

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

---

# Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

---

---

**Autor(en):**  
Jochen van Waasen

**Version:** 1.0                      **Datum:** 2003-10-15

---

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

### Dokumenten Änderung

Version <sup>1</sup>	Datum	Text	Autor
1.0	2003-08-22	Draft	Jochen van Waasen

---

<sup>1</sup> Entwürfe haben Versionsnummer 0.x.

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## Inhalt

<b>1</b>	<b>Abstract</b>	<b>4</b>
<b>2</b>	<b>Voraussetzungen</b>	<b>4</b>
<b>3</b>	<b>Installation Apache 2</b>	<b>5</b>
<b>4</b>	<b>Installation mod_jk2.dll</b>	<b>5</b>
<b>5</b>	<b>Installation Tomcat 4</b>	<b>6</b>
<b>6</b>	<b>Tomcat Instanzen anlegen</b>	<b>7</b>
<b>7</b>	<b>Tomcat Instanzen konfigurieren</b>	<b>8</b>
<b>8</b>	<b>Apache konfigurieren</b>	<b>9</b>
<b>9</b>	<b>Tomcat Instanzen starten</b>	<b>11</b>
<b>10</b>	<b>Apache neu starten</b>	<b>13</b>
<b>11</b>	<b>Test: Loadbalancing</b>	<b>15</b>
<b>12</b>	<b>Anmerkungen: mod_jk2</b>	<b>16</b>
<b>13</b>	<b>Konfigurationsmöglichkeiten: Loadbalancing</b>	<b>17</b>
<b>14</b>	<b>Anmerkungen: Session</b>	<b>21</b>

**Kommentare und Hinweise bitte an:**

**Jochen van Waasen**

[jvw@vawacon.de](mailto:jvw@vawacon.de)

<http://www.vawacon.de>

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 1 Abstract

Dieser Text beschreibt die Installation eines Tomcat Clusters und die Integration mit Apache unter Verwendung der Loadbalancing Möglichkeiten des mod\_jk2 Moduls. In diesem Text wird die Installation von zwei Tomcat Instanzen und einer Apache Instanz auf *einem* Computer beschrieben.

Diese Anleitung wurde getestet unter Windows 2000 mit Apache 2.0.47 und Tomcat 4.1.27.

## 2 Voraussetzungen

Korrekt installiertes **J2SE SDK** mit der Umgebungsvariablen **JAVA\_HOME**.

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

### 3 Installation Apache 2

1. Laden Sie die ausführbare Windowsdatei **apache\_2.0.47-win32-x86-no\_ssl.exe** herunter.
2. Installation von Apache 2 durch Doppelklick auf die heruntergeladene Datei. Folgen Sie den Installationsanweisungen.

### 4 Installation mod\_jk2.dll

*Sie müssen die korrekte Version der mod\_jk2.dll für die verwendete Apache 2 Version installieren.*

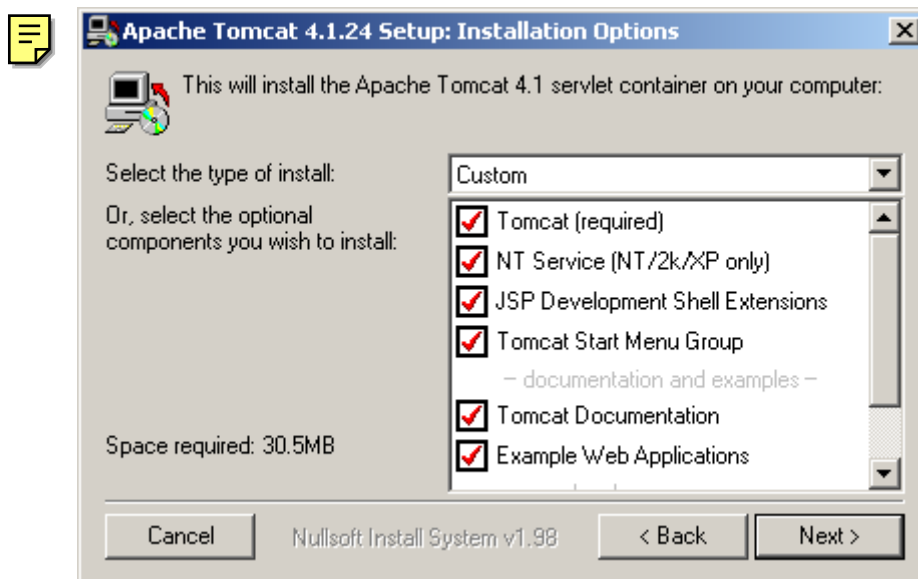
In diesem Beispiel wird Apache 2.0.47 verwendet!

