

Linuxha.net
Step-by-step Guide:
Apache HTTP Server

Introduction

The purpose of this document is to describe suggested procedures for creating a clustered Apache HTTP Server configuration with Linuxha.net. These procedures were tested using the following configuration:

- Fedora Core 5
- DRBD version 0.7.20
- Linuxha.net version 1.2
- Apache version 2.2.2

The following conventions are used throughout this document:

# ls	Single-line command entered as root user.
# useradd \ > --home-dir /home2/xyz \ > --gid abc \ > xyz	Multi-line command entered as root user.
\$ ls	Single-line command entered as non-root user
\$ cut \ > --delimiter=":" \ > --fields=1,5 \ / etc/passwd	Multi-line command entered as non-root user.
127.0.0.1 localhost 192.168.1.32 fc5s1 192.168.1.33 fc5s2 192.168.100.32 fc5s1b 192.168.100.33 fc5s2b	Command output or file contents

User Account Creation

The Apache HTTP server processes will run with the **apache** user credentials. The **apache** user and group must exist on both nodes and have the same user and groups ids.

All commands in this section are executed as **root** on both nodes, unless otherwise indicated.

1. Check whether the **apache** user exists and that the user and group ids are the same on both nodes.

```
# grep apache /etc/passwd
```

If there is no result continue to step 2, since the **apache** user does not exist. Otherwise, compare the user and group ids on both nodes. These are the third and fourth fields respectively of the user account record as shown below:

```
apache:x:48:48:Apache:/var/www:/sbin/nologin
```

If the user and group ids are not the same on both nodes, proceed to step 6. Otherwise go to the next section.

2. Identify the next available group id.

```
# expr `cut --delimiter=":" --fields=3 /etc/group | \  
> grep --regexp="^[5-9][0-9][0-9]$" | sort --numeric-sort|tail -1` + 1
```

3. Create **apache** group and assign the larger of the group ids returned in step 2, e.g. 508.

```
# groupadd --gid 508 apache
```

4. Identify the next available user id.

```
# expr `cut --delimiter=":" --fields=3 /etc/passw | \  
> grep --regexp="^[5-9][0-9][0-9]$" | sort --numeric-sort|tail -1` + 1
```

5. Create **apache** user and assign the larger of the user ids returned in step 4, e.g. 508.

```
# useradd --home-dir /var/www -M \  
> --comment "Apache" --gid apache --uid 508 \  
> apache
```

Proceed to the next section.

6. If the **apache** group ids are different, change the group id on **one of the nodes** to be the same as the other, e.g. 49. Ensure that the new group id is not already in use.

```
# grep 49 /etc/group # check that group id 49 is not in use  
# groupmod -g 49 apache # change group id to 49
```

7. If the **apache** user ids are different, change the user id on **one of the nodes** to be the same as the other, e.g. 49. Ensure that the new user id is not already in use.

```
# grep 49 /etc/passwd # check that user id 49 is not in use  
# usermod --uid 49 --gid apache apache # change user id to 49
```

8. Change the user and group ownership of files owned by **apache**.

The following commands will find all files whose user / group ownership is 48 and change their user / group ownership to **apache**.

```
# find /usr -gid 48 -exec chgrp apache {} \  
# find /usr -uid 48 -exec chown apache {} \  
#
```

Installation

This section provides basic instructions for downloading and installing the Apache HTTP Server version 2.2 from i386 binary RPM or source. Instructions for installing or upgrading using other package formats are beyond the scope of this guide.

All steps are to be performed on both nodes as **root** unless otherwise indicated.

1. Check whether Apache HTTP Server is installed, by executing

```
# find /usr -name httpd -print
```

If no results are returned, proceed to step 2. Otherwise, take a note of where **httpd** is located and continue to step 4.

2. Download Apache HTTP Server.

Apache 2.2.2 can be installed from RPM (binary) or by compiling the source code.

To download the RPM, execute:

```
# wget \  
> http://www.mirror.ac.uk/mirror/ftp.apache.org/httpd/binaries/rpm/i386/httpd-  
2.2.2-1.i386.rpm
```

To download the source, execute

```
# cd /tmp  
# wget \  
> http://www.mirror.ac.uk/mirror/ftp.apache.org/httpd/httpd-2.2.2.tar.gz
```

3. Install Apache HTTP server.

RPM:

```
# rpm --hash --install httpd-2.2.2-1.i386.rpm
```

Source (default configuration):

```
# cd /usr/local/src
# tar xvzf /tmp/httpd-2.2.2.tar.gz
# cd httpd-2.2.2
# ./configure
# make
# make install
```

