

Apache Tomcat

Administrator's Guide

Release 5.0

Table of Contents

| | |
|-----------------------------------|-----|
| Preface | iii |
| 1. Installation | 1 |
| 1.1. Download | 1 |
| 1.2. Install and build APR | 1 |
| 1.3. Configure | 2 |
| 1.4. Automatic startup | 3 |
| 2. Upgrade | 6 |
| 2.1. Download and Build APR | 6 |
| 2.2. Upgrade | 6 |
| 3. Downgrade | 7 |
| 3.1. Downgrade | 7 |

Preface

Apache Tomcat is the servlet container that is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies. The Java Servlet and JavaServer Pages specifications are developed by Sun under the Java Community Process. Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations

The Apache Tomcat web site [<http://tomcat.apache.org>] carries details on the latest release and other information to make your work or play with PostgreSQL more productive. And The mailing lists are a good place to have your questions answered, to share experiences with other users, and to contact the developers.

1

Installation

1.1. Download

Download and save Apache Tomcat binary archive from <http://tomcat.apache.org>

```
cd /usr/local
md5sum -cv apache-tomcat-5.XX.YY.tar.gz.MD5
#http-2.XX.YY.tar.gz OK
apache-tomcat-5.XX.YY.tar.gz OK
```

1.2. Install and build APR

Install binaries and create required user and directories.

```
tar -xzf apache-tomcat-5.XX.YY.tar.gz
ln -s apache-tomcat-5.XX.YY tomcat
useradd -g 2100 -u 2102 -s /bin/bash -m -d /home/tomcat tomcat
mkdir /etc/tomcat
mv /usr/local/tomcat/conf /etc/tomcat
mkdir /var/log/tomcat
rm -rf /usr/local/tomcat/logs
ln -s /var/log/tomcat /usr/local/tomcat/logs
chown -R tomcat /var/log/tomcat
ln -s /etc/tomcat/conf /usr/local/tomcat/conf
chmod +r /etc/tomcat/conf/*
mkdir /usr/local/tomcat/work
```

We use Apache Portable Runtime to increase tomcat performance, you can download here [<http://apr.apache.org>].

```
md5sum apr-X.X.X.tar.gz
tar xzf apr-X.X.X.tar.gz
cd apr-1.2.7
./configure --prefix=/usr/local/apr-X.X.X
make
make install
ln -s /usr/local/apr-X.X.X /usr/local/apr

cd /usr/local/tomcat/bin/
tar xzf tomcat-native.tar.gz
cd tomcat-native-Y.Y.Y/jni/native
./configure --with-apr=/usr/local/apr --with-apxs=/usr/local/apache/bin/apxs --with-java-home=/usr/local/
make
make install
```