1. Die Datei **mod\_jk2\_2.0.47.dll** muss in das Apache 2 Installationsverzeichnis **\modules** kopiert werden.
2. Jetzt kann Apache 2 konfiguriert werden um Tomcat 4 mittels **mod\_jk2\_2.0.47.dll** zu integrieren.

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 5 Installation Tomcat 4

1. Laden Sie die ausführbare Windowsdatei **tomcat-4.1.27.exe** herunter.
2. Installation von Tomcat 4 durch Doppelklick auf die heruntergeladene Datei. Folgen Sie den Installationsanweisungen.
- 2a. Als **Installation Options** ist **Custom** auszuwählen und **NT Service (NT/2K/XP only)** zu selektieren:



Tomcat wird als **NT Service** installiert!

*Mit dieser Option wird ein NT Dienst in der Windows Registry eingerichtet und der Dienst wird nach der vollständigen Installation gestartet. Der gestartete Dienst benutzt die Instanz und das Web Application **webapps** Verzeichnis im Tomcat Installationsverzeichnis welches während der Installation festgelegt wurde.*

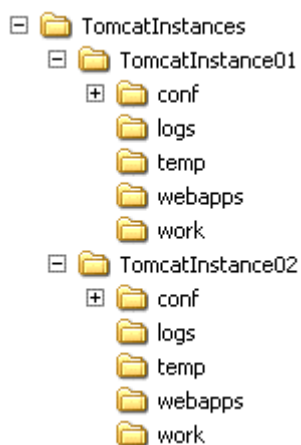
*Um den Dienst zu erzeugen und zu starten wurde die Datei **tomcat.exe** in das bin Verzeichnis des Tomcat Installationsverzeichnisses installiert. Diese Datei kann auch dazu benutzt werden NT Dienste manuell einzurichten und wird nur installiert, wenn man die NT Service Option während der Installation wählt.*

- 2b. Folgen Sie den nächsten Installationsanweisungen.
3. Setzen Sie die Umgebungsvariable **CATALINA\_HOME** auf das Tomcat 4.1 Installationsverzeichnis.

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 6 Tomcat Instanzen anlegen

1. Legen Sie ein neues Verzeichnis für Ihre Tomcat Instanzen an:  
*mkdir TomcatInstances*
2. Wechseln Sie in das neu angelegte Verzeichnis:  
*cd TomcatInstances*
3. Legen Sie neue Verzeichnisse für Ihre Tomcat Instanzen an:  
*mkdir TomcatInstance01, TomcatInstance02*
4. Legen Sie folgende Verzeichnisse in den Tomcat Instanz Verzeichnissen **TomcatInstance01** und **TomcatInstance02** an: **logs, temp, webapps, work**  
*mkdir logs temp webapps work*
5. Kopieren Sie das **conf** Verzeichnis aus dem Tomcat 4.1 Installationsverzeichnis in die Tomcat Instanz Verzeichnisse **TomcatInstance01** und **TomcatInstance02**.
6. **So sollte die Verzeichnisstruktur anschließend aussehen!**



Bis auf das Verzeichnis **conf** sind alle Verzeichnisse leer! Sie können weitere Instanzen auf die gleiche Art und Weise anlegen. Auch auf verschiedenen Laufwerken und anderen Computern im Netzwerk!

Dieser Text beschreibt die Installation von zwei Tomcat Instanzen auf einem Computer.

Es ist nicht explizit notwendig ein Verzeichnis für die Tomcat Instanzen (*TomcatApplication*) anzulegen. Die Tomcat Instanzen Verzeichnisse können auch in einer anderen Struktur untergebracht werden!

## 7 Tomcat Instanzen konfigurieren

Im **conf** Verzeichnis der Tomcat Instanzen Verzeichnisse befindet sich die Datei server.xml. Mit dieser Datei werden die Einstellungen für die Tomcat Instanzen konfiguriert.

