Progress®
Kemp® LoadMaster®

Feature Description HTTP2

Generated on: 16 September 2024

Copyright

Visit the following page online to see Progress Software Corporation's current Product Documentation Copyright Notice/Trademark Legend: Product Documentation Copyright Notice & Trademarks | Progress

Table of Contents

Chapter 1: Introduction 4

Document Purpose 6

Intended Audience 6

Limitations 6

Support 7

Chapter 2: Enable HTTP/2 in a Virtual Service 8

HTTP/2 Pass-through Service Type 10

Chapter 3: Evaluating HTTP/2 12

Example Test Environment 13

Create the HTTP 1.1 Virtual Service 14

Create the HTTP/2 Virtual Service 15

Performing Tests. 19

Chapter 4: References 20

1

Introduction

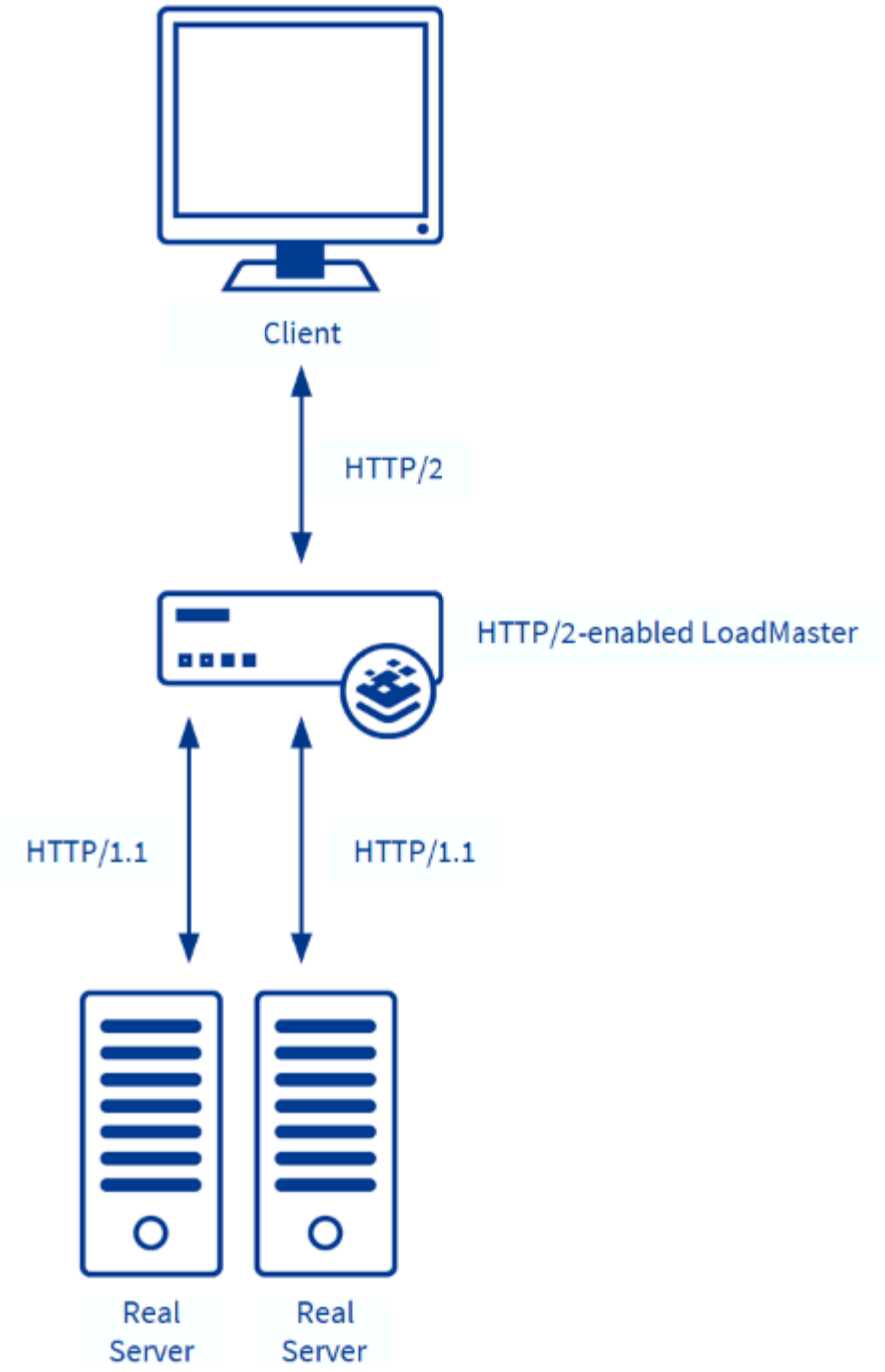
Introduction

Full HTTP/2 support is available across all Progress Kemp Application Delivery Controllers (ADCs) and can be enabled at the push of a button. Progress Kemp's HTTP/2 support greatly improves user experience on applications and websites, simplifying the performance upgrade path for site administrators.

HTTP/2 is the latest version of Hypertext Transfer Protocol (HTTP) and is designed to optimize the delivery of content on everything from websites to mobile apps leading to a much better end user experience. It is supported across modern browsers including the latest versions of Google Chrome, Mozilla Firefox, Apple Safari, Microsoft Internet Explorer, and Edge.