1.3. Configure

Edit and configure default `/etc/tomcat/conf/server.xml` file.

```
<Server port="8005" shutdown="SHUTDOWN">
  <Listener className="org.apache.catalina.core.AprLifecycleListener" />
  <Listener className="org.apache.catalina.mbeans.ServerLifecycleListener" />
  <Listener className="org.apache.catalina.mbeans.GlobalResourcesLifecycleListener" />
  <Listener className="org.apache.catalina.storeconfig.StoreConfigLifecycleListener"/>
  <GlobalNamingResources>
    <Resource name="UserDatabase" auth="Container"
      type="org.apache.catalina.UserDatabase"
      description="User database that can be updated and saved"
      factory="org.apache.catalina.users.MemoryUserDatabaseFactory"
      pathname="conf/tomcat-users.xml" />
  </GlobalNamingResources>
  <!-- Define the Tomcat Admin Service -->
  <Service name="CatalinaAdmin">
    <Connector port="8080" maxHttpHeaderSize="8192"
      maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
      enableLookups="false" redirectPort="8443" acceptCount="100"
      connectionTimeout="20000" disableUploadTimeout="true" />
    <Engine name="CatalinaAdmin" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
      <Host name="localhost" appBase="webapps"
        unpackWARs="true" autoDeploy="true"
        xmlValidation="false" xmlNamespaceAware="false">
        <Context
          docBase="/usr/local/tomcat/server/webapps/host-manager"
          path="/admin"
          privileged="true">
          <WatchedResource>/usr/local/tomcat/conf/context.xml</WatchedResource>
        </Context>
      </Host>
    </Engine>
  </Service>
  <!-- Define the Apache-Tomcat Service -->
  <Service name="Catalina">
    <Connector port="8009"
      enableLookups="false" redirectPort="8443" protocol="AJP/1.3"
      compression="on"
      compressionMinSize="2048"
      noCompressionUserAgents="gozilla, traviata"
      compressableMimeType="text/html,text/xml"
    />
    <Engine name="Catalina" defaultHost="localhost">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
    <Host
      appBase="/srv/XXXXXXXXXXXXXXXXXX/webapps"
      name="XXXXXXXXXXXXXXXXXX"
      autoDeploy="true"
      deployOnStartup="true"
      deployXML="true"
      unpackWARs="true"
      xmlValidation="false"
      xmlNamespaceAware="false">
      <Alias>XXXXXXXXXXXXXXXXXX</Alias>
      <Context
        docBase="/usr/local/tomcat/server/webapps/manager"
        path="/manager"

```

```

        privileged="true">
            <ResourceEnvRef
                name="users"
                type="org.apache.catalina.UserDatabase"/>
            <WatchedResource>/usr/local/tomcat/conf/context.xml</WatchedResource>
        </Context>
    </Host>
</Engine>
</Service>
</Server>

```

Add required DTDs in a /etc/tomcat/dtd directory dtds.tar.gz [libs/dtds.tar.gz]

Note

It's possible that an error occurred on XML parser version. You must add this library [libs/xercesImpl.jar] in /usr/local/tomcat/common/endorsed for solve compatibility.

Define environment variables in /home/tomcat/.bash_profile

```

JAVA_HOME=/usr/local/java
CATALINA_OPTS="-server -Xms512m -Xmx800m -XX:PermSize=128m -XX:MaxPermSize=256m -Ddtd.directory=/etc/tomcat/dtds"
JAVA_OPTS="-Dconfiguration.builder=fr.openmodel.configuration.JNDIHibernateConfigurationBuilder"
JAVA_OPTS="${JAVA_OPTS} -Dtemplate.engine.builder=fr.openmodel.template.VelocityTemplateEngineBuilder"
LANG=en_US.UTF-8
export JAVA_HOME CATALINA_OPTS JAVA_OPTS LANG
export LD_LIBRARY_PATH=/usr/local/apr/lib

umask 002

```

Edit /etc/tomcat/conf/tomcat-users.xml and configure access to tomcat manager :

```

<user name="<your-account-name>" password="<your-password>" roles="admin,manager" />

```

Add appropriate rights for tomcat user.

```

chown tomcat.wwwgroup -R /var/log/tomcat/
chown tomcat.wwwgroup -R /usr/local/tomcat/*
chown tomcat.wwwgroup -R /etc/tomcat

```

1.4. Automatic startup

Edit default startup script /etc/init.d/tomcat file.

```

#!/bin/sh
#
# tomcat Tomcat start/stop script
#
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
DAEMON=/usr/local/tomcat/bin/startup.sh
STOPDAEMON=/usr/local/tomcat/bin/shutdown.sh
NAME=Tomcat
DESC="Tomcat servlet server"
test -x $DAEMON || exit 0
# set -e
case "$1" in
    start)
        ps -U tomcat | grep -q java

