# Clustering using Mod-jk

## Apache side configuration.

1. Download Apache and Install it.

   http://www.apache.org/dist/httpd/httpd-2.2.29-win32-src.zip (Windows)
   http://www.apache.org/dist/httpd/httpd-2.2.27.tar.gz (Linux)

   ( Note: if any other versions required refer http://www.apache.org/dist/httpd/)

2. Download mod-cluster mod-cluster-1.0.3.GA-xxx-ssl.zip.
   http://downloads.jboss.org/mod_cluster//1.2.0.Final/mod_cluster-1.2.0.Final-linux2-x64-ssl.tar.gz (Linux)
   http://downloads.jboss.org/mod_cluster//1.2.0.Final/mod_cluster-1.2.0.Final-windows-x64-ssl.zip (Windows)
   ( Note: if any other versions required refer http://mod-cluster.jboss.org/downloads/1-2-0-Final)

3. Load the mod_jk.so in the apache modules folder
   mod_jk.so
   The module needs to be copied in the APACHE_HOME/modules dir.

4. Add the below line in ***httpd.conf*** file
   Include conf/mod_jk.conf

5. Create a ***conf*** folder in the apache distribution folder ***(apachehome/conf)*** and create a new file ***mod_jk.conf (apachehome/conf/mod_jk.conf)***

   ```
   #Load mod_jk module
   ## Specify the filename of the mod_jk lib
   LoadModule jk_module modules/mod_jk.so
   ## Where to find workers.properties
   JkWorkersFile conf/workers.properties
   ################ mod-jk.conf Configuration Start here ##############
   ## Where to put jk logs
   JkLogFile logs/mod_jk.log
   #
   ## Set the jk log level [debug/error/info]
   #JkLogLevel info
   JkLogLevel debug
   ```

#
## Select the log format
JkLogStampFormat  "[%a %b %d %H:%M:%S %Y]"
#
## JkOptions indicates to send SSK KEY SIZE
JkOptions +ForwardKeySize +ForwardURICompat -ForwardDirectories
#
## JkRequestLogFormat
JkRequestLogFormat "%w %V %T"
<VirtualHost 0.0.0.0:80>(0.0.0.0:80 apache server ip and its corresponding port)
 JkMount /ShoppingCart loadbalancer
 JkMount /ShoppingCart/* loadbalancer
 JkMount /jkstatus jkstatus
 JkMount /prweb/PRServlet loadbalancer
 JkMount /prweb/PRServlet/* loadbalancer
 </VirtualHost>

################ mod-jk.conf Configuration End here ###############

6. Create a new file ***workers.properties*** in the conf folder ***(apachehome/conf/workers.properties)***
   Add the following lines in the workers.properties file

################ workers.properties Configuration Start here ###############
# Define list of workers that will be used # for mapping requests
#worker.list=node1,node2
worker.list=loadbalancer,jkstatus
# Define Node1 : modify the host as your host IP or DNS name.
worker.node1.port=8009
worker.node1.host=(hostname of node 1)
worker.node1.type=ajp13
worker.node1.lbfactor=1
worker.node1.ping_mode=A
#worker.node1.cachesize=10
# Define Node2 : modify the host as your host IP or DNS name.
worker.node2.port=8009
worker.node2.host=(hostname of node 2)
worker.node2.type=ajp13
worker.node2.lbfactor=1
worker.node2.ping_mode=A
#worker.node2.cachesize=10
# Load-balancing behaviour
worker.loadbalancer.type=lb
worker.loadbalancer.balance_workers=node1,node2
worker.loadbalancer.sticky_session=1
#worker.list=loadbalancer

# Status worker for managing load balancer
worker.jkstatus.type=status
################ workers.properties Configuration End Here ################

7. Create a new file ***uriworkermap.properties*** in the conf folder ***(apachehome/conf/uriworkermap.properties)***

Add the following lines in the uriworkermap.properties file

################ uriworkermap.properties Configuration Start here ################
# Simple worker configuration file
# Mount the Servlet context to the ajp13 worker
/jmx-console=loadbalancer
/jmx-console/*=loadbalancer
/web-console=loadbalancer
/web-console/*=loadbalancer
/ShoppingCart/*=loadbalancer
/prweb/PRServlet/*=loadbalancer
################ uriworkermap.properties Configuration End here ################

# Jboss Side Configuration

## For Domain mode

********** Running jboss in domain mode**********

## Clustering in same machine

1. Unzip Jboss EAP 6.1 distribution on a path of your choice, for example:

   D:\jboss-eap-6_Demo

2. Go to ***Domain.xml*** file and create a new server group and give the profile as ha(enables cluster)

***eg***:
```
<server-groups>
    <server-group name="ha-server-group" profile="ha">
      <jvm name="default">
        <heap size="64m" max-size="512m"/>
      </jvm>
      <socket-binding-group ref="ha-sockets"/>
    </server-group>
</server-groups>
```

3. Go to ***Host.xml*** file and create servers as required and map it to the corresponding server group and give a unique name to the host

***eg:***
```
<host name="master" xmlns="urn:jboss:domain:1.3">
</host>
```

***eg:***
```
<servers>
    <server name="ha-server-1" group="ha-server-group" auto-start="true">
      <socket-bindings port-offset="100"/>
    </server>
    <server name="ha-server-2" group="ha-server-group" auto-start="true">
      <socket-bindings port-offset="200"/>
    </server>
</servers>
```

4. Before we start the domain controller,create a management user for the domain controller. This user is necessary when the host controller needs to establish a connection to the domain controller. For this there is an add-user.sh/ add-user.bat script in the bin directory of the JBoss AS distribution.