HTTP/2 has huge potential. There are a number of inflexibility, inefficiency and performance challenges with HTTP/1 that are solved by HTTP/2. Several of the benefits of HTTP/2 are below:

- **Multiplexing and concurrency:** Several requests can be sent in rapid succession in the same TCP connection. Responses can be received out of order – eliminating the need for multiple connections between the client and the server.
- **Stream dependencies:** The client can indicate to the server which of the resources are more important than others
- **Header compression:** HTTP header size is drastically reduced
- **Server push:** The server can send resources that the client has not yet requested



The LoadMaster also supports HTTP/2 – the LoadMaster can convert HTTP/2 traffic to HTTP/1.1 traffic in the back-end before it hits the Real Servers. These Real Servers do not need to have SSL enabled.

The HTTP/2 functionality in the LoadMaster provides optimizations, such as request pipelining and request multiplexing to reduce the request load on back-end servers. This results in a significantly improved end user experience when using a browser with HTTP/2 support. HTTP/2 works with a number of LoadMaster features, such as content switching, content caching, advanced persistence, header injection and the Web Application Firewall (WAF).

Related Links

- Document Purpose
- Intended Audience
- Limitations
- Support

Document Purpose

Document Purpose

The purpose of this document is to show you how to enable HTTP/2 in the LoadMaster and provide test website content for you to test the functionality against.

Intended Audience

Intended Audience

This document is intended to be used by anyone interested in enabling HTTP/2 in the LoadMaster.

Limitations

Limitations

Some limitations are listed below:

- Certain ciphers are not supported when using HTTP/2 – but these are automatically disabled when HTTP/2 support is enabled in the LoadMaster.
- If either NT LAN Manager (NTLM) or Kerberos authentication is enabled on a Virtual Service, HTTP/2 will be disabled. Similarly, if HTTP/2 support is enabled on a Virtual Service, NTLM/Kerberos authentication will be disabled.
- SAML authentication does not currently work correctly with HTTP/2.
- **TLS 1.2** must be enabled in the **Supported Protocols** in the **SSL Properties** section for HTTP/2 to work
- The **Process Responses** option in the **WAF Options** section cannot be enabled if HTTP/2 is enabled
- Content switching when matching inside a POST does not work with HTTP/2.
- When using HTTP/2, after killing a session the HTTP/2 client still has access to the application until the underlying connection is closed. Examples of closing the underlying connection are; the user closes the browser, the user does a hard reset (**Ctrl + F5**), or the connection reaches an idle timeout.
- When using HTTP/2, transparency is not used even if it is enabled. This is because one connection is mapped to multiple connections. HTTP/1.1 clients are still transparent when HTTP/2 and transparency are both enabled.

- Packet counts are not reported when using HTTP/2.

Support

Support

If you have any questions or need assistance, please contact Progress Kemp Support: https://support.kemptechnologies.com.