```

```

if [ $? -eq 0 ] ; then
    echo "Tomcat is already running"
    echo "Please, stop it first"
    exit 1
else
    if [ "$2" = "rmlogs" ] ; then
        echo "Removing Tomcat logs"
        rm /usr/local/tomcat/logs/*
    fi
    echo "Remove bad files generated by tomcat (files contains hard reference to site version
    find /etc/tomcat/conf/ -name "mpng2-*.xml" -print -exec rm {} \;
    echo "Starting $DESC: $NAME"
    su - tomcat -c "${DAEMON}"
    echo "."
fi
;;
stop)
echo "Stopping $DESC: $NAME "
su - tomcat -c "${STOPDAEMON}"
sleep 10
ps -U tomcat | grep -q java
if [ $? -eq 0 ] ; then
    ps h -U tomcat | grep java | awk '{print $1}' |xargs kill
    sleep 2
    echo "Tomcat didn't stoped after 10 seconds, has been killed"
    sleep 2
    ps -U tomcat | grep -q java
    if [ $? -eq 0 ] ; then
        ps h -U tomcat | grep java | awk '{print $1}' |xargs kill -9
        sleep 2
        echo "WARNING !! Tomcat couldn't be killed, has been hardly killed (kill -9) !"
    fi
fi
if [ "$2" = "rmlogs" ] ; then
    echo "Removing Tomcat logs"
    rm /usr/local/tomcat/logs/*
fi
echo "Remove bad files generated by tomcat (files contains hard reference to site version number)
find /etc/tomcat/conf/ -name "mpng2-*.xml" -print -exec rm {} \;
echo "."
;;
restart|force-reload)
echo -n "Restarting $DESC: $NAME"
su - tomcat -c "${STOPDAEMON}"
sleep 10
ps -U tomcat | grep -q java
if [ $? -eq 0 ] ; then
    ps h -U tomcat | grep java | awk '{print $1}' |xargs kill
    sleep 2
    echo "Tomcat didn't stoped after 10 seconds, has been killed"
    sleep 2
    ps -U tomcat | grep -q java
    if [ $? -eq 0 ] ; then
        ps h -U tomcat | grep java | awk '{print $1}' |xargs kill -9
        sleep 2
        echo "WARNING !! Tomcat couldn't be killed, has been forcibly killed (kill -9)
    fi
fi
if [ "$2" = "rmlogs" ] ; then
    echo "Removing Tomcat logs"
    rm /usr/local/tomcat/logs/*
fi
echo "Remove bad files generated by tomcat (files contains hard reference to site version number)
find /etc/tomcat/conf/ -name "mpng2-*.xml" -print -exec rm {} \;
su - tomcat -c "${DAEMON}"
echo "."

```

```
;;
*)
# N=/etc/init.d/$NAME
N=$0
echo "Usage: $N {start|stop|restart|force-reload}" >&2
echo "An optionnal second parameter \"rmlogs\" can be specified to
remove Tomcat logs" >&2
exit 1
;;
esac
exit 0
```

Add execution right and register automatic startup.

```
chmod a+x /etc/init.d/tomcat
update-rc.d -f tomcat defaults
```

2

Upgrade

2.1. Download and Build APR

Same as installation section.

2.2. Upgrade

Upgrade old version to the new version.

```
/etc/init.d/tomcat stop
rm /usr/local/tomcat
ln -s /usr/local/apache-tomcat-5.XX.YY /usr/local/tomcat
cp -r /etc/tomcat/conf /etc/tomcat/conf.YYYYMMDD
rm -rf /usr/local/tomcat/logs
ln -s /var/log/tomcat /usr/local/tomcat/logs
rm -rf /usr/local/tomcat/conf
ln -s /etc/tomcat/conf /usr/local/tomcat/conf
/etc/init.d/tomcat start
```

3

Downgrade

3.1. Downgrade

Downgrade the current version (X.X.X) to the previous (Y.Y.Y).

```
/etc/init.d/tomcat stop
rm /usr/local/tomcat
ln -s /usr/local/apache-tomcat-Y.Y.Y /usr/local/tomcat
cp -r /etc/tomcat/conf.YYYMMDD /etc/tomcat/conf
/etc/init.d/tomcat start
```