*Dieser Text beschreibt die Installation von zwei Tomcat Instanzen auf einem Computer. Die Ports müssen daher für jede Instanz unterschiedlich sein. Da diese Konfigurationsdateien aus dem Tomcat Installationsverzeichnis kopiert worden sind, werden die Ports bereits von der Tomcat Installation verwendet die als Service gestartet wird!*

Die Ports müssen also in beiden server.xml Dateien geändert werden:

```
<!-- Example Server Configuration File -->
[...]  
<Server *port="8005" shutdown="SHUTDOWN" debug="0">
```

In **TomcatInstance01/conf/server.xml**\* ändern zu: port="11001".  
In **TomcatInstance02/conf/server.xml**\* ändern zu: port="22001".

```
[...]  
<!-- Define the Tomcat Stand-Alone Service -->  
<Service *name="Tomcat-Standalone" >
```

Sie können den Namen für die Instanz ändern (optional):

In **TomcatInstance01/conf/server.xml**\* ändern zu: name="TI01".  
In **TomcatInstance02/conf/server.xml**\* ändern zu: name="TI02".

```
[...]  
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on port 8081 -->  
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"  
*port="8080" minProcessors="5" maxProcessors="75" enableLookups="true"  
*redirectPort="8443" acceptCount="100" debug="0" connectionTimeout="20000"  
useURIVValidationHack="false" disableUploadTimeout="true" />
```

In **TomcatInstance01/conf/server.xml**\* ändern zu: port="11002".  
In **TomcatInstance01/conf/server.xml**\* ändern zu: redirectPort="11003".  
In **TomcatInstance02/conf/server.xml**\* ändern zu: port="22002".  
In **TomcatInstance02/conf/server.xml**\* ändern zu: redirectPort="22003".

```
[...]  
<!-- Define a Coyote/JK2 AJP 1.3 Connector on port 8009 --> <Connector  
className="org.apache.coyote.tomcat4.CoyoteConnector" *port="8009"  
minProcessors="5" maxProcessors="75" enableLookups="true"  
*redirectPort="8443" acceptCount="10" debug="0" connectionTimeout="20000"  
useURIVValidationHack="false"  
protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler"/>
```

In **TomcatInstance01/conf/server.xml**\* ändern zu: port="11004".  
In **TomcatInstance01/conf/server.xml**\* ändern zu: redirectPort="11003".  
In **TomcatInstance02/conf/server.xml**\* ändern zu: port="22004".  
In **TomcatInstance02/conf/server.xml**\* ändern zu: redirectPort="22003".



Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

```
[...]  
<!-- Define the top level container in our container hierarchy -->  
<Engine name="Standalone" defaultHost="localhost" debug="0" * * >
```

In **TomcatInstance01/conf/server.xml**\* hinzufügen: `jvmRoute="tomcat_01"`.  
In **TomcatInstance02/conf/server.xml**\* hinzufügen: `jvmRoute="tomcat_02"`.

*Dieser Eintrag ist notwendig für das Loadbalancing. Apache unterscheidet anhand dieses Eintrags die Tomcat Instanzen.*

```
[...]
```

## 8 Apache konfigurieren

1. Modifizieren Sie die **httpd.conf** Datei im **conf** Verzeichnis des Apache 2 Installationsverzeichnisses.

Fügen Sie am Ende des Dynamic Shared Object (DSO) Teils folgendes\* ein:

httpd.conf

```
[...]  
#  
# Dynamic Shared Object (DSO) Support  
#  
[...]  
LoadModule userdir_module modules/mod_userdir.so  
#LoadModule usertrack_module modules/mod_usertrack.so  
#LoadModule vhost_alias_module modules/mod_vhost_alias.so  
#LoadModule ssl_module modules/mod_ssl.so
```

```
*LoadModule jk2_module modules/mod_jk2_2.0.47.dll
```

```
[...]
```

2. Erzeugen Sie eine neue Textdatei mit dem Namen **workers2.properties** im **conf** Verzeichnis des Apache 2 Installationsverzeichnisses.

*Standardmäßig wird Apache 2 ohne diese Datei installiert und auch Tomcat 4 stellt diese Datei mit der Installationsversion nicht zur Verfügung!*

```
#--- workers2.properties
```

```
# Define the TCP socket communication channel for the Tomcat Instance 1  
[channel.socket:localhost:11004]  
info=Ajp13 forwarding over a TCP socket.  
tomcatId=tomcat_01  
# tomcatId corresponds to jvmRoute in Tomcat instance server.xml file!  
debug=0
```