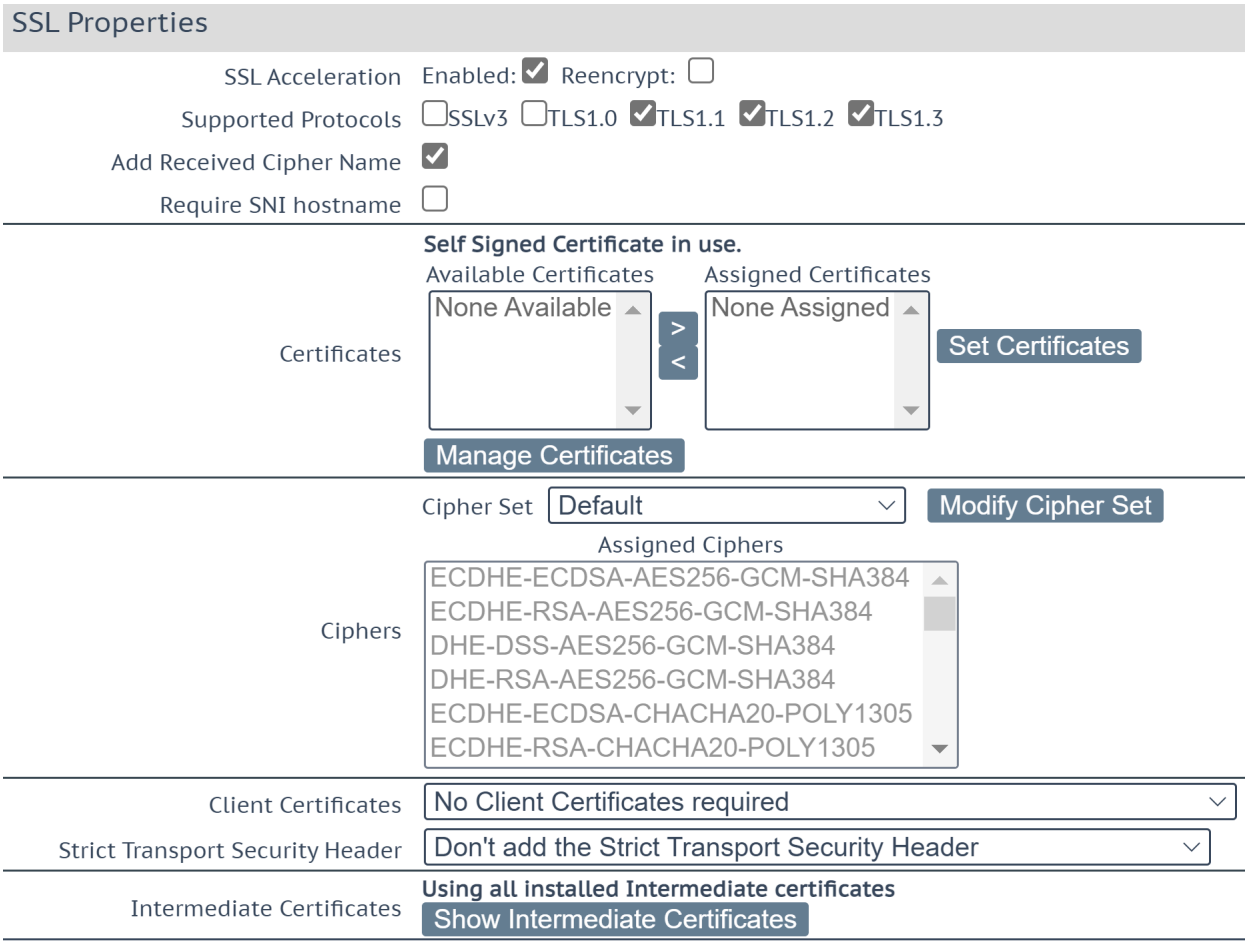
2

Enable HTTP/2 in a Virtual Service

Enable HTTP/2 in a Virtual Service

Follow the steps below to enable HTTP/2 in a Virtual Service in the LoadMaster:

1. In the LoadMaster Web User Interface (WUI), go to **Virtual Services > View/Modify Services**.
2. Click **Modify** on the relevant Virtual Service.
3. Expand the **SSL Properties** section.

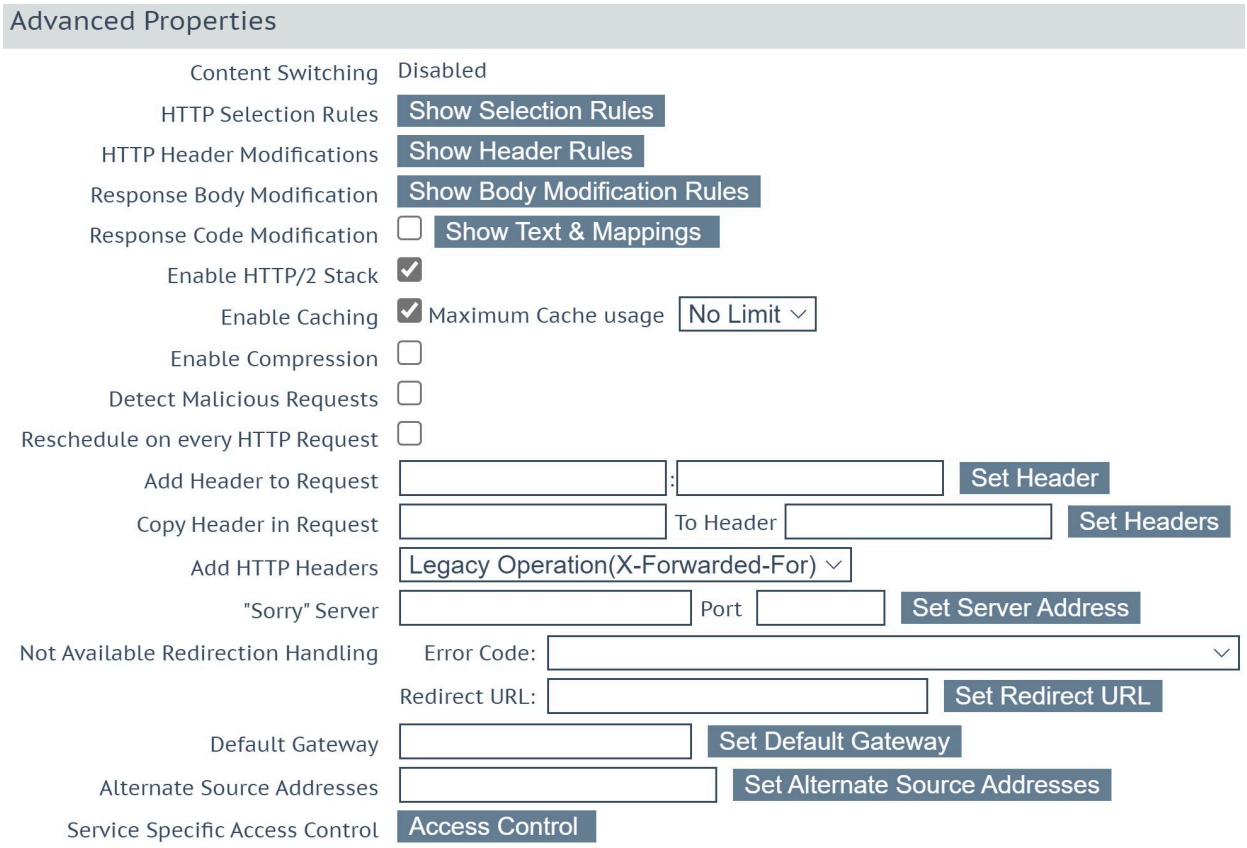


4. Select the **Enabled** check box.

Note: HTTP/2 is only available if **SSL Acceleration** is **Enabled**.

Note: HTTP/2 also works with SSL re-encryption, which helps with applications that require both encrypted flows in addition to L7 functionality.

