



Infor Web UI High Availability Deployment

Important Notices

The material contained in this publication (including any supplementary information) constitutes and contains confidential and proprietary information of Infor.

By gaining access to the attached, you acknowledge and agree that the material (including any modification, translation or adaptation of the material) and all copyright, trade secrets and all other right, title and interest therein, are the sole property of Infor and that you shall not gain right, title or interest in the material (including any modification, translation or adaptation of the material) by virtue of your review thereof other than the non-exclusive right to use the material solely in connection with and the furtherance of your license and use of software made available to your company from Infor pursuant to a separate agreement, the terms of which separate agreement shall govern your use of this material and all supplemental related materials ("Purpose").

In addition, by accessing the enclosed material, you acknowledge and agree that you are required to maintain such material in strict confidence and that your use of such material is limited to the Purpose described above. Although Infor has taken due care to ensure that the material included in this publication is accurate and complete, Infor cannot warrant that the information contained in this publication is complete, does not contain typographical or other errors, or will meet your specific requirements. As such, Infor does not assume and hereby disclaims all liability, consequential or otherwise, for any loss or damage to any person or entity which is caused by or relates to errors or omissions in this publication (including any supplementary information), whether such errors or omissions result from negligence, accident or any other cause.

Without limitation, U.S. export control laws and other applicable export and import laws govern your use of this material and you will neither export or re-export, directly or indirectly, this material nor any related materials or supplemental information in violation of such laws, or use such materials for any purpose prohibited by such laws.

Trademark Acknowledgements

The word and design marks set forth herein are trademarks and/or registered trademarks of Infor and/or related affiliates and subsidiaries. All rights reserved. All other company, product, trade or service names referenced may be registered trademarks or trademarks of their respective owners.

Publication Information

Release: Infor Web UI

Publication date: December 31, 2012

Document code: B0072C US

Contents

About this guide	7
Intended audience.....	7
Related documents	7
Terminology	8
Contacting Infor.....	9
Chapter 1 Introduction.....	11
What is a high-available environment?	11
Why implement high availability?	11
Availability of the total environment.....	11
Additional costs and knowledge.....	12
Scope of this document	12
Chapter 2 Deployment recommendations	13
Introduction	13
Customer deployment recommendations.....	13
Network load balancer recommendations	13
Example scenarios.....	14
High-availability scenario 1	14
High-availability scenario 2	16
Infor Web UI issues around high availability	17
User context files.....	17
Auto-complete functionality	17
Older versions of Infor Web UI.....	18
Chapter 3 Example scenarios	19
Introduction	19
Horizontal clustering scenario	19

Installing the horizontal clustering scenario	21
Horizontal and vertical clustering scenario	22
Installing the horizontal and vertical clustering scenario	24
Post installation	26
Available Infor Web UI links	27
Tomcat administration console	27
Load balancer status console	27

About this guide

This white paper describes guidelines and best practices for implementing Infor Web UI 10.x in a high-available environment. Infor Web UI is a web-based interface for all Infor ERP versions.

This document is an addition to *Infor Workspace 10.x with Infor Web UI 10.x - Sizing and Deployment Guide (B0032x US)*, which gives performance guidelines and sizing information for the Infor10 Workspace and Infor Web UI server.

A separate sizing document (B0045x US) describes the sizing information for the Infor LN application server.

For the latest information about Infor Web UI, refer to the Infor Web UI release notes via Infor Xtreme solution 22881482.

Intended audience

This white paper is intended for Infor consultants, partners and customers who are responsible for implementing Infor Web UI. This document contains information that can help system administrators to setup their Infor Web UI environment in a high-available environment.

Related documents

You can find the documents in the product documentation section of the Infor Xtreme Support portal, as described in "Contacting Infor" on page **Error! Bookmark not defined..**

Reference	Document name
B0032x US	Infor Workspace 10.x with Infor Web UI 10.x - Sizing and Deployment Guide
U9484x US	Infor Enterprise Server Extensions - Installation Guide
U8715x US	Infor Web UI - Installation and Configuration Guide

Reference	Document name
B0045x US	Sizing Guide for Infor LN
B0067x US	Infor Web UI 10.x - Sizing and Deployment for a Thin Client Solution
B0073x US	Deploying Infor LN in a Virtualized environment