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

```
lb_factor=1
```

```
# Define the TCP socket communication channel for the Tomcat Instance 2  
[channel.socket:localhost:22004]
```

```
info=Ajp13 forwarding over a TCP socket.
```

```
tomcatId=tomcat_02
```

```
# tomcatId corresponds to jvmRoute in Tomcat instance server.xml file!
```

```
debug=0
```

```
lb_factor=1
```

```
[status:]
```

```
info=Status worker, displays runtime information.
```

```
[uri:/jkstatus/*]
```

```
info=Displays status information and checks the config file for changes.
```

```
group=status:
```

```
# Map the Tomcat "examples" webapp to the webserver uri space
```

```
[uri:/examples/*]
```

```
info=Map the entire "examples" webapp.
```

```
# Configure the shared memory file
```

```
[shm]
```

```
file=D:/programme/apachegroup/apache2/logs/shm.file*
```

```
# Adjust this path to your environment !!!
```

```
size=1048576
```

```
debug=0
```

```
#---- end of workers2.properties
```

**Der Pfad\* für die shm Datei muss entsprechend angepasst werden!!**

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 9 Tomcat Instanzen starten

*Es sollten immer zuerst die Tomcat Instanzen gestartet werden und anschließend die Apache Instanz.*

*Falls eine oder mehrere Tomcat Instanzen neu gestartet worden sind, sollte auch die Apache Instanz neu gestartet werden.*

1. Wechseln Sie in das **bin** Verzeichnis des Tomcat Installationsverzeichnisses.
2. Überprüfen Sie die Umgebungsvariable **CATALINA\_HOME**. Diese muss auf das Tomcat Installationsverzeichnis gesetzt sein.
3. Setzen Sie die Umgebungsvariable **CATALINA\_BASE**. Diese muss auf das Tomcat Instanzenverzeichnis **TomcatInstance01** gesetzt werden.

```
set CATALINA_BASE=D:\TomcatInstances\TomcatInstance01
```

4. **Starten** Sie die Tomcat Instanz **TomcatInstance01**.

*startup*

```
Oct 14, 2003 9:45:19 PM org.apache.commons.modeler.Registry loadRegistry
INFO: Loading registry information
Oct 14, 2003 9:45:19 PM org.apache.commons.modeler.Registry getRegistry
INFO: Creating new Registry instance
Oct 14, 2003 9:45:20 PM org.apache.commons.modeler.Registry getServer
INFO: Creating MBeanServer
StandardEngine[Standalone]: setJvmRoute=tomcat_01
Oct 14, 2003 9:45:26 PM org.apache.coyote.http11.Http11Protocol init
INFO: Initializing Coyote HTTP/1.1 on port 11002
Starting service TI01
Apache Tomcat/4.1.27
Oct 14, 2003 9:45:28 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on port 11002
Oct 14, 2003 9:45:28 PM org.apache.jk.common.ChannelSocket init
INFO: JK2: ajp13 listening on /0.0.0.0:11004
Oct 14, 2003 9:45:28 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/111
config=D:\TomcatInstances\TomcatInstance01\conf\jk2.properties
```

Falls kein Portkonflikt vorliegt, läuft jetzt die **TomcatInstance01**.

Bei einer Fehlermeldung sollten Sie die verwendeten Ports überprüfen, diese dürfen nicht bereits belegt sein!

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

5. Setzen Sie die Umgebungsvariabel **CATALINA\_BASE**. Diese muss auf das Tomcat Instanzenverzeichnis **TomcatInstance02** gesetzt werden.

```
set CATALINA_BASE=D:\TomcatInstances\TomcatInstance02
```

6. **Starten** Sie die Tomcat Instanz **TomcatInstance02**.

```
startup
```

```
Oct 14, 2003 9:45:40 PM org.apache.commons.modeler.Registry loadRegistry
INFO: Loading registry information
Oct 14, 2003 9:45:40 PM org.apache.commons.modeler.Registry getRegistry
INFO: Creating new Registry instance
Oct 14, 2003 9:45:41 PM org.apache.commons.modeler.Registry getServer
INFO: Creating MBeanServer
StandardEngine[Standalone]: setJvmRoute=tomcat_02
Oct 14, 2003 9:45:46 PM org.apache.coyote.http11.Http11Protocol init
INFO: Initializing Coyote HTTP/1.1 on port 22002
Starting service TI02
Apache Tomcat/4.1.27
Oct 14, 2003 9:45:48 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on port 22002
Oct 14, 2003 9:45:48 PM org.apache.jk.common.ChannelSocket init
INFO: JK2: ajp13 listening on /0.0.0.0:22004
Oct 14, 2003 9:45:48 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/50
config=D:\TomcatInstances\TomcatInstance02\conf
\jk2.properties
```

Falls kein Portkonflikt vorliegt, läuft jetzt die **TomcatInstance02**.