5. Select **BestPractices** as the **Cipher Set**.
6. Expand the **Advanced Properties** section.



7. Select the **Enable HTTP/2 Stack** check box.
8. Configure any other settings as needed.

Note: As with HTTP/1, enabling caching improves performance.

Note: For details on each of the options in the WUI, refer to the Web User Interface (WUI), Configuration Guide.

Related Links

- HTTP/2 Pass-through Service Type

HTTP/2 Pass-through Service Type

HTTP/2 Pass-through Service Type

The previous sections describe how HTTP/2 is configured and works with a Virtual Service type of **HTTP-HTTP/2-HTTPS** – you select the **Enable HTTP/2 Stack** option under the Virtual Service's **Advanced Properties**, which enables HTTP/2 processing on the client side and HTTP/1.1 on the server side.

The LoadMaster also supports passing unmodified HTTP/2 traffic directly to the back-end Real Servers. This behavior is enabled by selecting the **HTTP/2 Pass-through** Virtual **Service Type** when creating a Virtual Service.

An **HTTP/2 Pass-through** Virtual Service simply passes incoming HTTP/2 traffic to the back-end servers without modification. Since the HTTP/2 traffic is not modified, it remains HTTP/2 traffic – it is not sent to the back-end servers as HTTP/1.1 traffic (as is done with the **Enable HTTP/2 Stack** option).

The **HTTP/2 Pass-through** alternative offers better performance, at the cost of having no ability to inspect and modify the encrypted traffic at the LoadMaster – which means the LoadMaster cannot perform operations like examining and modifying header values. This is why an **HTTP/2 Pass-through** Virtual Service provides fewer options than an **HTTP-HTTP/2-HTTPS** Virtual Service with the **Enable HTTP/2 Stack** enabled.

3

Evaluating HTTP/2

Evaluating HTTP/2

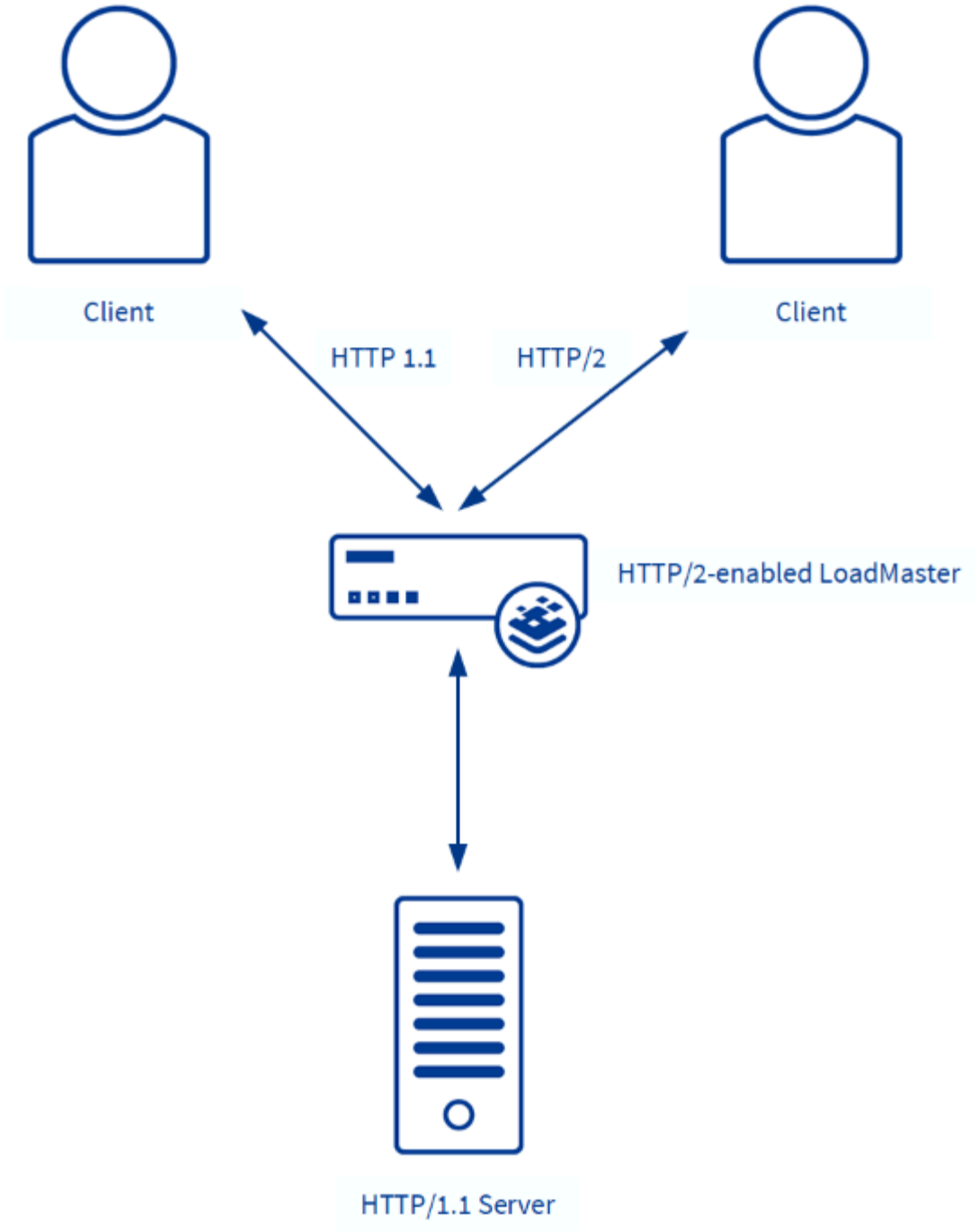
To assist with the evaluation of HTTP/2, we have provided a simple web page that consists of an image made up of 1024 individual image ‘tiles’. This page provides a visual guide to the optimization available with HTTP/2 as the tiled image renders much faster with HTTP/2. The sections below provide details about the test environment setup but to access the test page refer to the Performing Tests section.