4. Delete the Apache HTTP Server startup and shutdown scripts.

```
# rm /etc/rc3.d/S??httpd /etc/rc0.d/K??httpd
```

5. Configure firewall

If the firewall is running, it needs to be configured to allow access to Apache HTTP Server port (default: port 80), by inserting the following into **/etc/syconfig/iptables**:

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
```

and restarting the firewall by executing

```
# /etc/init.d/iptables restart
```

Directories / Files for Replication

The location of Apache HTTP Server files depends upon whether it was installed from binary RPM or compiled from source.

Directory / File		Comment
<i>RPM Install</i>	<i>Source Install (default configuration)</i>	
/etc/httpd/conf	/usr/local/apache2/conf	Configuration files directory
/etc/httpd/conf.d		Configuration files directory
/var/www/html	/usr/local/apache2/htdocs	Web document root directory
/var/www/icons	/usr/local/apache2/icons	Icons directory
/var/www/error	/usr/local/apache2/error	HTTP Error document directory

Create Replicated File Systems

All commands are to be executed as **root** on both nodes. Steps 5 to 6 may be omitted on the secondary node.

1. Use **fdisk** to create a partition (*/dev/sda6*) large enough to accommodate all replicated web-sites.

2. Initialise the physical volume:

```
# pvcreate /dev/sda6
```

3. Create the volume group (*apachevg*):

```
# vgcreate apachevg /dev/sda6
```

4. Create mount point for replicated file system

```
# mkdir /apache
```

5. Create logical volume (*httplv*):

```
# lvcreate --size 512M --name httplv apachevg
```

6. Create file system on logical volume:

```
# mkfs -t ext3 /dev/apachevg/httplv
```

Populate Replicated File Systems

All commands are to be executed as **root** on the primary node.

RPM Installation

1. Shut down the Apache HTTP Server

```
# /usr/sbin/apachectl stop
```

2. Mount logical volume

```
# mount -t ext3 /dev/apachevg/httplv /apache
```

3. Create directory structure

```
# mkdir /apache/bin
# ln -s /usr/sbin/apachectl /apache/bin/apachectl
# ln -s /var/log/httpd /apache/logs
# ln -s /var/run /apache/run
# ln -s /usr/lib/httpd/modules /apache/modules
```

4. Copy web documents, icons and configuration files to **/apache** directory

```
# cd /etc/httpd
# find conf conf.d -print | cpio --pass-through /apache
# cd /var/www
# find error html icons -print | cpio --pass-through /apache
```

5. Edit **/apache/conf/httpd.conf** to reflect the new directory structure and configured to listen on the virtual IP address (*192.168.1.40*), as shown in the following table:

Source Installation

1. Shut down the Apache HTTP Server

```
# /usr/local/apache2/bin/apachectl stop
```

2. Mount logical volume

```
# mount -t ext3 /dev/apachevg/httplv /apache
```

3. Create directory structure

```
# mkdir /apache/bin
# ln -s /usr/local/apache2/bin/apachectl /apache/bin/apachectl
# ln -s /var/log/httpd /apache/logs
# ln -s /usr/local/apache2/modules /apache/modules
```

4. Copy web documents, icons and configuration files to **/apache** directory

```
# cd /usr/local/apache2
# find conf error htdocs icons -print | cpio --pass-through /apache
```

Configuration

On the primary node, edit **/apache/conf/httpd.conf** to reflect the new directory structure and configured to listen on the virtual IP address (*192.168.1.40*), as shown in the following table:

RPM Installation	
Original	New
ServerRoot "/etc/httpd"	ServerRoot "/apache"
Listen 80	Listen 192.168.1.40:80
DocumentRoot "/var/www/html"	DocumentRoot "/apache/html"
<Directory "/var/www/html">	<Directory "/apache/html">
Alias /icons/ "/var/www/icons/"	Alias "/icons/ /apache/icons/"
<Directory "/var/www/icons">	<Directory "/apache/icons">
Alias /error/ "/var/www/error/"	Alias /error/ "/apache/error"
<Directory "/var/www/error">	<Directory "/apache/error">
Source Installation	
Original	New
ServerRoot "/usr/local/apache2"	ServerRoot "/apache"
Listen 80	Listen 192.168.1.40:80
DocumentRoot "/usr/local/apache2/htdocs"	DocumentRoot "/apache/htdocs"
<Directory "/usr/local/apache2/htdocs">	<Directory "/apache/htdocs">
User daemon	User apache
Group daemon	Group apache

Build Application

The commands in this section are to be executed as **root** on the primary node.

1. Create directory **/etc/cluster/apache**.