   $ ./add-user.sh/ add-user.bat

   What type of user do you wish to add?

a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : domainadmin
Password :
Re-enter Password :
Are you sure you want to add user 'domain' yes/no? y

About to add user 'domain for realm 'ManagementRealm'
Is this correct yes/no? y
Added user 'domain' to file '/standalone/configuration/mgmt-users.properties'
Added user 'domain' to file '/domain/configuration/mgmt-users.properties'

Is this new user going to be used for one AS process to connect to another AS process e.g.slave domain controller?
yes/no? ***y***

5. You need to answer the last question with yes or y to indicate that the user will be used to connect to the domain controller from the host controller. The generated secret value is the Base64-encoded password of the new created user.

6. Now from the terminal pointing the bin of the jboss extraction run the following command
sudo sh domain.sh ï b 1.0.0.0

7. Unzip another Jboss EAP 6.1 distribution in the **SAME MACHINE** on a path of your choice, for example:

D:\jboss-eap-6_Demo2

8. Go to ***Domain.xml*** file and create a new server group and give the profile as ha(enables cluster)

***eg***:
```
<server-groups>
    <server-group name="ha-server-group" profile="ha">
        <jvm name="default">
            <heap size="64m" max-size="512m"/>
        </jvm>
        <socket-binding-group ref="ha-sockets"/>
    </server-group>
</server-groups>
```

9. The first thing is to choose a unique name for each host in our domain to avoid name conflicts. Otherwise, the default is the host name of the server.
Go to **Host.xml** file of machine 2

***eg:***
```
<host name="server1" xmlns="urn:jboss:domain:1.3">
</host>
```

10. In the same ***Host.xml*** file create servers as required and map it to the corresponding server group and map the domain controller in its corresponding tag.

Note:(The host controller needs to know how to establish a connection to the domain controller. The following configuration on the host specifies where the domain controller is located.
Thus, the host controller can register to the domain controller itself. The remote tag must include the username and the security realm of the domain controller).

***eg:***
```
<servers>
    <server name="ha-server-1" group="ha-server-group" auto-start="true">
        <socket-bindings port-offset="100"/>
    </server>
    <server name="ha-server-2" group="ha-server-group" auto-start="true">
        <socket-bindings port-offset="200"/>
    </server>
</servers>
```

***eg*** :
```
<host name="server1" xmlns="urn:jboss:domain:1.3">
        <domain-controller>
                <remote host="localhost"  port="${jboss.domain.master.port:9999}"
                security-realm="ManagementRealm"/>
        </domain-controller>
</host>
```

11. Another thing is to announce the secret value. This is needed for authentication together with the username.
The secret value was the last output from add-user.shscript that we executed previously in same machine. This value is only the Base64-encoded password of the domain user.
Go to **Host.xml** and add the secret value(64 bit encoded password obtained from add-user process in domain controller) under the server-identities.

```
<host name="server1" xmlns="urn:jboss:domain:1.3">
 <management>
      <security-realms>
        <security-realm name="ManagementRealm">
          <server-identities>
```

```
                <!-- Replace this with either a base64 password of your own, or use a vault with a vault exp
                <secret value="c2VjcmV0"/>
            </server-identities>
            <authentication>
                <local default-user="$local" />
                <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>
            </authentication>
        </security-realm>
        ...
    </security-realms>
    ...
</management>
...
</host>
```

12. Now from the terminal pointing the bin of the jboss extraction run the following command
    sudo sh domain.sh ï b 1.0.0.0(where 1.0.0.0 is the Domain controller/Machine1 ip)

13. Now navigate to the url:ñhttp://localhost:9990/consoleñ which shows the admin console of jboss

## Clustering in different machines

14. Unzip two Jboss EAP 6.1 distribution on the corresponding machines with path of your choice.

***In Machine1(Consider the Machine 1 as a Domain Controller )***

15. Go to Domain.xml file and create a new server group and give the profile as ha(ha profile enables cluster)

***eg***:
```
<server-groups>
    <server-group name="ha-server-group" profile="ha">
        <jvm name="default">
            <heap size="64m" max-size="512m"/>
        </jvm>
        <socket-binding-group ref="ha-sockets"/>
    </server-group>
</server-groups>
```

16. Go to Host.xml file and create servers as required and map it to the corresponding server group and with the following configuration in the host.xml file, the host becomes the domain controller