Related Links

- Example Test Environment

Example Test Environment

Example Test Environment



The test environment is focused on providing a visual comparison of the same page being loaded using HTTP/2 and HTTP 1.1. A web page is hosted on a Real Server and the LoadMaster is configured with a

Virtual Service for HTTP/2 and a Virtual Service for HTTP 1.1. Both Virtual Services use the same Real Server.

The following are recommended in order to evaluate HTTP/2 correctly:

- **Browser:** Google Chrome is probably the best browser to use for HTTP/2 testing. To check what browsers support HTTP/2, please visit the following website: http://caniuse.com/#feat=http2
- **LoadMaster:** Use LoadMaster firmware version 7.1.35 or later.
- **Web Server:** Any HTTP 1.1 capable server will suffice. The sample page provided is a simple web page with multiple images.

For the test, Progress Kemp created a:

- HTTP 1.1 Virtual Service on port 80 with **SSL Acceleration** disabled.
- HTTP/2 Virtual Service on port 8080 with **SSL Acceleration** enabled.

In the example below, the Virtual Services are on 192.168.0.9 and the Real Server is on 192.168.0.10.

Related Links

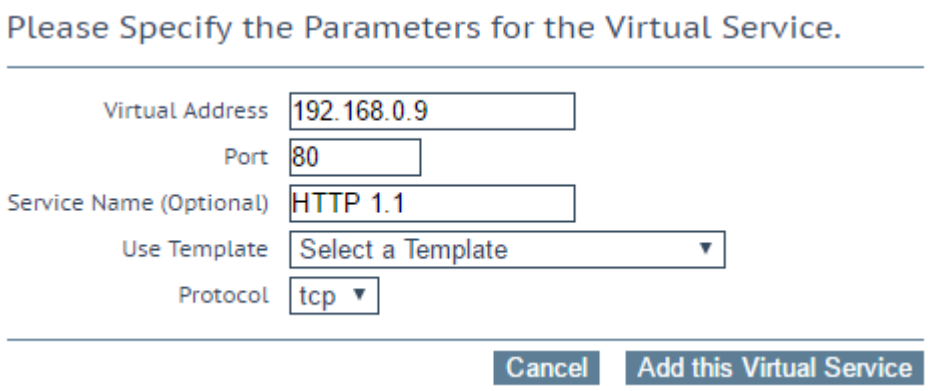
- Create the HTTP 1.1 Virtual Service
- Create the HTTP/2 Virtual Service
- Performing Tests

Create the HTTP 1.1 Virtual Service

Create the HTTP 1.1 Virtual Service

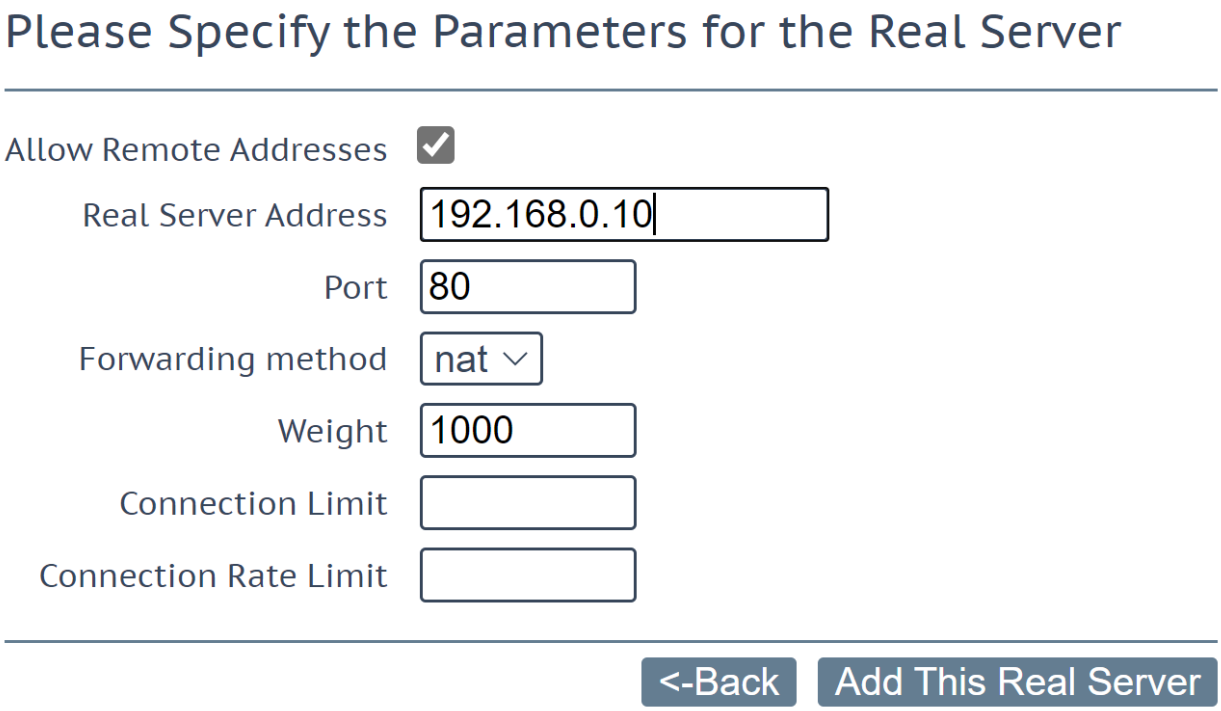
To create the HTTP 1.1 Virtual Service, follow the steps below:

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



2. Enter a valid IP address in the **Virtual Address** text box.
3. Enter **80** in the **Port** text box.
4. Enter a **Service Name**, for example **HTTP 1.1**.
5. Click **Add this Virtual Service**.

6. Configure any other details as needed.
7. Expand the **Real Servers** section.
8. Click **Add New**.



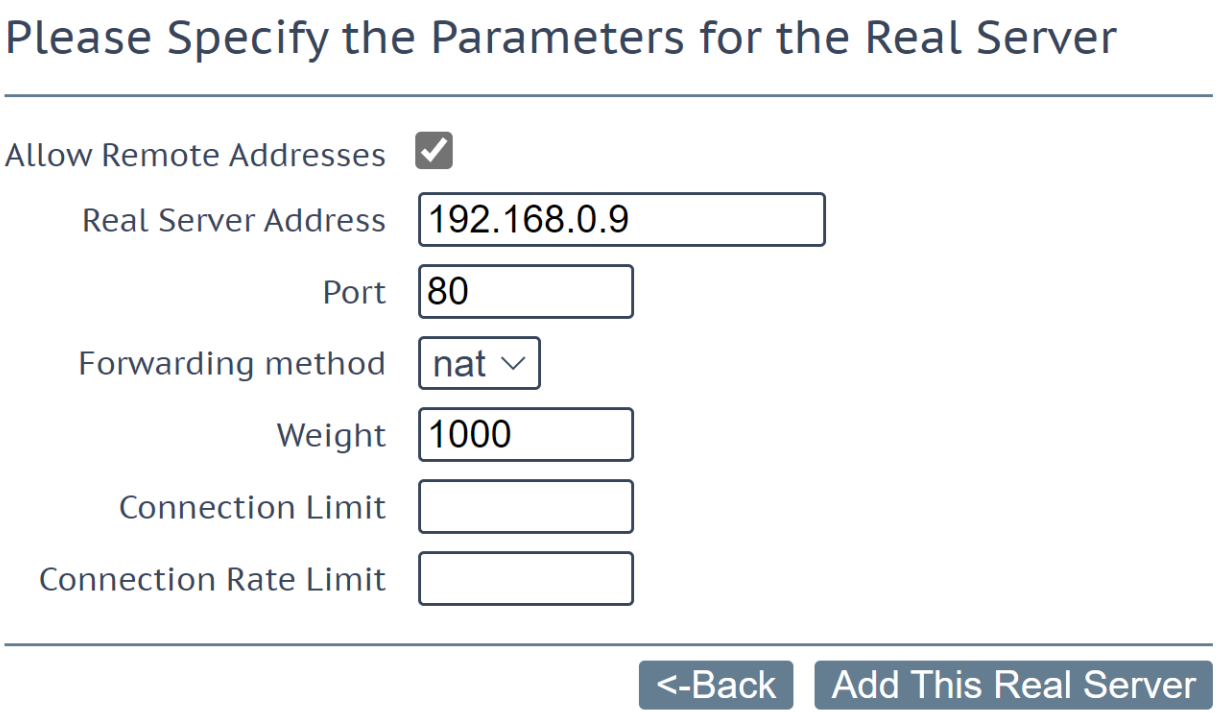
9. Enter the **Real Server Address**.
10. Enter **80** as the **Port**.
11. Click **Add This Real Server**.

