Внедрение широкого кластера Tomcat с поддержкой технологии Sso

Оглавление

[Внедрение широкого кластера Tomcat с поддержкой технологии Sso 1](#_Toc487794191)

[1. Термины и сокращения 3](#_Toc487794192)

[2. Введение 3](#_Toc487794193)

[3. Настройка широкого кластера 3](#_Toc487794194)

[3.1 Определение IP-адресов и задание TCP-портов инстансов 3](#_Toc487794195)

[3.2 Остановка серверов 3](#_Toc487794196)

[3.3 Подключение необходимых серверных библиотек 3](#_Toc487794197)

[3.4 Конфигурация серверов 3](#_Toc487794198)

[3.5 Запуск серверов 5](#_Toc487794199)

[3.5.1 Проверка корректности конфигурации серверов 5](#_Toc487794200)

[4. Настройка приложений SsoUi (графической оболочки для авторизации) 5](#_Toc487794201)

[4.1 Установка 5](#_Toc487794202)

[4.2 Добавление приложений SsoUi в конфигурацию Apache HTTPD 5](#_Toc487794203)

[4.3 Конфигурация web.xml 5](#_Toc487794204)

[4.4 Перезапуск серверов 6](#_Toc487794205)

[5. Установка обновлённых приложений 6](#_Toc487794206)

# Термины и сокращения

*Широкий кластер –* кластер Tomcat, инстансы которого установлены на разных физических машинах, в противовес *«узкому»* кластеру – с инстансами только на одной машине*.*

# Введение

Данная инструкция предназначена для настройки широкого кластера серверов Tomcat, в предположении, что у специалистов имеется богатый опыт настройки серверов Tomcat, Apache HTTPD, а также работы с данными технологиями и их администрирования.

# Настройка широкого кластера

## Определение IP-адресов и задание TCP-портов инстансов

Для каждого инстанса Tomcat определить IP-адрес машины и задать свободные TCP-порты таким образом, что пары <IP-адрес>:<TCP-порт> однозначно определяют инстанс в кластере.

Узнать IP-адрес машины можно консольной командой ipconfig; увидеть список занятых TCP-портов можно консольной командой netstat (столбец Local address) – свободные TCP-порты можно использовать для нужд кластера.

Рассмотрим пример с кластером из двух физических машин с IP-адресами 1.1.1.1 и 2.2.2.2 соответственно, на каждом из которых установлено по 2 инстанса Tomcat. Таким образом, будем конфигурировать кластер из четырёх инстансов.

Пара <IP-адрес>:<TCP-порт> должна однозначно определять инстанс томката во всём кластере, поэтому зададим TCP-порты инстансов следующим образом:

1. 1.1.1.1:4000
2. 1.1.1.1:4001
3. 2.2.2.2:4000
4. 2.2.2.2:4001

Значение TCP-порта 4000 предлагается Томкатом по умолчанию, поэтому будем использовать значения вблизи него (в предположении, что эти порты не заняты никакими другими сервисами компьютера).

## Остановка серверов

Остановить все настраиваемые инстансы Tomcat.

## Подключение необходимых серверных библиотек

В папку *lib* каждого инстанса скопировать библиотеки:

1. catalina-jepria.jar (из папки *lib* проекта CatalinaJepria)
2. sso-utils.jar (из бинарного репозитория, в директории *<BIN\_HOME>/org/jepria/ssoutils*)

Их загрузка произойдёт при очередном перезапуске/запуске сервера.

## Конфигурация серверов

На каждом инстансе, в файле *<TOMCAT>/conf/server.xml* указать соответствующую конфигурацию кластера путём добавления следующего текста внутрь тега Server/Service/Engine/Host (удалив предыдущую конфигурацию кластера, если имелась):

<Cluster className="org.apache.catalina.ha.tcp.SimpleTcpCluster" channelSendOptions="8">

<Manager className="org.apache.catalina.ha.session.DeltaManager"

expireSessionsOnShutdown="false" notifyListenersOnReplication="true"/>

<Channel className="org.apache.catalina.tribes.group.GroupChannel">

<Membership className="org.apache.catalina.tribes.membership.McastService"  
 address="228.0.0.4" port="45566" frequency="500" dropTime="3000"/>

<Sender className="org.apache.catalina.tribes.transport.ReplicationTransmitter">

<Transport className="org.apache.catalina.tribes.transport.nio.PooledParallelSender"/>

</Sender>

<!-- SELF -->

<Receiver className="org.apache.catalina.tribes.transport.nio.NioReceiver"

**address="1.1.1.1" port="4000"**  
 autoBind="100" selectorTimeout="5000" maxThreads="6" />

<Interceptor className="org.apache.catalina.tribes.group.interceptors.TcpFailureDetector"/>

<!-- ANOTHER MEMBER -->

<Interceptor className="org.apache.catalina.tribes.group.interceptors.StaticMembershipInterceptor">

<Member className="org.apache.catalina.tribes.membership.StaticMember"

**host="1.1.1.1" port="4001"**

securePort="-1" domain="staging-cluster"

**uniqueId="{0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,5}**"/>

</Interceptor>

<!-- ANOTHER MEMBER -->

<Interceptor className="org.apache.catalina.tribes.group.interceptors.StaticMembershipInterceptor">

<Member className="org.apache.catalina.tribes.membership.StaticMember"

**host="2.2.2.2" port="4000"**

securePort="-1" domain="staging-cluster"