***eg.*** <domain-controller>
<local/>
</domain-controller>

***eg:***<servers>
<server name="ha-server-1" group="ha-server-group" auto-start="true">
<socket-bindings port-offset="100"/>
</server>
<server name="ha-server-2" group="ha-server-group" auto-start="true">
<socket-bindings port-offset="200"/>
</server>

17. Before we start the domain controller, we need to create a management user for the domain controller. This user is necessary when the host controller needs to establish a connection to the domain controller. For this there is an add-user.sh /add-user.bat script in the bin directory of the JBoss AS distribution.
$./add-user.sh/ add-user.bat

What type of user do you wish to add?
a) Management User (mgmt-users.properties)
b) Application User (application-users.properties)
(a): a

Enter the details of the new user to add.
Realm (ManagementRealm) :
Username : domainadmin
Password :
Re-enter Password :
Are you sure you want to add user 'domain' yes/no? y

About to add user 'domain for realm 'ManagementRealm'
Is this correct yes/no? y
Added user 'domain' to file '/standalone/configuration/mgmt-users.properties'
Added user 'domain' to file '/domain/configuration/mgmt-users.properties'

Is this new user going to be used for one AS process to connect to another AS process e.g.slave domain controller?
yes/no? ***y***

18. You need to answer the last question with yes or y to indicate that the user will be used to connect to the domain controller from the host controller. The generated secret value is the Base64-encoded password of the new created user.

19. Make the highlighted changes in the primary jboss server***(Domain.xml) /Machine 1***or Domain machine

***eg:***<subsystem xmlns="urn:jboss:domain:web:1.5" default-virtual-server="default-host" native="false" ***instance-id="node1"***>
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<connector name="ajp" protocol="AJP/1.3" scheme="http" socket-binding="ajp"/>
<virtual-server name="default-host" enable-welcome-root="true">
<alias name="node1 machine ip"/>
<alias name=" node1hostname"/>
</virtual-server>
</subsystem>

20. Now from the bin terminal of the jboss extraction run the following command
sudo sh domain.sh ï b 1.0.0.0(where 1.0.0.0 is the domain controller/Machine1 ip)

21. Now the last thing which you would have to do is to deploy the application using the console *http://1.0.0.0):9990/console* which would be running in the domain controller hence you would be using the URL *http://1.0.0.0:8080/Application* to view the application deployed.

***In Machine2(Consider this machine as Slave/Host Controller )***

22. The first thing is to choose a unique name for each host in our domain to avoid name conflicts. Otherwise, the default is the host name of the server.
Go to **Host.xml** file of machine 2

***eg:***<host name="server1" xmlns="urn:jboss:domain:1.3">
</host>

23. The host controller needs to know how to establish a connection to the domain controller. The following configuration on the host specifies where the domain controller is located. Thus, the host controller can register to the domain controller itself. The remote tag must include the username and the security realm of the domain controller.

***eg:***<host name="server1" xmlns="urn:jboss:domain:1.3">
<domain-controller>
<remotehost="${jboss.domain.master.address}"port="${jboss.domain.master.port:9999}
" username="**domainadmin**" security-realm="ManagementRealm"/>
</domain-controller>
</host>

24. Go to **Host.xml** file and create servers as required and map it to the corresponding server group

***eg***:
```
<servers>
    <server name="ha-server-1" group="ha-server-group" auto-start="true">
        <socket-bindings port-offset="100"/>
    </server>
    <server name="ha-server-2" group="ha-server-group" auto-start="true">
        <socket-bindings port-offset="200"/>
    </server>
```

25. Another thing is to announce the secret value. This is needed for authentication together with the username.
The secret value was the last output from add-user.shscript that we executed previously in machine 1. This value is only the Base64-encoded password of the domain user.
Go to **Host.xml** and add the secret value(64 bit encoded password obtained from add-user process in domain controller) under the server-identities.

***eg***:
```
<host name="server1" xmlns="urn:jboss:domain:1.3">
 <management>
      <security-realms>
        <security-realm name="ManagementRealm">
          <server-identities>
              <!-- Replace this with either a base64 password of your own, or use a vault exp
              <secret value="c2VjcmV0"/>
          </server-identities>
          <authentication>
             <local default-user="$local" />
             <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>
          </authentication>
        </security-realm>
        ...
      </security-realms>
      ...
   </management>
  ...
</host>
```

26. Now from the bin terminal of the jboss extraction run the following command

***eg***: sudo sh domain.sh ï b 1.0.0.0(where 1.0.0.0 is the Domain controller/Machine1 ip)

27. Make the highlighted changes in the secondary jboss server***(Domain.xml)/Machine 2*** or slave machine

***eg***:<subsystem xmlns="urn:jboss:domain:web:1.5" default-virtual-server="default-host" native="false" ***instance-id="node2"***>
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<connector name="ajp" protocol="AJP/1.3" scheme="http" socket-binding="ajp"/>
<virtual-server name="default-host" enable-welcome-root="true">
<alias name=" node2machine ip "/>
<alias name=" node2hostname "/>
</virtual-server>
</subsystem>