Bei einer Fehlermeldung sollten Sie die verwendeten Ports überprüfen, diese dürfen nicht bereits belegt sein!

Es sollten jetzt beide Tomcat Instanzen auf dem System laufen. Falls Fehler aufgetreten sind, überprüfen Sie, ob Sie alle Schritte korrekt durchgeführt haben.

Die Tomcat Instanzen können auch als **Dienst** eingerichtet werden!

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 10 Apache neu starten

*Es sollten immer zuerst die Tomcat Instanzen gestartet werden und anschließend die Apache Instanz.*

*Falls eine oder mehrere Tomcat Instanzen neu gestartet worden sind, sollte auch Apache neu gestartet werden.*

1. Apache wurde als NT Dienst installiert, es gibt zwei Möglichkeiten Apache neu zu starten:

*Alternative 1:*

Neustart des Apache Dienstes mit dem Apache Service Monitor.

*Alternative 2:*

Wählen Sie in der Windows Taskleiste:

**Start > Einstellungen > Systemsteuerung > Verwaltung > Dienste**

Den Apache 2 Dienst auswählen und den Dienst neu starten.

*Falls Apache nicht korrekt startet überprüfen Sie die Installation der mod\_jk2.dll und den Eintrag in der httpd.conf Datei!*

2. Um die Konfiguration des mod\_jk2 Moduls und die Integration von Tomcat zu überprüfen rufen Sie im Browser folgende URL auf:

**http://localhost/jkstatus**

**Ergebnis:**

Status information for child 0										
<b>ajp13 runtime info</b>										
ajp13 information, using getAttribute()										
id	name	lb_factor	lb_value	route	errorState	graceful	epCount	errorTime		
0	localhost:11004	1	1	tomcat_01	N	N	0	0		
1	localhost:22004	1	1	tomcat_02	N	N	0	0		
<b>Scoreboard info (ver=0 slots=1)</b>										
Totals:										
Req	Err	Max	Avg							
0	0	0	-							
<b>Endpoint info ( no shm )</b>										
Worker	Req	Err	LastReq	ConnectionTime	TotalTime	MaxTime	AvgTime	ReqStart	+jk	+end

# Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

Totals:

Req	Err	Max	Avg
0	0	0	-

**uri runtime info**

uri information, using getAttribute()

id	name	host	uri	group	context
0			null	null	null
0	/jkstatus/*	*	/jkstatus/*	status:	/
0	/examples/*	*	/examples/*	lb:lb	/
0	*	*	null	null	null
0	*/	*	/	lb:lb	/

**Configured Properties**

Original data set by user

Object name	Property	Value
config:	file	\${serverRoot}/conf/workers2.properties
<b>shm:</b>	file	D:/programme/apache2/apache2/logs/shm.file
	size	1048576
	debug	0
<b>channel.socket:localhost:11004</b>	info	Ajp13 forwarding over a TCP socket.
	tomcatld	tomcat_01
	debug	0
	lb_factor	1
<b>channel.socket:localhost:22004</b>	info	Ajp13 forwarding over a TCP socket.
	tomcatld	tomcat_02
	debug	0
	lb_factor	1
<b>status:</b>	info	Status worker, displays runtime information.
<b>uri:/jkstatus/*</b>	info	Displays status information and checks the config file for changes.
	group	status:
<b>uri:/examples/*</b>	info	Map the entire "examples" webapp.

All settings ( automatic and configured ), after substitution

Name	Value
fs	\
ps	;
so	dll
arch	i386
serverRoot	D:/programme/ApacheGroup/Apache2
file	D:/programme/ApacheGroup/Apache2/conf/workers2.properties
channel.socket:localhost:11004.info	Ajp13 forwarding over a TCP socket.
channel.socket:localhost:11004.tomcatld	tomcat_01
channel.socket:localhost:11004.debug	0
channel.socket:localhost:11004.lb_factor	1
channel.socket:localhost:22004.info	Ajp13 forwarding over a TCP socket.
channel.socket:localhost:22004.tomcatld	tomcat_02
channel.socket:localhost:22004.debug	0
channel.socket:localhost:22004.lb_factor	1
status:.info	Status worker, displays runtime information.
uri:/jkstatus/*.info	Displays status information and checks the config file for changes.
uri:/jkstatus/*.group	status:
uri:/examples/*.info	Map the entire "examples" webapp.
shm:.file	D:/programme/apache2/apache2/logs/shm.file
shm:.size	1048576
shm:.debug	0

