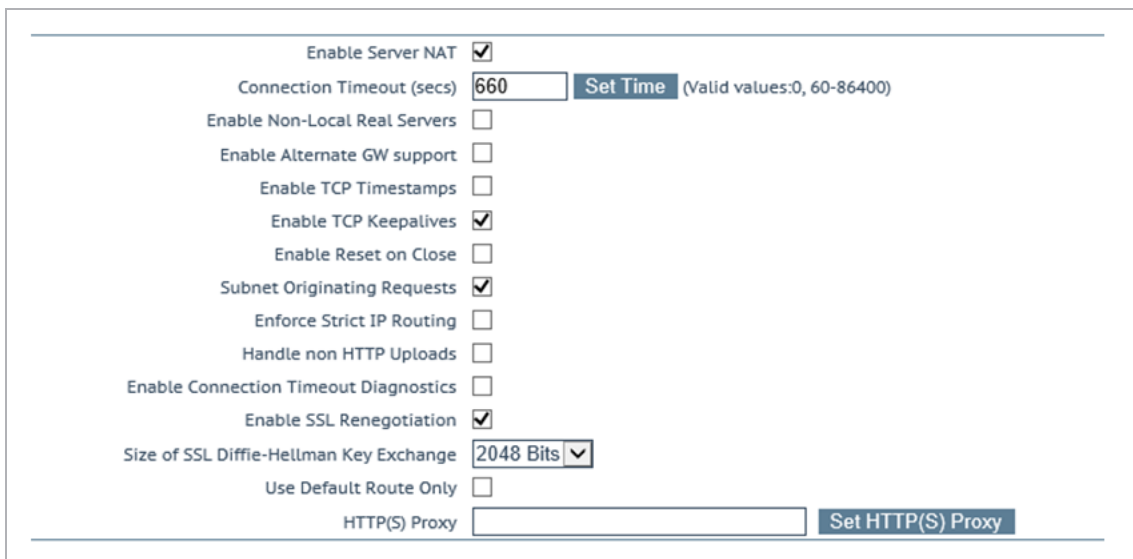
When **Subnet Originating Requests** is enabled, the LoadMaster routes traffic so that the Real Server sees traffic arriving from the LoadMaster interface that is in that network/subnet.

When **Subnet Originating Requests** is enabled globally, it is automatically enabled on all Virtual Services. If the **Subnet Originating Requests** option is disabled globally, you can choose whether to enable **Subnet Originating Requests** on a per-Virtual Service basis.

To enable **Subnet Originating Requests** globally, follow the steps below:

1. In the main menu of the LoadMaster Web User Interface (WUI), go to **System Configuration > Miscellaneous Options > Network Options**.

## 4 Configure the LoadMaster



Enable Server NAT ☒

Connection Timeout (secs)  [Set Time](#) (Valid values:0, 60-86400)

Enable Non-Local Real Servers ☐

Enable Alternate GW support ☐

Enable TCP Timestamps ☐

Enable TCP Keepalives ☒

Enable Reset on Close ☐

Subnet Originating Requests ☒

Enforce Strict IP Routing ☐

Handle non HTTP Uploads ☐

Enable Connection Timeout Diagnostics ☐

Enable SSL Renegotiation ☒

Size of SSL Diffie-Hellman Key Exchange

Use Default Route Only ☐

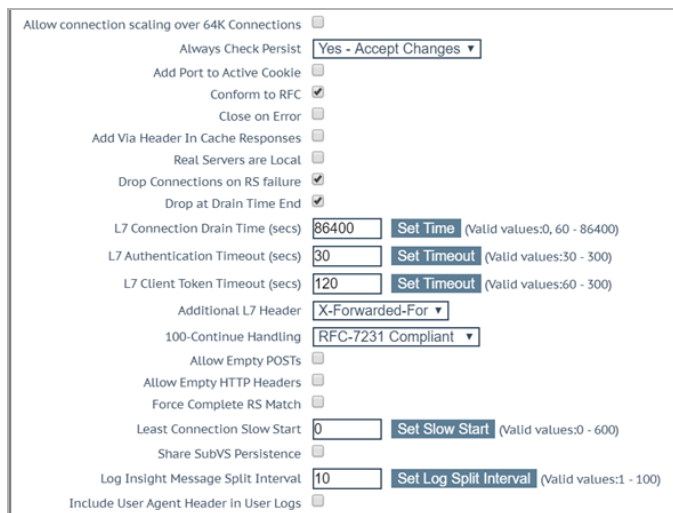
HTTP(S) Proxy  [Set HTTP\(S\) Proxy](#)