**uniqueId="{0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,6}**"/>

</Interceptor>

<!-- ANOTHER MEMBER -->

<Interceptor className="org.apache.catalina.tribes.group.interceptors.StaticMembershipInterceptor">

<Member className="org.apache.catalina.tribes.membership.StaticMember"

**host="2.2.2.2" port="4001"**

securePort="-1" domain="staging-cluster"

**uniqueId="{0,1,2,3,4,5,6,7,8,9,0,1,2,3,4,7}**"/>

</Interceptor>

</Channel>

<Valve className="org.apache.catalina.ha.tcp.ReplicationValve" filter=""/>

<Valve className="org.apache.catalina.ha.session.JvmRouteBinderValve"/>

<ClusterListener className="org.apache.catalina.ha.session.JvmRouteSessionIDBinderListener"/>

<ClusterListener className="org.apache.catalina.ha.session.ClusterSessionListener"/>

</Cluster>

Необходимо обратить внимание на выделенные элементы: Тег Receiver конфигурирует данный инстанс, теги Interceptor – другие инстансы кластера. Значения параметров *host* и *port* подставляются из определений выше. Параметр *uniqueId* – произвольный набор из 16 цифр, перечисленных через запятую, уникальный для каждого Interceptor’а, входящего в данный конфигурационный файл (в примере наборы отличаются последней цифорй).

Прочие (не выделенные) параметры приведенного фрагмента в настройке не нуждаются.

Аналогично, для поддержания механизма единого входа, общего на весь кластер, необходимо в этой же секции Host добавить строку

<Host name="hostname" ... >

**<Valve className="org.apache.catalina.ha.authenticator.ClusterSingleSignOn"/>**

</Host>

(либо заменить этой строкой указанный прежде Valve Sso).

IP-адрес 228.0.0.4, используемый для multicast-сервиса, зарезервирован в Интернете специально для этих целей (см. [RFC 3171](https://tools.ietf.org/html/rfc3171), раздел 2).

## Запуск серверов

Запустить все настраиваемые инстансы Tomcat.

### Проверка корректности конфигурации серверов

На данном этапе (когда в *server.xml* каждого инстанса томката внесены указанные выше изменения), после запуска и загрузки всех инстансов, нужно на каждом инстансе проанализировать лог-файл с именем *catalina.DATE.log* на наличие следующих записей (можно искать контекстным поиском по фрагменту, отмеченному **жирным**):

INFO: **Receiver Server Socket bound to**:/1.1.1.1:4000

Запись свидетельствует о том, что данный инстанс успешно привязан к определённым нами IP-адресу и TCP-порту.

INFO: **Replication member added**:org.apache.catalina.tribes.membership.StaticMember[tcp://1.1.1.1:4001,…

INFO: **Replication member added**:org.apache.catalina.tribes.membership.StaticMember[tcp://2.2.2.2:4000,…

INFO: **Replication member added**:org.apache.catalina.tribes.membership.StaticMember[tcp://2.2.2.2:4001,…

Записи свидетельствуют о том, что успешно «подцепились» данным инстансом другие инстансы, с указанными IP-адресами и TCP-портами.

SEVERE: Unable to send AbstractReplicatedMap.ping message

org.apache.catalina.tribes.ChannelException: Send failed, attempt:2 max:1; **Faulty members**:tcp://2.2.2.2:4000;

Запись свидетельствует о том, что инстанс с указанным адресом отключился от кластера (либо неверно настроен, а поэтому не смог быть подключенным). Однако это не следует воспринимать как ошибку при перезагрузке одного из инстансов в кластере (в этом случае его отключение ожидаемо), либо если какой-либо инстанс еще не успел загрузиться при одновременном старте.

# Настройка приложений SsoUi (графической оболочки для авторизации)

## Установка

На каждый из настраиваемых инстансов установить стандартным образом (командой ant tomcat.deploy) одну и ту же версию приложения SsoUi, но под уникальным для каждого сервера контекстным именем (контекстное имя указывается дополнительным параметром в команде установки, например: ant tomcat.deploy ... -DCONTEXT\_PATH=/SsoUi\_01). Приложение SsoUi требует обязательного явного указания данного параметра.

## Добавление приложений SsoUi в конфигурацию Apache HTTPD

Стандартным образом добавить записи обо всех установленных приложениях SsoUi в файл *mod\_jk.conf.*

Перезапустить Apache HTTPD, чтобы изменения вступили в силу.

## Конфигурация web.xml

В серверном файле *conf/web.xml* каждого инстанса кластера добавить определение параметра *ssoUiContext* в тело тега <web-app> :

<context-param>

<param-name>ssoUiContext</param-name>

<param-value>/SsoUi\_01</param-value>

</context-param>

где /SsoUi\_01 – уникальное в кластере контекстное имя приложения SsoUi, под которым оно загружено на данный инстанс.

## Перезапуск серверов

Для того, чтобы изменения вступили в силу, необходимо перезапустить все инстансы.

# Установка обновлённых приложений

Обновить все имеющиеся на инстансах приложения новыми сборками, поддерживающими автоматическое перенаправление на соответствующее серверу приложение SsoUi, используя внесённый в *web.xml* серверов параметр (логика реализована на системном уровне в файле *security/Login.jsp* при сборке на Jepria-10.2.1 и выше) и содержащими обновленный файл *META-INF/context.xml*