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 11 Test: Loadbalancing

*Um das Loadbalancing zu testen benötigen wir noch Inhalt der von den jeweiligen Tomcat Instanzen angezeigt werden soll. In den **workers2.properties** ist bereits die Web-Applikation **examples** konfiguriert, die noch mit Inhalt gefüllt werden muss.*

1. Legen Sie im Tomcat Instanz Verzeichnis **TomcatInstance01** folgendes Verzeichnis an:

### **examples**

2. Erzeugen Sie im examples Verzeichnis eine Textdatei mit dem Namen **instance.txt** und folgendem Inhalt:

*Tomcat Instance 01*

3. Legen Sie im Tomcat Instanz Verzeichnis **TomcatInstance02** folgendes Verzeichnis an:

### **examples**

4. Erzeugen Sie im examples Verzeichnis eine Textdatei mit dem Namen **instance.txt** und folgendem Inhalt:

*Tomcat Instance 02*

5. **Starten Sie die Tomcat Instanzen und Apache neu!**

6. Rufen Sie im Browser folgende URL auf:

**http://localhost/examples/instance.txt**

7. Aktualisieren Sie die Seite im Browser mehrmals. Es wird der jeweilige Text der Tomcat Instanz *Tomcat Instance 01* und *Tomcat Instance 02* angezeigt.

*Bei der gewählten Konfiguration werden die Tomcat Instanzen immer abwechselnd angesprochen!*

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 12 Anmerkungen: mod\_jk2

Das Modul **mod\_jk2** stellt nicht nur das Loadbalancing für die Tomcat Instanzen zur Verfügung, es ist auch in der Lage, Probleme mit den konfigurierten Tomcat Instanzen zu erkennen.

Die **failover** Eigenschaft des mod\_jk2 erkennt folgendes:

### **Ausfall einer Tomcat Instanz**

Falls eine Tomcat Instanz über den **connector service** nicht mehr erreichbar ist, wird die Instanz aus dem Pool der aktiven Instanzen entfernt. Das Modul verteilt automatisch die Requests auf die anderen Tomcat Instanzen.

### **Tomcat Instanz ist nach einem Ausfall wieder verfügbar**

Das Modul überprüft die **connector services** um festzustellen, ob eine ausgefallene Instanz wieder zur Verfügung steht. Falls eine solche Instanz wieder zur Verfügung steht, wird sie automatisch in den Pool der aktiven Instanzen aufgenommen und es werden wieder Requests an diese Instanz weitergeleitet.



Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

## 13 Konfigurationsmöglichkeiten: Loadbalancing

1. Die Verteilung der Requests auf die einzelnen Tomcat Instanzen kann über einen Parameter in der **workers2.properties** Datei konfiguriert werden.

### **workers2.properties**

[...]

# Define the TCP socket communication channel for the Tomcat Instance 1

[channel.socket:localhost:11004]

[...]

**lb\_factor=1**

[...]

Der Parameter **lb\_factor** kann ganzzahlige positive Werte inklusive 0 annehmen.

Dabei gilt:

- Je **niedriger** der Wert, desto **mehr Requests** werden an die Instanz geleitet.
- Je **höher** der Wert, desto **weniger Requests** werden an die Instanz geleitet.
- Bei einem Wert von **0**, werden **alle Requests** an diese Instanz geleitet. Erst wenn die Instanz nicht mehr verfügbar ist, werden die Requests an die anderen Instanzen weitergeleitet.