```
# mkdir /etc/cluster/apache
```

2. Create **/etc/cluster/apache/appconf.xml** as shown:

```
<appconf>
  <global>
    <version>0.1</version>
    <name>apache</name>
    <takeover>normal</takeover>
    <syncrate>2000</syncrate>
    <preferred_node>fc5s1</preferred_node>
  </global>
```

```

<networks>
  <network net="main" ip="192.168.1.40" netmask="255.255.255.0" />
</networks>

<vg>
  <name>apachevg</name>
  <type>filesystems</type>
</vg>

<application>
  <startscript>/apache/bin/apachectl -d /apache -k start</startscript>
  <stopscript>/apache/bin/apachectl -d /apache -k stop</stopscript>
  <maxstoptime>10</maxstoptime>
  <maxstarttime>20</maxstarttime>
</application>
</appconf>

```

The bolded values are configuration-specific, as described in the following table:

Entry	Value
global/preferred_node	Host name of one of the nodes in the cluster, or LEAST_CPU_LOAD
networks/network.net	Same as one of <i>node/network.name</i> in /etc/cluster/clconf.xml
networks/network.ip	Virtual IP address of application
networks/network.netmask	Netmask corresponding to virtual IP address
vg/name	Name of application volume group

3. Create **/etc/cluster/apache/lems.local.xml** as shown:

```

<?xml version="1.0"?>
<lems_config>
  <globals modules="/sbin/cluster/lems/modules"
    programs="/sbin/cluster/lems/programs"
    logs="/var/log/cluster/lems"
  />

  <check>
    <name>flag_check</name>
    <type>internal</type>
    <module>flag_check apache</module>
    <interval>5</interval>
    <action_list>
      <action rc="0" action="NOP"/>
      <action rc="1" action="%RCDATA%"/>
      <action rc="2" action="ABORT"/>
    </action_list>
  </check>

  <check>
    <name>httpd</name>
    <type>internal</type>
    <module>procmon /etc/cluster/apache/httpd.xml</module>
    <interval>10</interval>
    <action_list>
      <action rc="0" action="NOP"/>
      <action rc="1" action="STOP"/>
      <action rc="2" action="FAILOVER"/>
    </action_list>
  </check>

  <check>
    <name>ip</name>
    <type>internal</type>

```

```

<module>ip_module apache</module>
<interval>10</interval>
<action_list>
  <action rc="0" action="NOP"/>
  <action rc="1" action="RUN move_ip"/>
  <action rc="2" action="STOP"/>
</action_list>
</check>
<check>
  <name>fsmonitor</name>
  <type>internal</type>
  <module>fsmon apache</module>
  <interval>10</interval>
  <action_list>
    <action rc="0" action="NOP"/>
    <action rc="1" action="PAUSE 30"/>
    <action rc="2" action="STOP"/>
    <action rc="3" action="FAILOVER"/>
    <action rc="10" action="PAUSE 60"/>
  </action_list>
</check>
</lems_config>

```

4. Create **/etc/cluster/apache/httpd.xml** as shown:

```

<?xml version="1.0"?>
<procmon>
  <global>
    <logdir>/var/log/cluster</logdir>
    <restarts>5</restarts>
    <resetwindow>3600</resetwindow>
    <restartcmd>/apache/bin/apachectl -d /apache -k restart</restartcmd>
  </global>
  <process>
    <label>Apache Parent Process</label>
    <user>root</user>
    <process_string>
      /usr/(sbin|local/apache2/bin)/httpd *
    </process_string>
    <min_count>1</min_count>
    <max_count>1</max_count>
  </process>
  <process>
    <label>Apache HTTP Server</label>
    <user>apache</user>
    <process_string>
      /usr/(sbin|local/apache2/bin)/httpd *
    </process_string>
    <min_count>2</min_count>
    <max_count>30</max_count>
  </process>
</procmon>

```

5. Verify LEMS configuration by executing:

```

# lems.pl --config /etc/cluster/apache/lems.local.xml \
> --application apache --verbose --check --file /dev/tty

```

If successful, a result similar to the following will be displayed.

```

INFO 10/08/2006 11:34:39 Using modules from : /sbin/cluster/lems/modules
INFO 10/08/2006 11:34:39 Using programs from : /sbin/cluster/lems/programs
INFO 10/08/2006 11:34:39 Writing logs to : /var/log/cluster/lems

```

```

INFO 10/08/2006 11:34:39 Listening on port : 9917
INFO 10/08/2006 11:34:39 Global initialisation complete.
INFO 10/08/2006 11:34:39 Started local server on port 9917
INFO 10/08/2006 11:34:39 Validating monitor entry ip...
INFO 10/08/2006 11:34:39 Validated monitor entry ip successfully.
INFO 10/08/2006 11:34:39 Validating monitor entry fsmonitor...
INFO 10/08/2006 11:34:39 Validated monitor entry fsmonitor successfully.
INFO 10/08/2006 11:34:39 Validating monitor entry httpd...
INFO 10/08/2006 11:34:39 Validated monitor entry httpd successfully.
INFO 10/08/2006 11:34:39 Validating monitor entry flag_check...
INFO 10/08/2006 11:34:39 Validated monitor entry flag_check successfully.
INFO 10/08/2006 11:34:39 Check mode - transferring validated config to remote
node.
INFO 10/08/2006 11:34:39 Configuration transferred successfully.
INFO 10/08/2006 11:34:39 Calculated a check interval of 2.5 seconds.

```

6. Build **apache**.