Terminology

Term	Description
Horizontal clustering	<p>The installation of multiple Infor Web UI instances on different physical servers. Horizontal clustering is also referred to as a “scale-out” scenario.</p> <p>Horizontal clustering protects users from hardware failures</p>
Vertical clustering	<p>The installation of multiple Infor Web UI instances on a single physical server. Vertical clustering is also referred to as a “scale-in” scenario.</p> <p>Implementing vertical clustering provides additional scalability of the application on each of the servers</p>
Apache web server	An open source web server that is suited to serve static content. Apache can be configured to pass requests for dynamic content to another server; in this case, Apache is called the front-end web server
Tomcat	An open source software implementation of the Java Servlet and Java Server Pages technologies. Additionally, Tomcat has its own built-in web server with default port 8312. The Infor Web UI installer includes a Tomcat server
Session stickiness or client affinity	Most applications store session information in web server memory. Therefore, subsequent user requests must always be redirected to the same web server. This action is called “session stickiness” or “client affinity”
Mod_JK	An Apache module which connects the Apache web server to Tomcat
PBC	Performance and Benchmark Center. This department is responsible for this white paper
Tomcat worker	A Tomcat instance that is waiting to execute servlets on behalf of some web server. In this white paper there is an Apache web server forwarding servlet requests to a Tomcat worker.
AJP	The communication protocol used between Apache and Tomcat

Term	Description
Network Load balancer (NLB)	NLB allows you to combine application servers to provide a level of scaling and availability that is not possible with an individual server. NLB distributes incoming client requests among the servers in the cluster or farm to more evenly balance the workload of each server and prevent overload on any one server. To client computers, the NLB cluster appears as a single server that is highly scalable and fault tolerant.

Contacting Infor

If you have questions about Infor products, go to the Infor Xtreme Support portal at www.infor.com/inforxtreme.

If we update this document after the product release, we will post the new version on this Web site. We recommend that you check this Web site periodically for updated documentation.

If you have comments about Infor documentation, contact documentation@infor.com.

What is a high-available environment?

A high available environment consists of at least two physically separated servers, each of which has an Infor Web UI installation. These servers are placed in a logical unit called a “cluster.”

In a normal situation, the users are assigned to a server in the cluster using a round robin mechanism; this mechanism ensures that the user load is “balanced” amongst all servers in the cluster. If one of the servers in the cluster fails because of hardware or software problems, the users are automatically redirected to another server in the cluster.

Why implement high availability?

Implementing a single Infor Web UI server results in a single point of failure, because, if hardware or software problems occur on this server all users are unable to access the Infor LN environment. By implementing Infor Web UI in a high-available environment, a single server failure of the webserver will not disrupt user access to Infor LN.

Implementing high availability is recommended only for business critical medium- and large-sized enterprises, because of the additional costs and expertise involved.

Availability of the total environment

This white paper describes high availability for Infor Web UI, but there are other components in the Infor LN environment which can be a single point of failure. Some of these components are as follows:

- The network (LAN/WAN)
- Power and cooling (Airco)
- The Infor LN application server
- The Infor LN database server
- The License Server

- Storage

For a high available Infor LN environment, all of the above components must be made high available.

Additional costs and knowledge

Implementing a high-available environment is costly and requires additional knowledge, including the following:

- Additional servers and load balancers
- Additional software licenses
- Training of system and application managers

Scope of this document

The following implementation choices were made for this white paper:

- This document does not apply to Infor Web UI installations running in Workspace/Sharepoint. The described configuration is only applicable for standalone Web UI installations.
- Use of a cluster with two physical web servers.
- Use of a Microsoft Network Load Balancer (NLB). This network load balancer can be replaced by any software or hardware load balancer.
- Use of Apache web servers; Apache is used as the front-end web server.
- Use of Tomcat servers for Infor Web UI; a Tomcat server is a servlet engine and has a simple build-in web server. Each Tomcat server contains an Infor Web UI installation.

Not included in the scope of this white paper is the deployment of Infor Web UI in IBM Websphere.

Introduction

This chapter contains deployment recommendations for Infor Web UI in high-available environments.

Customer deployment recommendations

For availability reasons, Infor recommends that you deploy Infor Web UI in a high-available environment for business critical medium- and large-sized enterprises. For the deployment, you must do the following:

- Install at least two different Tomcat servers, each of which must have Infor Web UI installed.
- Install the Tomcat servers on different physical hardware.
- In case a different front-end web server is used, for example, Apache, install at least two different front-end web servers, each on different physical hardware.
- Use a high-available load balancer.

Network load balancer recommendations

An important part of the configuration is the network load balancer. The network load balancer is a single point of failure and must be made redundant.

The network load balancer must be service aware. The most simple load balancers will check only whether servers in the cluster are accessible by network (TCP/IP). Requests will still be passed to a server experiencing software failures, for example, in case the web server has crashed. An example of such a load balancer is the Microsoft Network Load balancer (NLB).