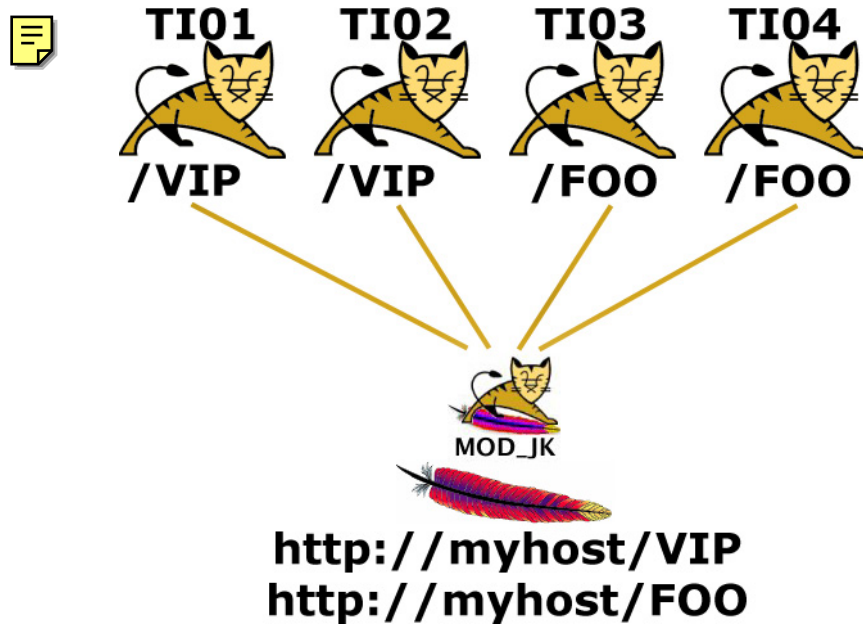
2. Es können auch explizite Loadbalancer in der **workers2.properties** Datei konfiguriert werden. Dies ist nur notwendig bei mehreren Clustern bzw. falls Anwendungen nicht auf allen Instanzen zur Verfügung stehen.

### **Szenario:**

Es werden vier Tomcat Instanzen **TI01**, **TI02**, **TI03** und **TI04** eingesetzt. Diese Instanzen sollen über einen Apache HTTP Server angesprochen werden. Die Tomcat Instanzen sind mit dem Modul mod\_jk2 integriert. Auf dem Tomcat Cluster laufen zwei Webapplikationen **VIP** und **FOO**. Diese sind über **http://myHost/VIP** und **http://myHost/FOO** im Browser aufrufbar.

Aus Performance- und Sicherheitsgründen laufen nicht alle Applikationen auf allen Instanzen. Auf den Instanzen **TI01** und **TI02** läuft die **VIP** Webapplikation und auf den Instanzen **TI03** und **TI04** läuft die **FOO** Webapplikation.

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?



### Problem: Standard Loadbalancer

Der Loadbalancer verteilt die Requests nach dem **round-robin** Prinzip. Dies entspricht der Standard-Konfiguration ohne explizite Loadbalancer.

### workers2.properties

[...]

```
# Define the TCP socket communication channel for the Tomcat Instance 1
[channel.socket:localhost:11004]
info=Ajp13 forwarding over a TCP socket.
tomcatId=tomcat_01
# tomcatId corresponds to jvmRoute in Tomcat instance server.xml file!
debug=0
lb_factor=1
```

**[Analog Tomcat Instance 2, Tomcat Instance 3, Tomcat Instance 4]**

[...]

```
# Map the Tomcat "VIP" webapp to the webserver uri space
[uri:/VIP/*]
info=Map the entire "VIP" webapp.
```

```
# Map the Tomcat "FOO" webapp to the webserver uri space
[uri:/FOO/*]
info=Map the entire "FOO" webapp.
```

[...]

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

Der Standard Loadbalancer verteilt die Requests bei der obigen Konfiguration automatisch nach dem **round-robin** Prinzip. Wir rufen im Browser **http://myHost/VIP** auf und der Request wird an die Tomcat Instanz **TI01** weitergeleitet. Auf **TI01** läuft die **VIP** Webapplikation und der Aufruf wird korrekt verarbeitet.

Anschließend rufen wir im Browser **http://myHost/FOO**. Nach dem **round-robin** Prinzip wird dieser Request an die Tomcat Instanz **TI02** weitergeleitet. Auf dieser Instanz läuft aber die Webapplikation **VIP** und nicht **FOO**. Der Aufruf wird von Tomcat mit einer entsprechenden Fehlermeldung quittiert. Wird jetzt nochmals **http://myHost/FOO** aufgerufen, wird dieser Request an die Tomcat Instanz **TI03** weitergeleitet. Auf **TI03** läuft die **FOO** Webapplikation und der Aufruf wird korrekt verarbeitet. Der nächste Request geht an **TI04** und dann an **TI01** usw.

Mit dem Standard Loadbalancer werden die Requests nicht zielgerichtet auf die entsprechenden Instanzen verteilt. Eine Einstellung über den Parameter **lb\_factor** hilft hier auch nicht weiter. Es wird nur die Quantität der Requests bestimmt.

Falls über die Einstellung mittels **lb\_factor** die Mehrzahl der Requests an **TI01** weitergeleitet werden, werden beim Aufruf von **http://myHost/FOO** noch mehr Fehler produziert, da jetzt häufiger **TI01** mit der **VIP** Applikation angesprochen wird als bei einer gleichmäßigen Verteilung der Requests.