Create the HTTP/2 Virtual Service

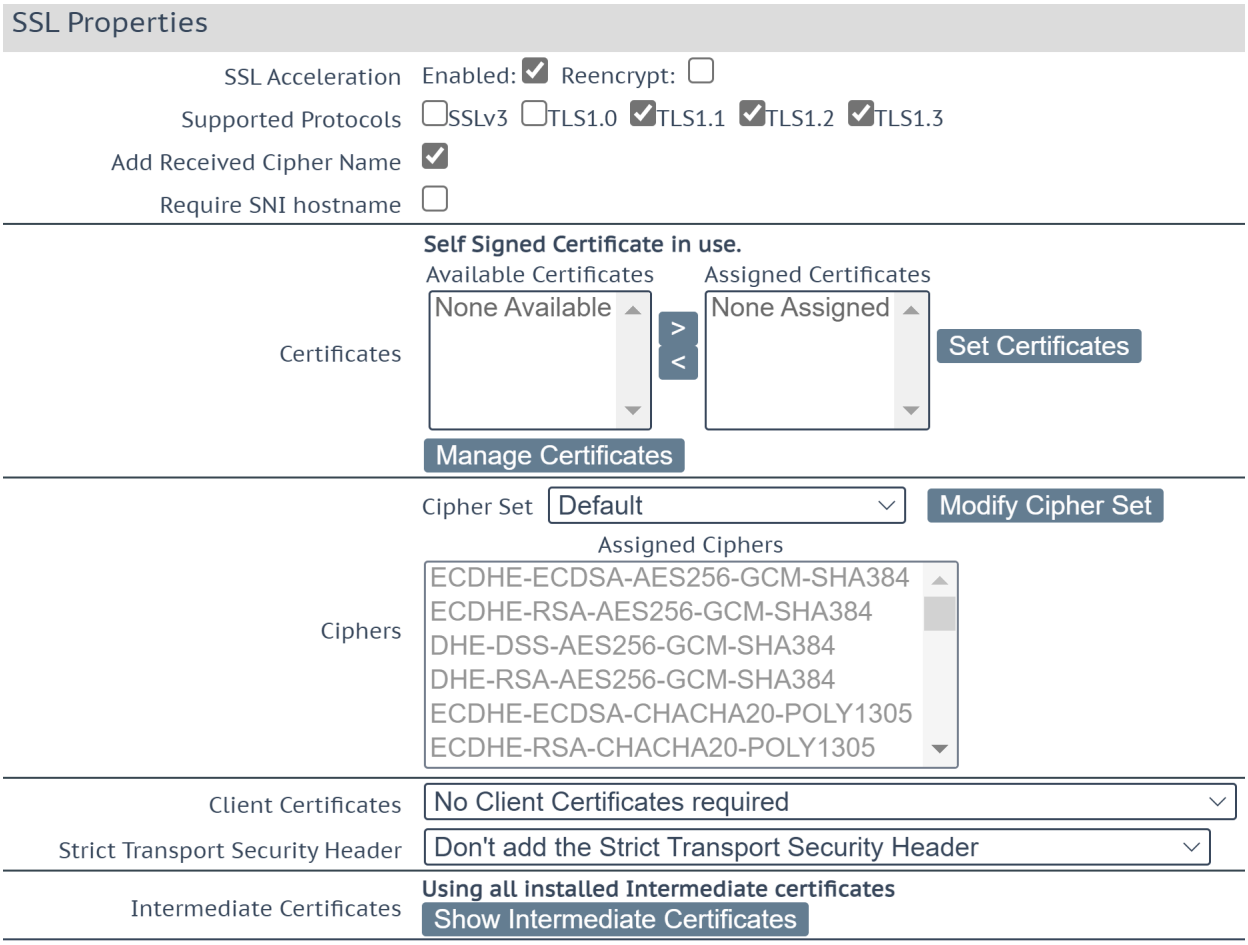
Create the HTTP/2 Virtual Service

To create the HTTP/2 Virtual Service, follow the steps below:

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



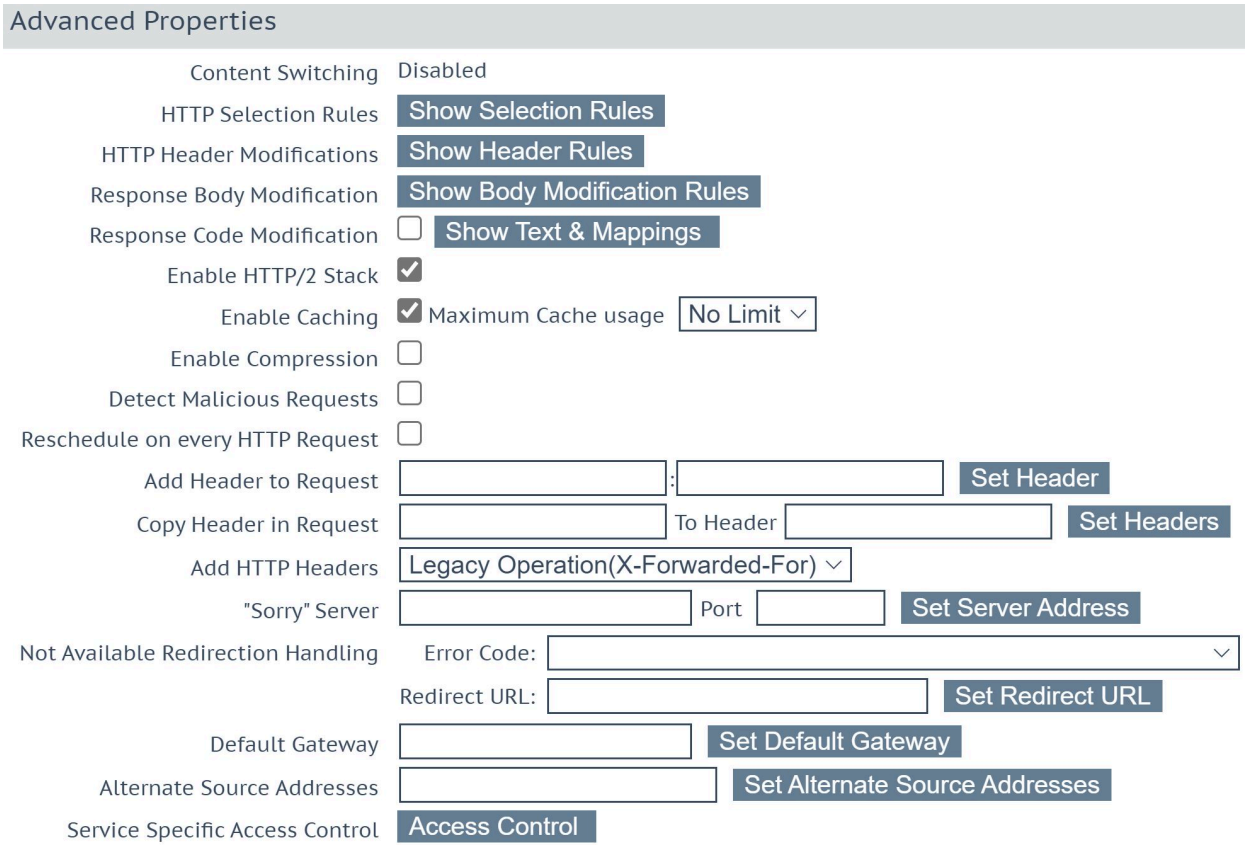
2. Enter a valid IP address in the **Virtual Address** text box.
3. Enter **8080** (or any other available port) in the **Port** text box.
4. Enter a **Service Name**, for example **HTTP2 Test**.
5. Click **Add this Virtual Service**.
6. Expand the **SSL Properties** section.