A service-aware network load balancer must be configured to check both the status of the Apache web server and the Tomcat server. In case of software failure, the entire server must be put offline in the cluster.

To summarize, the recommendations for the network load balancer are as follows:

- 1 Ensure that the network load balancer is redundant and not a single point of failure.
- 2 Select a service-aware network load balancer.
- 3 Configure the network load balancer to check the availability of both the Apache web server (default port 80) and the Tomcat server (default ports 8132 and 8009) of each server in the cluster.

Example scenarios

The following common scenarios for deploying Infor LN in a high-available configuration are described in this white paper:

- High-availability scenario 1: Horizontal clustering. Horizontal clustering is also referred to as a “scale-out” scenario.
- High-availability scenario 2: Both horizontal and vertical clustering. Vertical clustering is also referred to as a “scale-in” scenario.

High-availability scenario 1

Figure 1 shows an example of a high-available web server cluster, using two different physical servers. Each web server has a single Web UI instance installed. This scenario uses horizontal clustering, and is also referred to as a “scale-out” scenario.

This scenario can be extended by adding more web servers to the cluster, up to the number of servers that are supported by the load balancer.

In horizontally clustered configurations, the use of the Apache web server is not mandatory. You can directly send all user requests to the Tomcat servers.

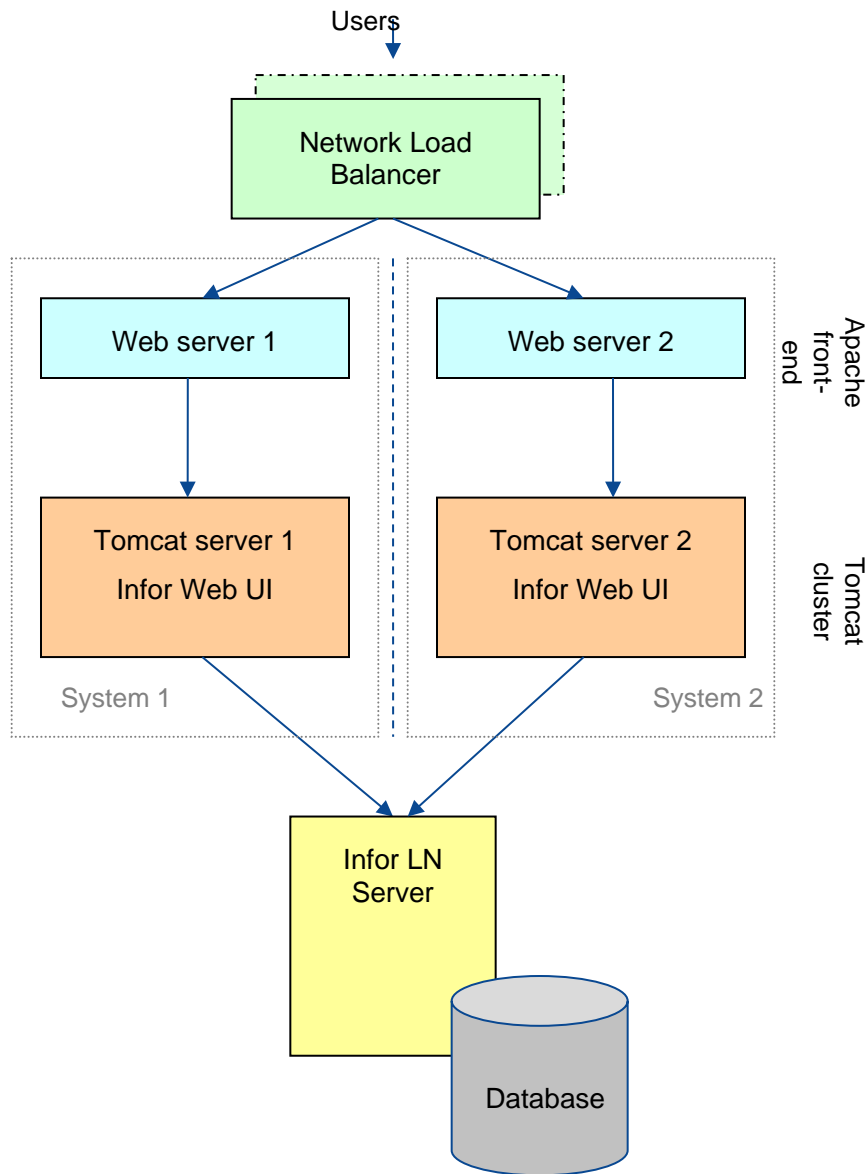


Figure 1: Horizontal clustering scenario

High-availability scenario 2

Figure 2 shows an example of a high-available web server cluster, using two different physical servers. Each web server has two Web UI instances installed. This scenario uses horizontal clustering and vertical clustering.