### **Lösung: Explizite Loadbalancer**

Der Standard Loadbalancer verteilt die Requests gleichmäßig auf alle vier Instanzen nach dem **round-robin** Prinzip.

#### **workers2.properties**

Zur Lösung dieses Problems müssen zwei explizite Loadbalancer, **[lb:lb\_VIP]** und **[lb:lb\_FOO]**, konfiguriert und den URIs **[uri:/VIP/\*]** und **[uri:/FOO/\*]** entsprechend zugeordnet werden. Die expliziten Loadbalancer funktionieren auch nach dem **round-robin** Prinzip, verteilen die Requests aber innerhalb einer festgelegten Gruppe. Ein Eintrag, **group=lb\_VIP** und **group=lb\_FOO**, in der **connector service** Konfiguration der Instanzen bestimmt, welche Instanzen den Loadbalancern zugeordnet sind.

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

```
#---- workers2.properties

# Define the TCP socket communication channel for Tomcat Instance 01
[channel.socket:localhost:11004]
info=Ajp13 forwarding over a TCP socket
tomcatId=tomcat_01
debug=0
group=lb_VIP
lb_factor=1

# Define the TCP socket communication channel for Tomcat Instance 02
[channel.socket:localhost:22004]
info=Ajp13 forwarding over a TCP socket
tomcatId=tomcat_02
debug=0
group=lb_VIP
lb_factor=1

# Define the TCP socket communication channel for Tomcat Instance 03
[channel.socket:localhost:33004]
info=Ajp13 forwarding over a TCP socket
tomcatId=tomcat_03
debug=0
group=lb_FOO
lb_factor=1

# Define the TCP socket communication channel for Tomcat Instance 04
[channel.socket:localhost:44004]
info=Ajp13 forwarding over a TCP socket
tomcatId=tomcat_04
debug=0
group=lb_FOO
lb_factor=1

[status:]
info=Status worker, displays runtime information.

[lb:lb_VIP]
info=Loadbalancer for VIP application.
debug=0

[lb:lb_FOO]
info=Loadbalancer for FOO application.
debug=0

[uri:/jkstatus/*]
info=Displays status information and checks the config file for changes.
group=status:

[uri:/VIP/*]
group=lb_VIP

[uri:/FOO/*]
group=lb_FOO
```

Wie installiert und integriert man ein Tomcat Cluster mit Apache und Loadbalancing?

```
# Configure the shared memory file
[shm]
file=D:\SERVER\ApacheGroup\Apache2\logs\shm.file*
size=1048576
debug=0

#---- end of workers2.properties
```

**Der Pfad\* für die shm Datei muss entsprechend angepasst werden!!**

## 14 Anmerkungen: Session

Apache leitet HTTP Requests die für eine Tomcat Instanz bestimmt sind an das Modul **mod\_jk2**. Dieses Modul untersucht das Session Cookie. Der **jvmRoute** Parameter gibt Auskunft darüber, von welcher Tomcat Instanz die Session erzeugt wurde. Somit werden alle Requests die zu einer Session gehören immer an dieselbe Tomcat Instanz weitergeleitet.

Dieses Feature heißt bei Tomcat **sticky session**, ein anderer Begriff dafür ist **session affinity**. Über den Loadbalancer [**lb: ...**] in der **workers2.properties** Datei kann die **sticky session** konfiguriert werden. Standardmäßig ist **sticky session** aktiviert.

In verteilten (**distributable**) Anwendungen ist dies die einzige sinnvolle Einstellung! Ansonsten wechselt der Benutzer eventuell mitten in seiner Session auf eine Tomcat Instanz, auf der die Session unbekannt ist, und alle Session Daten sind verloren.

Trotz der **sticky session** ist ein Tomcat Cluster mit mehreren Instanzen nicht fehlertolerant (**fault tolerant**). Falls die Tomcat Instanz, welche die Session erzeugt hat ausfällt, wird der Benutzer automatisch an eine andere Tomcat Instanz weitergeleitet (**failover**). Die Session ist verloren, da die andere Tomcat Instanz die Session nicht kennt. Es findet keine **session replication**, also das Verteilen aller Sessions auf alle Tomcat Instanzen, statt.

Das Modul **mod\_jk2** unterstützt keine **session replication**!

Das Verteilen von Sessions auf alle Tomcat Instanzen muss durch den Einsatz weiterer Tools realisiert werden!