7. Tick the **Enabled** check box.

Note: SSL is mandatory for HTTP/2.

8. Select **BestPractices** as the **Cipher Set**.
9. Expand the **Advanced Properties** section.

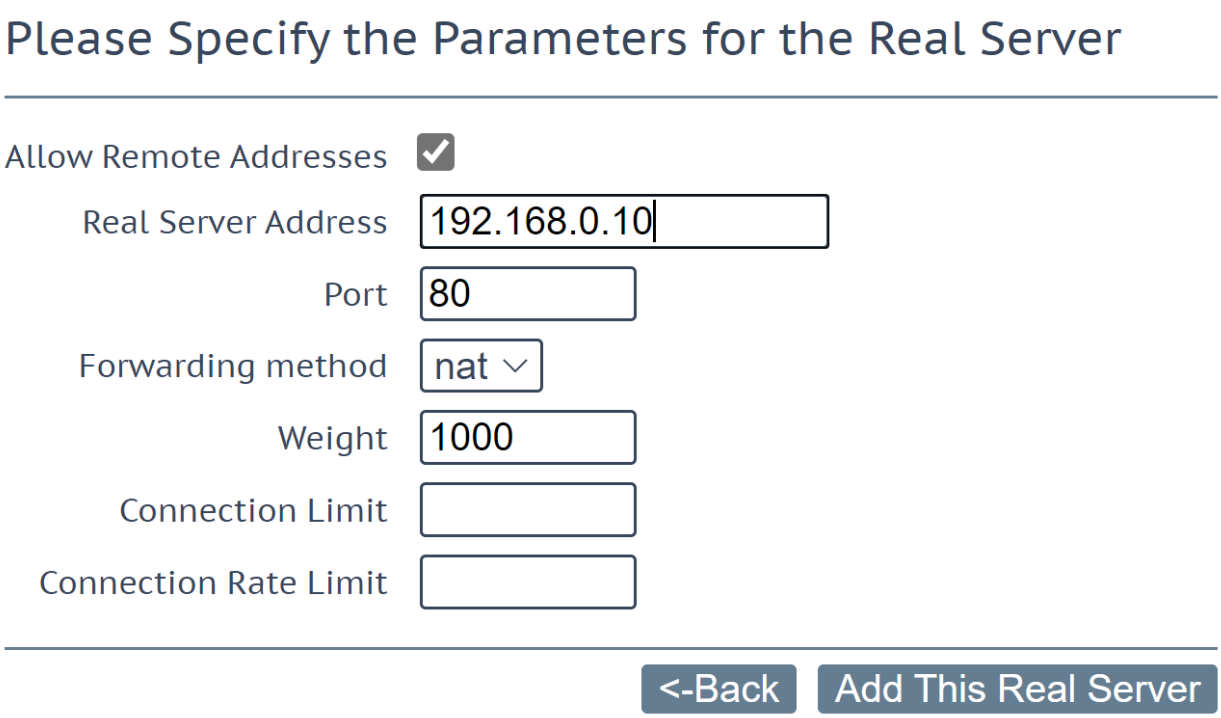


10. Select the **Enable HTTP/2 Stack** check box.
11. Configure any other details as needed.

Note: As with HTTP/1, enabling caching improves performance.

---

12. Expand the **Real Servers** section.
13. Click **Add New**.



14. Enter the **Real Server Address**.
15. Enter **80** as the **Port**.
16. Click **Add This Real Server**.

The HTTP/2 Virtual Service on the LoadMaster will communicate with the server using HTTP 1.1.

Performing Tests

Performing Tests

To test the performance gains from HTTP/2, the simplest way is to visualize the impact by using a web page which contains a large number of elements, such as images. Progress Kemp have provided a sample web page that displays an image made up of 1024 image 'tiles'. Simply browse to the HTTP/2 and HTTP 1.1 Virtual Services to see the difference in performance. Ensure you use a HTTP/2-enabled browser, such as Chrome, when performing this test.

The Progress Kemp HTTP/2 test page is available here: http://kemptechnologies.com/files/assets/tools/Kemp-TechPreview-HTTP2-TestPage.zip

Other tools and utilities for testing HTTP/2 are listed here: https://blog.cloudflare.com/tools-for-debugging-testing-and-using-http-2/

4

References

References

Unless otherwise specified, the following documents can be found at https://docs.progress.com/.

Web User Interface (WUI), Configuration Guide