The Apache load balancer is used to balance the load between the Web UI instances. For vertical clustering, the use of a front-end web server such as Apache is mandatory.

This scenario can be extended by adding more web servers to the cluster, up to the number of servers that are supported by the load balancer. You can install more Tomcat instances per server.

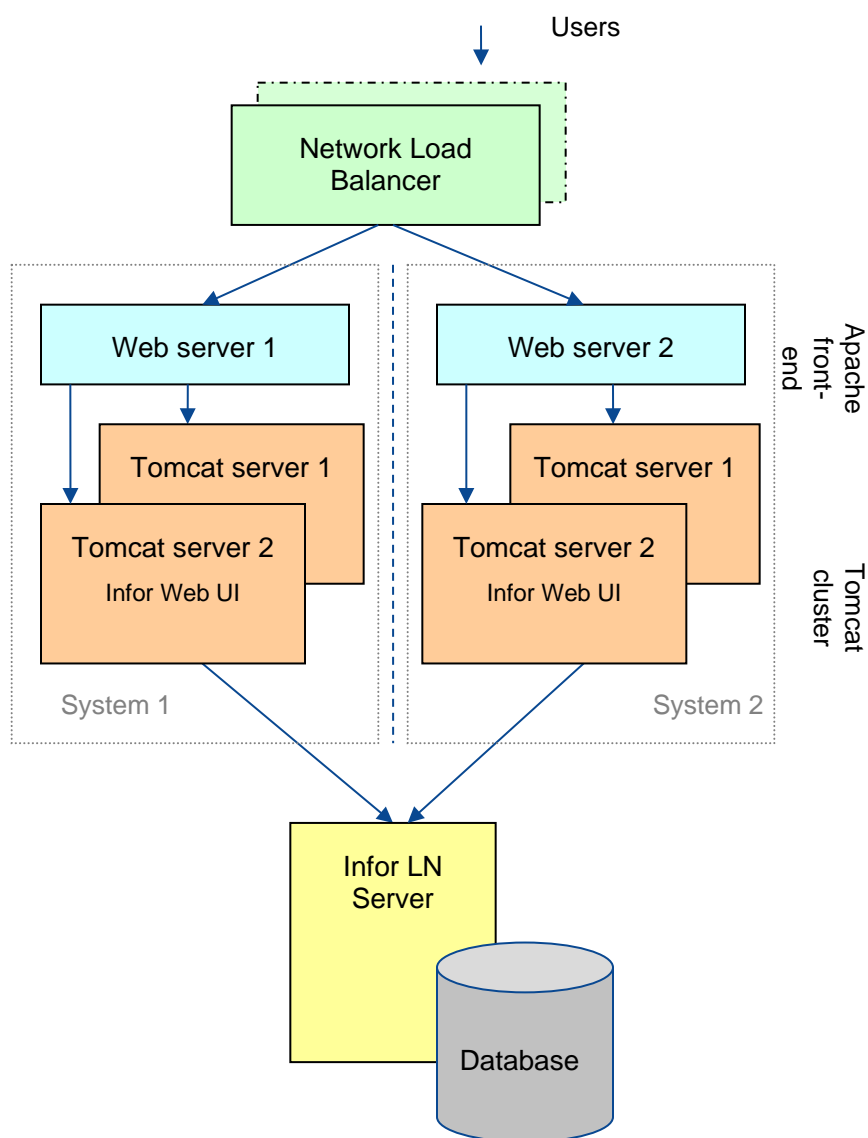


Figure 2: Horizontal clustering and vertical clustering scenario

Infor Web UI issues around high availability

Infor Web UI is not completely fault tolerant when it is deployed in a cluster. Because of the storage of user state on the web server, the user connection will be lost if a system failure occurs. Therefore, users must log in to the application again.

The Infor Web UI maintains user state in the following ways:

- In memory session state on the Tomcat server.
- User context files stored on the Tomcat server in the user context directory.

The following restrictions are due to the in-memory session state:

- Subsequent user requests must always be handled by the same web server; this functionality is called “sticky sessions” or “session affinity” and is available in the load balancer.
- If a web server failure occurs, the connected users must log in again on a different web server. This action will start a new bshell on the Infor LN server. The existing Infor LN bshell of this user will be closed because the UI connection is lost; therefore, unsaved work in the bshell will be lost.

User context files

User-related files are stored in the Tomcat server directory and contain user context information. The user context files contain, for example, the configured Web UI skin and the used Infor LN environment. We recommend that you make the user context files available on all Tomcat servers in the cluster.