2. Select the **Subnet Originating Requests** check box.

## 4.2 Enable Check Persist Globally

It is recommended that you change the **Always Check Persist** option to **Yes – Accept Changes**. Use the following steps:

1. Go to **System Configuration > Miscellaneous Options > L7 Configuration**.



Allow connection scaling over 64K Connections ☐

Always Check Persist

Add Port to Active Cookie ☐

Conform to RFC ☒

Close on Error ☐

Add Via Header In Cache Responses ☐

Real Servers are Local ☐

Drop Connections on RS failure ☒

Drop at Drain Time End ☒

L7 Connection Drain Time (secs)  [Set Time](#) (Valid values:0, 60 - 86400)

L7 Authentication Timeout (secs)  [Set Timeout](#) (Valid values:30 - 300)

L7 Client Token Timeout (secs)  [Set Timeout](#) (Valid values:60 - 300)

Additional L7 Header

100-Continue Handling

Allow Empty POSTs ☐

Allow Empty HTTP Headers ☐

Force Complete RS Match ☐

Least Connection Slow Start  [Set Slow Start](#) (Valid values:0 - 600)

Share SubVS Persistence ☐

Log Insight Message Split Interval  [Set Log Split Interval](#) (Valid values:1 - 100)

Include User Agent Header in User Logs ☐

2. Click the **Always Check Persist** drop-down arrow and select **Yes – Accept Changes**.



### 4.3 Create the Apache HTTP Virtual Services

The following sections describe the recommended settings for the Apache HTTP Virtual Services.

#### 4.3.1 Create an Apache HTTP Virtual Service

The following are the steps involved and the recommended settings to configure the Apache HTTP Virtual Service:

1. In the main menu of the LoadMaster Web User Interface (WUI), go to **Virtual Services > Add New**.

Please Specify the Parameters for the Virtual Service.

Virtual Address

10.154.11.141

Port

80

Service Name (Optional)

Apache HTTP

Use Template

Select a Template

Protocol

tcp

Cancel

Add this Virtual Service

2. Type a valid **Virtual Address**.
3. Type **80** as the **Port**.
4. Enter a recognizable **Service Name**, such as Apache HTTP Virtual Service.
5. Click **Add this Virtual Service**.
6. Configure the settings as recommended in the following table:

Section	Option	Value
Standard Options	Persistence Mode	Active Cookie
	Timeout	1 Hour
	Cookie name	JSESSIONID
	Scheduling Method	least connection

Section	Option	Value
	Idle Connection Timeout	900

7. Add the Real Servers:

- Expand the **Real Servers** section.
- Click **Add New**.
- Enter the address of the relevant Real Server.
- Complete the other fields as required.
- Click **Add this Real Server** then click **OK** to the pop-up message.
- Repeat the steps above to add more Real Servers as needed, based on your environment.

### 4.3.2 Create an Apache HTTPS Virtual Service

The following are the steps involved and the recommended settings to configure the Apache HTTPS Virtual Service:

- In the main menu of the LoadMaster WUI, go to **Virtual Services > Add New**.

**Please Specify the Parameters for the Virtual Service.**

---

Virtual Address

Port

Service Name (Optional)

Use Template

Protocol

---

- Type a valid **Virtual Address**.
- Type **443** as the **Port**.
- Enter a recognizable **Service Name**, such as Apache HTTPS Virtual Service.
- Click **Add this Virtual Service**.

6. Configure the settings as recommended in the following table:

Section	Option	Value	Comment
Standard Options	Persistence Mode	Source IP Address	
	Timeout	1 Hour	
	Scheduling Method	least connection	
	Idle Connection Timeout	900	
Advanced Properties	Add a Port 80 Redirector VS	https://%h%s	Click <b>Add HTTP Redirector</b> . This automatically creates a redirect on port 80.

7. Add the Real Servers:

- Expand the **Real Servers** section.
- Click **Add New**.
- Enter the address of the relevant Real Server.
- Complete the other fields as required.
- Click **Add this Real Server** then click **OK** to the pop-up message.
- Repeat the steps above to add more Real Servers as needed, based on your environment.