```
# clbuildapp --application apache --sync
```

The output of a successful build is shown below:

```

INFO 09/08/2006 14:44:23 Backups directory defaulted to /clbackup
INFO 09/08/2006 14:44:23
INFO 09/08/2006 14:44:23 Validation of Application 'apache' started.
INFO 09/08/2006 14:44:23 ['/var/log/cluster/build/apache-check-
300608091444.log']
INFO 09/08/2006 14:44:24 Initial Validation of Application successful.
INFO 09/08/2006 14:44:24
INFO 09/08/2006 14:44:24 NOTE: Build of new application is being performed.
INFO 09/08/2006 14:44:24
INFO 09/08/2006 14:44:24 Host Environment Validation started.
INFO 09/08/2006 14:44:24 ['/var/log/cluster/build/apache-envcheck-
300608091444.log']
INFO 09/08/2006 14:44:26 Host Environment Validation successful.
INFO 09/08/2006 14:44:26
INFO 09/08/2006 14:44:26 Cluster state : RUNNING
INFO 09/08/2006 14:44:26 Application state: UNDEFINED
INFO 09/08/2006 14:44:26
INFO 09/08/2006 14:44:26 Volume Group Configuration started.
INFO 09/08/2006 14:44:26 ['/var/log/cluster/build/apache-lvm-300608091444.log']
INFO 09/08/2006 14:44:28 Volume Group Configuration successful.
INFO 09/08/2006 14:44:28
INFO 09/08/2006 14:44:28 Application Resource Allocation started.
INFO 09/08/2006 14:44:28 ['/var/log/cluster/build/apache-build-
300608091444.log']
INFO 09/08/2006 14:44:34 Application Resource Allocation successful.
INFO 09/08/2006 14:44:34
INFO 09/08/2006 14:44:34 Application Data Synchronisation started.
INFO 09/08/2006 14:44:34 ['/var/log/cluster/build/apache-syncdata-
300608091444.log']
Storage Syncing: 0Mb/ 0Mb [100 % Complete]
INFO 09/08/2006 14:44:43 Application Data Synchronisation successful.
INFO 09/08/2006 14:44:43

```

Run Application

The commands in this section are executed as **root** on either node except where indicated.

1. If necessary, form the cluster

```
# clform
```

2. Start **apache** on the primary node (*fc5s1*).

```
# clrunapp --application apache --node fc5s1
```

3. Verify the application state using **clstat**.

```
# clstat --application apache
```

A result similar to the following will be returned if **apache** is running:

```
Cluster: cluster1 - UP

Application      Node      State  Runnig  Monitor  Stale  Fail-over?
  apache         fc5s1    STARTED  0:00:18  Running    0      Yes

File Systems

Mount Point      Valid  Type      State  % Complete  Completion
/apache          both   drbd      Sync

Process Monitors

Name  Status  Restarts  Current  Reset at
httpd  Running    5         0        N/A

General Monitors

Type      Name      Status
Flag Check  flag_check  Running
FS Monitor  fsmonitor  Running
IP Monitor  ip         Running
```

4. On the node where **apache** is running, list the **httpd** processes by executing:

```
# ps -u root -f | grep httpd | grep -v grep ; \
> ps -u apache -f | tail -1
```

An output similar to the following process list should be returned. The path of **httpd** will be either **/usr/sbin** or **/usr/local/apache2/bin** depending on installation method:

```
root    2744    1  2   14:54  ?   00:00:00  /usr/sbin/httpd -d /apache -k start
apache  2747  2744  0   14:54  ?   00:00:00  /usr/sbin/httpd -d /apache -k start
```

5. Verify that the HTTP server is accepting connections by browsing to its virtual IP address (192.168.1.40) from any remote computer, or by using **curl** as follows:

```
# curl http://192.168.1.40
```

If the HTTP server is running, **curl** will display the HTML code of the default home page.

6. On the node where **apache** is running, stop it by executing:

```
# clhaltapp --application apache
```

7. Start **apache** on the secondary node (*fc5s2*).

```
# clrunapp --application apache --node fc5s2
```

8. Repeat steps 3 to 6 to test **apache** on the secondary node.