The following are options to ensure that the user context files are available on all web servers:

- Replicate and synchronize user context directories between Tomcat servers. There are several tools available to synchronize directories between servers.
- Use a shared directory for the user context files, for example, SAN, NAS, or a network share.

The user context files are stored on each Tomcat server in the following directory:

`\Infor\ERPLN\ese\Webtop\config\usercontexts`

Auto-complete functionality

Auto-complete functionality in Infor Web UI shows the user a list of possible values that can be specified for certain fields. For example, when the user enters a business partner, a list of possible values is displayed.

Auto-complete functionality is enabled by default in all Infor Web UI versions.

As of Infor Web UI 8.5 and Infor Enterprise Server 8.5, auto-complete functionality is completely implemented in the Infor LN bshell. Therefore, there are no web server restrictions for using the functionality in a high-available environment.

Older versions of Infor Web UI

In Infor Web UI 8.4 and later, the auto-complete functionality is implemented in the Tomcat server. The Tomcat server stores information in an in-memory database and a number of files in the following Tomcat server directory:

\\Infor\ERPLN\ese\Webtop\config\usercontexts

The auto-complete database is bound to the Tomcat instance and cannot be replicated or synchronized with the other Tomcat instances.

Introduction

This chapter describes the setup of two common scenarios of Infor Web UI in a high-available configuration.

Horizontal clustering scenario

The scenario shown in Figure 3 describes the setup of two servers, each of which has Apache web server, Tomcat server, and Infor Web UI installed. This scenario is commonly referred to as horizontal clustering or a “scale-out” scenario.

The Apache mod_JK module is used for communication between Apache and Tomcat. You can also install this configuration without the use of the Apache web server, and directly send all requests to the Tomcat server included in the Infor Web UI installation.

In this scenario, Apache is configured to pass all requests to Tomcat. You can have Apache serve the static content, and pass only the dynamic servlet requests to the Tomcat instances.

A network load balancer is used to balance the users between the web servers.