### Create an Apache HTTPS Redirect Virtual Service

Clicking the **Add HTTP Redirector** button automatically creates a port 80 redirect Virtual Service. This is optional, but the purpose of this Virtual Service is to redirect any clients who have connected using HTTP to the HTTPS Virtual Service. Kemp also recommends changing the **Real Server Check Method** and **Persistence Mode** to **None**.

### 4.3.3 Create an Apache HTTPS Offloaded Virtual Service

The following are the steps involved and the recommended settings to configure the Apache HTTPS Offloaded Virtual Service:

- In the main menu of the LoadMaster WUI, go to **Virtual Services > Add New**.

Please Specify the Parameters for the Virtual Service.

Virtual Address

10.154.11.141

Port

443

Service Name (Optional)

Apache HTTPS Offlo

Use Template

Select a Template

Protocol

tcp

Cancel

Add this Virtual Service

- 2. Type a valid **Virtual Address**.
- 3. Type **443** as the **Port**.
- 4. Enter a recognizable Service Name, such as **Apache HTTPS Offloaded** Virtual Service.
- 5. Click **Add this Virtual Service**.
- 6. Configure the settings as recommended in the following table:

Section	Option	Value	Comments
Standard Options	Persistence Mode	Active Cookie	
	Timeout	1 Hour	
	Cookie name	JSESSIONID	
	Scheduling Method	least connection	
	Idle Connection Timeout	900	
SSL Properties	SSL Acceleration	Enabled	
	Cipher Set	BestPractices	
Advanced Properties	Add a Port 80 Redirector VS	https://%h% Click <b>Add HTTP Redirector</b> . This automatically creates a redirect on port 80.	

7. Add the Real Servers:

- a) Expand the **Real Servers** section.
- b) Click **Add New**.
- c) Enter the address of the relevant Real Server.
- d) Complete the other fields as required.
- e) Click **Add this Real Server** then click **OK** to the pop-up message.
- f) Repeat the steps above to add more Real Servers as needed, based on your environment.

### Create an Apache HTTPS Offloaded Redirect Virtual Service

Clicking the **Add HTTP Redirector** button automatically creates a port 80 redirect Virtual Service. This is optional, but the purpose of this Virtual Service is to redirect any clients who have connected using HTTP to the HTTPS Virtual Service. Kemp also recommends changing the **Persistence Mode** to **None**.

### 4.3.4 Create an Apache HTTPS Re-encrypt Virtual Service

The following are the steps involved and the recommended settings to configure the Apache HTTPS Re-encrypt Virtual Service:

1. In the main menu of the LoadMaster WUI, go to **Virtual Services > Add New**.

**Please Specify the Parameters for the Virtual Service.**

---

Virtual Address

Port

Service Name (Optional)

Use Template

Protocol

---

2. Type a valid **Virtual Address**.
3. Type **443** as the **Port**.
4. Enter a recognizable **Service Name**, such as Apache HTTPS Re-encrypt Virtual Service.

5. Click **Add this Virtual Service**.

6. Configure the settings as recommended in the following table:

Section	Option	Value	Comments
Standard Options	Persistence Mode	Active Cookie	
	Timeout	1 Hour	
	Cookie name	JSESSIONID	
	Scheduling Method	least connection	
	Idle Connection Timeout	900	
SSL Properties	SSL Acceleration	Enabled	
	Cipher Set	Best Practices	
Advanced Properties	Add a Port 80 Redirector VS	https://%h%s	Click <b>Add HTTP Redirector</b> . This automatically creates a redirect on port 80.

7. Add the Real Servers:

- Expand the **Real Servers** section.
- Click **Add New**.
- Enter the address of the relevant Real Server.
- Complete the other fields as required.
- Click **Add this Real Server** then click **OK** to the pop-up message.
- Repeat the steps above to add more Real Servers as needed, based on your environment.

### Create an Apache HTTPS Re-encrypt Redirect Virtual Service

Clicking the **Add HTTP Redirector** button automatically creates a port 80 redirect Virtual Service. This is optional, but the purpose of this Virtual Service is to redirect any clients who have connected

using HTTP to the HTTPS Virtual Service. Kemp also recommends changing the **Real Server Check Method** and **Persistence Mode** to **None**.

# References

Unless otherwise specified, the following documents can be found at <http://kemptechnologies.com/documentation>.

## **Virtual Services and Templates, Feature Description**



# Last Updated Date

This document was last updated on 27 July 2023.