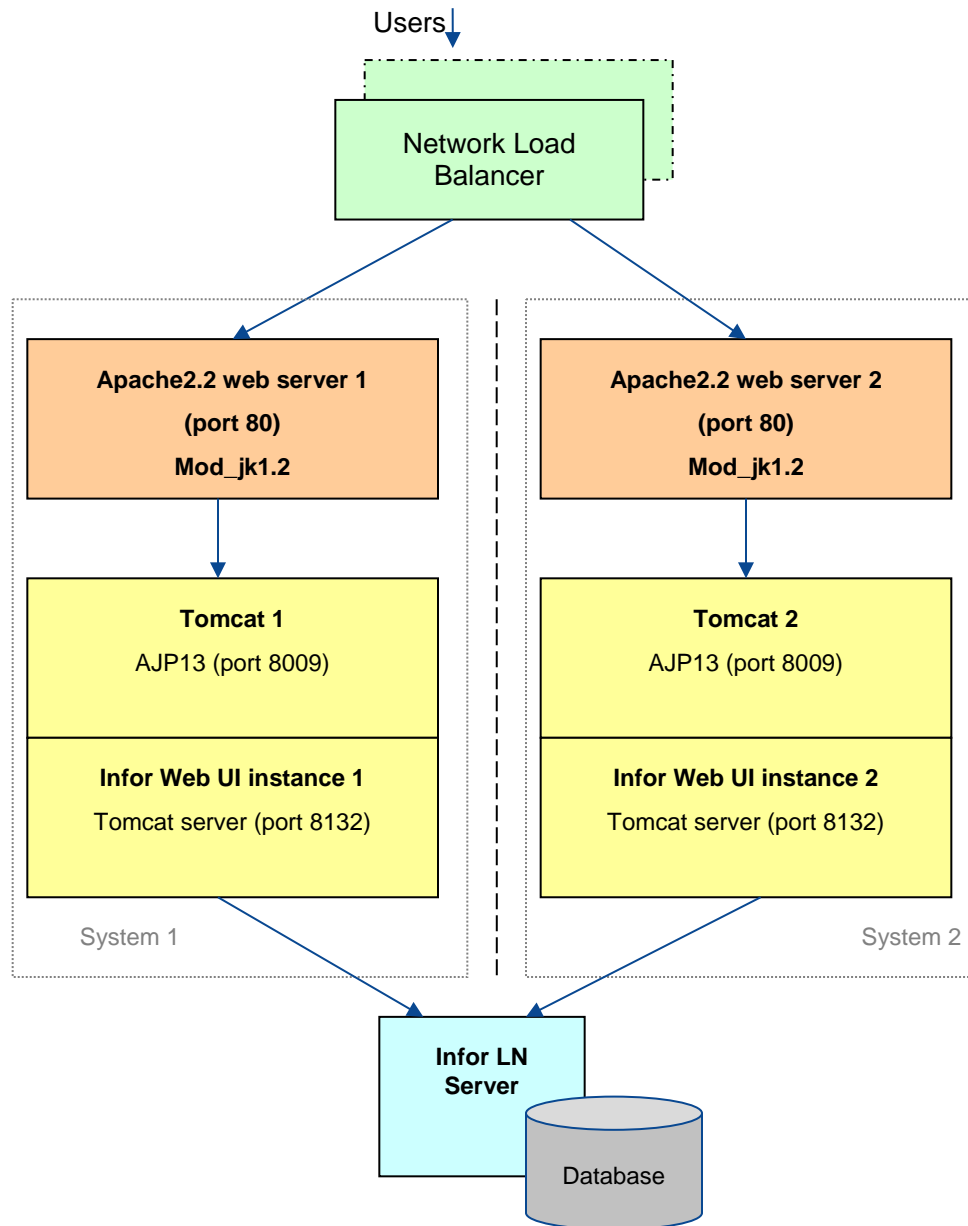


Figure 3: Horizontal clustering scenario

Installing the horizontal clustering scenario

To install the horizontal clustering scenario, complete the following steps:

- 4 Install the network load balancer. The load balancer must be configured using single client affinity, which means all requests from a client are sent to the same web server. The network load balancer is a single point of failure and must be made redundant.
- 5 Complete the following steps for all web servers in the configuration:
- 6 Install Apache 2.2.x
- 7 Download and install mod_jk1.2 in the Apache modules directory:
 - Filename: mod_jk-1.2.30-httpd-2.2.3.so
 - Directory: \Program Files (x86)\Apache Software Foundation\Apache2.2\modules
 - Rename the module to **mod_jk2.so**
- 8 Add the following to the Apache configuration file:
 - Filename: httpd.conf
 - Directory: \Program Files (x86)\Apache Software Foundation\Apache2.2\conf

```
LoadModule jk_module modules/mod_jk2.so

# Where to find workers.properties
JkWorkersFile conf/workers.properties
# Where to put jk shared memory
JkShmFile      conf/mod_jk2.shm
# Where to put jk logs
JkLogFile      logs/mod_jk2.log
# Set the jk log level [debug/error/info]
JkLogLevel     error
# Select the timestamp log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# All requests go to worker1
JkMount /* worker1
```

- 9 Create an Apache configuration file:
 - Filename: workers.properties
 - Directory: \Program Files (x86)\Apache Software Foundation\Apache2.2\conf

```
# the list of workers
worker.list= worker1

# worker "worker1" will talk to Tomcat1
worker.worker1.type=ajp13
worker.worker1.host=tomcatserver
worker.worker1.port=8009
worker.worker1.lbfactor=1
```

- 10 Install the Infor Web UI. The setup includes the Tomcat server.

- 11 Edit the Tomcat configuration file (\Infor\ERPLN\ese\apache-tomcat-<version>\conf\server.xml) and enable the AJP port. By default, this port is enabled. For example, add or uncomment the following lines to enable AJP port 8009:

```
<!-- Define an AJP 1.3 Connector on port 8009 -->  
<Connector port="8009" enableLookups="false"  
redirectPort="8443" protocol="AJP/1.3" />
```

- 12 Restart the Tomcat and Apache services.
- 13 Refer to the post-installation steps later in this chapter.

Horizontal and vertical clustering scenario

The scenario shown in Figure 4 is an extension of the first scenario, and includes both horizontal and vertical clustering.

The scenario describes the setup of two servers, each of which has Apache web server, Tomcat server, and Infor Web UI installed.

Vertical clustering is achieved by installing a second Tomcat server and Infor Web UI installation on each of the servers.

The Apache mod_JK module is used for communication between Apache and Tomcat. The balancer included in the mod_JK module is used to balance the load between the Tomcat instances. In this configuration, the use of Apache is mandatory.

A special worker called “jkstatus” is included in the installation for monitoring the Tomcat instance availability.

In this scenario, Apache is configured to pass all requests to Tomcat. You can have Apache serve the static content, and pass only the dynamic servlet requests to the Tomcat instances.

A network load balancer is used to balance the users between the web servers.

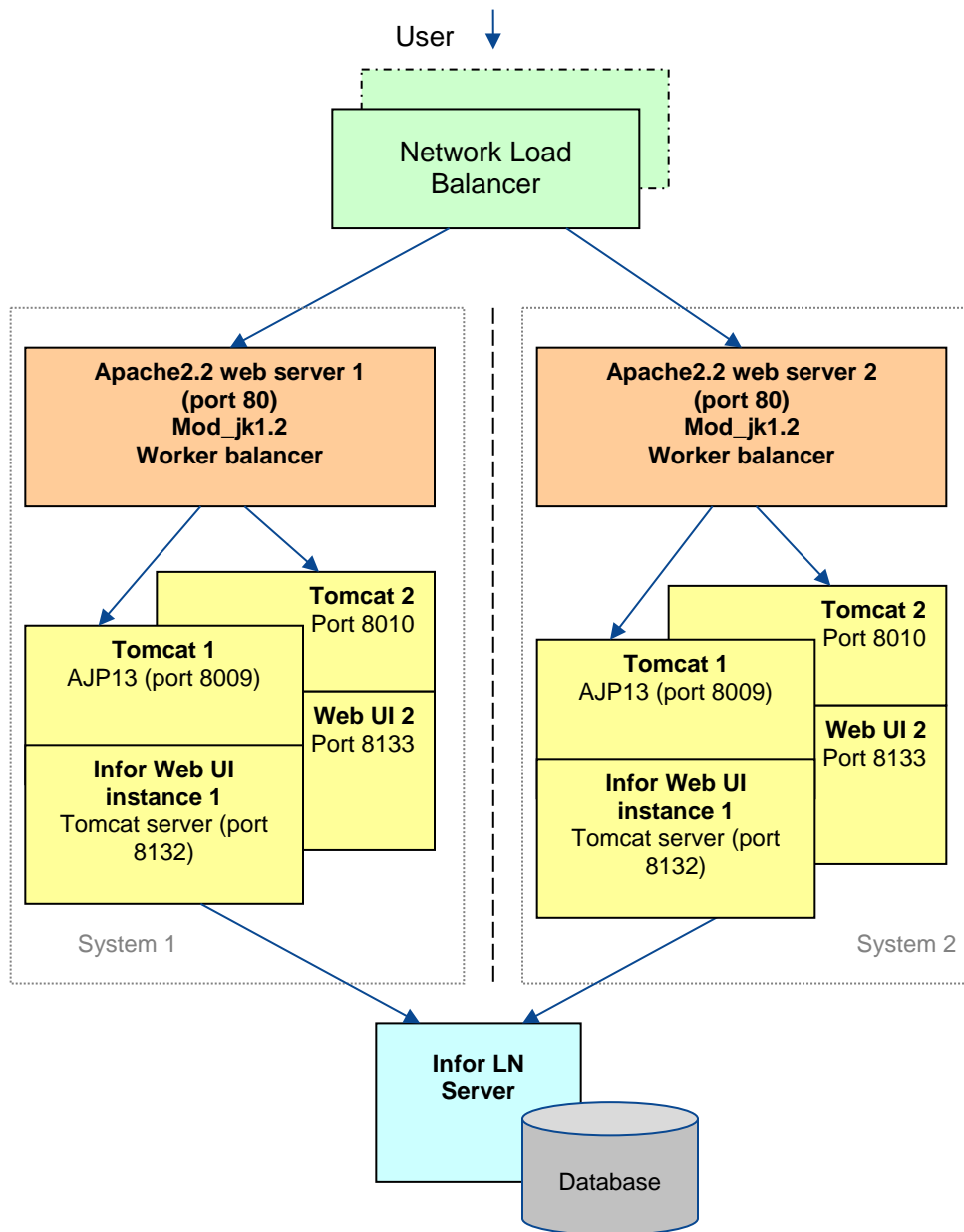


Figure 4: Horizontal and vertical clustering scenario

Installing the horizontal and vertical clustering scenario

To install the horizontal and vertical clustering scenario, complete the following steps:

- 1 Install the network load balancer. The load balancer must be configured using single client affinity, which means all requests from a client are sent to the same web server. The network load balancer is a single point of failure and must be made redundant.

Complete the following steps for all web servers in the configuration:

- 2 Install Apache 2.2.x
- 3 Download and install mod_jk1.2 in Apache modules directory:
 - Filename: mod_jk-1.2.30-httpd-2.2.3.so
 - Directory: \Program Files (x86)\Apache Software Foundation\Apache2.2\modules
 - Rename the module to **mod_jk2.so**
- 4 Add the following to the Apache configuration file:
 - Filename: httpd.conf
 - Directory: \Program Files (x86)\Apache Software Foundation\Apache2.2\conf

```
LoadModule jk_module modules/mod_jk2.so

# Add the jkstatus mount point
JkMount /jkmanager/* jkstatus

# Where to find workers.properties
JkWorkersFile conf/workers.properties
# Where to put jk shared memory
JkShmFile      conf/mod_jk2.shm
# Where to put jk logs
JkLogFile      logs/mod_jk2.log
# Set the jk log level [debug/error/info]
JkLogLevel     error
# Select the timestamp log format
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "
# All requests go to the loadbalancer
JkMount /* loadbalancer
```

- 5 Create an Apache configuration file:
 - Filename: wokers.properties
 - Directory: \Program Files (x86)\Apache Software Foundation\Apache2.2\conf


```
# the list of workers
worker.list= worker1, worker2, loadbalancer, jkstatus

# worker "worker1" will talk to Tomcat1
worker.worker1.type=ajp13
worker.worker1.host=tomcatserver
worker.worker1.port=8009
worker.worker1.lbfactor=1

# worker "worker2" will talk to Tomcat2
worker.worker2.type=ajp13
worker.worker2.host=tomcatserver
worker.worker2.port=8010
worker.worker2.lbfactor=1

# Defines a worker named "loadbalancer" that loadbalances
several Tomcat processes transparently.
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=worker1,worker2
worker.loadbalancer.sticky_session=True

# Define a 'jkstatus' worker using status
worker.jkstatus.type=status
```

- 6 Perform two separate installations of the Infor Web UI; the setup includes the Tomcat server. Each installation must use a different directory. The port numbers in the following table must be assigned during the installation:

	Port first installation	Port Second installation
HTTP	8312	8313
AJP	8009	8010
Shutdown	8005	8006

- 7 Edit the Tomcat configuration file (\Infor\ERPLN\ese\apache-tomcat-<version>\conf\server.xml) and enable the AJP port. By default, this port is enabled. For example, add or uncomment the following lines to enable AJP port 8009:

```
<!-- Define an AJP 1.3 Connector on port 8009 -->
<Connector port="8009" enableLookups="false"
redirectPort="8443" protocol="AJP/1.3" />
```

- 8 Edit the Tomcat configuration file (\Infor\ERPLN\ese\apache-tomcat-<version>\conf\server.xml) and add the JVM routing to each of the installed Tomcat instances.

The JVM routing is required for session affinity. To use session stickiness, you must set different `jvmRoute` attributes in the `Engine` element in Tomcat's `server.xml`. Furthermore, the names of the workers which are managed by the balancer must match those of the `jvmRoute` of the Tomcat instance with which they connect.

For the first Tomcat instance, the worker's name is **worker1**. For the second Tomcat instance, the worker's name is **worker2**. The workers' names must match the names in the Apache load balancer configuration.

For example, replace these lines in the first Tomcat installation:

```
<!-- Define the top level container in our container hierarchy -->  
<Engine name="Catalina" defaultHost="localhost">
```

With these lines:

```
<!-- Set jvmRoute to support load-balancing via AJP -->  
<Engine name="Catalina" defaultHost="localhost"  
  jvmRoute="worker1">
```

- 9 Restart the Tomcat and Apache services.
- 10 Refer to the post-installation steps later in this chapter.

Post installation

After you install a scenario, complete the post-installation steps.

Optionally, configure additional Infor LN environments by using the Infor Web UI administration console (<http://hostname/webui/servlet/admin>).

- 11 Synchronize the configured Infor LN environments between the Tomcat servers. Configured Infor LN environments are stored in the following file:
 \Infor\ERPLN\ese\Webtop\config\WebtopProperties.xml
 Synchronize this file with all the Tomcat servers.
- 12 Synchronize user context files between the Tomcat servers.
- 13 To ensure that the latest user context is available on all the Tomcat servers, use a tool to synchronize the contents of this directory between all the Tomcat servers in the following cluster:
 \Infor\ERPLN\ese\Webtop\config\usercontexts\
 The synchronization must be a continuous process, because user context files frequently change.

Available Infor Web UI links

The following table shows an overview of the Infor Web UI links:

Description	Link
Infor Web UI login	http://hostname/webui/servlet/login
Infor Web UI administration console	http://hostname/webui/servlet/admin
Individual Tomcat server Web UI login	http://hostname:8312/webui/servlet/login
Individual Tomcat server Web UI administration console	http://hostname:8312/webui/servlet/admin

Tomcat administration console

To view the status of the individual Tomcat servers, use the login tomcat password **tomcat**:
`http://<Tomcat server host>:8312/manager/html`

Load balancer status console

The load balancer status console is available in the vertical clustered configuration. Use the load balancer status console to view the status of all the workers in the configuration: `http://<Apache server host>/jkmanager/`