CLUSTERING (HORIZONTAL AND VERTICAL) using mod-cluster

Steps to be followed for clustering setup:

For Standalone Mode

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

```
 D:\jboss-eap-6_Demo\nodeA\node1
   D:\jboss-eap-6_Demo\nodeA\node2
```

(Note: Copy the standalone folder and paste in the same file and rename it as node1 and node 2)

2. In order to start your cluster nodes you need to provide an unique node name and, since you are running multiple servers on the same machine specify a port offset so that you donât have port conflicts between servers.

3. Starting the Node 1:

```
standalone.bat -c standalone-ha.xml -Djboss.node.name=nodeA -
Djboss.server.base.dir=./node1
```

4. Starting the Node 2:

```
standalone.bat -c standalone-ha.xml -Djboss.node.name=nodeA -
Djboss.server.base.dir=./node2 -     Djboss.socket.binding.port-offset=200
```

5. **Standalone Mode:**

- We would have to add the instance-id attribute in web subsystem as shown below in **standalone-ha.xml**

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host"
instance-id="${jboss.node.name}" native="false">
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<connector name="ajp" protocol="AJP/1.3" scheme="http" socket-binding="ajp"/>
</subsystem>
```

- You just have to add the proxy-list in the attribute in mod-cluster-config of modcluster subsystem, which would be having IP Address and Port on which your Apache server is running so that JBoss server can communicate with it, as shown below in **standalone-ha.xml** for the respective profile ha which is been used.

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
<mod-cluster-config advertise-socket="modcluster" proxy-list="127.0.0.1:80">
```

```
<!--<load-metric type="busyness"/>-->(fault tolerance)
<load-metric type="requests"/>(for round robin technique)
</mod-cluster-config>
</subsystem>
```

6. Check:

   http://localhost:8080 (node1) and http://localhost:8180 (node2)

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

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. Domain Mode(1st Extraction/Domain controller):

- We would have to add the instance-id attribute in web subsystem as shown below in domain.xml for the respective profile ha and full-ha in **domain.xml** file

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host"
instance-id="${jboss.node.name}" native="false">
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<connector name="ajp" protocol="AJP/1.3" scheme="http" socket-binding="ajp"/>
</subsystem>
```

- You just have to add the proxy-list in the attribute in mod-cluster-config of modcluster subsystem, which would be having IP Address and Port on which your Apache server is running so that JBoss server can communicate with it, as shown below in **domain.xml** for the respective profile ha which is been used.

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
<mod-cluster-config advertise-socket="modcluster" proxy-list="127.0.0.1:80">
<!--<load-metric type="busyness"/>-->(fault tolerance)
<load-metric type="requests"/>(for round robin technique)
</mod-cluster-config>
</subsystem>
```

7. 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
   ```

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

   D:\jboss-eap-6_Demo2

9. 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>
   ```

10. 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>
    ```

11. 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>
```

12. 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>
```

13. Domain Mode(2^{ND} Extraction /Slave 1):

- We would have to add the instance-id attribute in web subsystem as shown below in domain.xml for the respective profile ha and full-ha in **domain.xml** file

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host"
instance-id="${jboss.node.name}" native="false">
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<connector name="ajp" protocol="AJP/1.3" scheme="http" socket-binding="ajp"/>
</subsystem>
```

- You just have to add the proxy-list in the attribute in mod-cluster-config of modcluster subsystem, which would be having IP Address and Port on which your Apache server is running so that JBoss server can communicate with it, as shown below in **domain.xml** for the respective profile ha which is been used.

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
<mod-cluster-config advertise-socket="modcluster" proxy-list="127.0.0.1:80">
<!--<load-metric type="busyness"/>-->(fault tolerance)
<load-metric type="requests"/>(for round robin technique)
</mod-cluster-config>
</subsystem>
```

14. 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)
15. Now navigate to the url:ñhttp://localhost:9990/consoleñ which shows the admin console of jboss

Clustering in different machines

1. 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)

2. 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>
```

3. 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>
```

4. Before we start the domain controller is 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 script in the bin directory of the JBoss AS distribution.

```
 $./add-user.sh

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 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)

7. 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)

8. 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

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

9. 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.

```
 <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>
```

10. 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>
```

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 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.

```
<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 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)

13. Now the last thing is opening the application in machine2, hence you would be using the URL *http:/2.0.0.0:8080/Application* (where 2.0.0.0 is the host controller/ Machine 2 ip)

JBOSS SIDE CONFIGURATION (Integrating Apache and Jboss)

Domain Mode(domain controller/Machine1):

- We would have to add the instance-id attribute in web subsystem as shown below in domain.xml for the respective profile ha and full-ha in **domain.xml** file

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" instance-id="${jboss.node.name}" native="false">
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<connector name="ajp" protocol="AJP/1.3" scheme="http" socket-binding="ajp"/>
</subsystem>
```

- You just have to add the proxy-list in the attribute in mod-cluster-config of modcluster subsystem, which would be having IP Address and Port on which your Apache server is running so that JBoss server can communicate with it, as shown below in **domain.xml** for the respective profile ha which is been used.

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
<mod-cluster-config advertise-socket="modcluster" proxy-list="127.0.0.1:80">
<!--<load-metric type="busyness"/>-->(fault tolerance)
<load-metric type="requests"/>(for round robin technique)
</mod-cluster-config>
</subsystem>
```

Domain Mode(HostController /Machine 2):

- We would have to add the instance-id attribute in web subsystem as shown below in domain.xml for the respective profile ha and full-ha in **domain.xml** file

```
<subsystem xmlns="urn:jboss:domain:web:1.1" default-virtual-server="default-host" instance-id="${jboss.node.name}" native="false">
<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http"/>
<connector name="ajp" protocol="AJP/1.3" scheme="http" socket-binding="ajp"/>
</subsystem>
```

- You just have to add the proxy-list in the attribute in mod-cluster-config of modcluster subsystem, which would be having IP Address and Port on which your Apache server is running so that JBoss server can communicate with it, as shown below in **domain.xml** for the respective profile ha which is been used.

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.0">
<mod-cluster-config advertise-socket="modcluster" proxy-list="127.0.0.1:80">
<!--<load-metric type="busyness"/>-->(fault tolerance)
<load-metric type="requests"/>(for round robin technique)
</mod-cluster-config>
</subsystem>
```

****************** Configuring Apache Webserver*********************

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 following modules in the apache modules folder
   mod_slotmem.so
   mod_manager.so
   mod_proxy_cluster.so
   mod_advertise.so

The modules need to be copied in the APACHE_HOME/modules dir.

4. Then, add this configuration to the bottom of your Apache's httpd.conf file(if the below configuration is there by default no changes are to be made)

```
Listen *:80
<VirtualHost *:80>
<Directory />
 Order deny,allow
 Deny from all
 Allow from *
</Directory>
KeepAliveTimeout 60
MaxKeepAliveRequests 0
 ManagerBalancerName mycluster
 AdvertiseFrequency 5
</VirtualHost>
```

Here we assume that your Apache listen to the IP Address 127.0.0.1 and accepts requests on port 80.
(Note: if the maxhost attribute is inside the <virtual host>,configure maxhost attribute outside the <virtual host> system)

References:

1. http://blog.akquinet.de/2012/06/29/managing-cluster-nodes-in-domain-mode-of-jboss-as-7-eap-6/
2. https://docs.jboss.org/author/display/AS71/AS7+Cluster+Howto
3. http://middlewaremagic.com/jboss/?cat=4
4. http://middlewaremagic.com/jboss/?p=147
5. http://middlewaremagic.com/jboss/?p=1969
6. https://developer.jboss.org/message/906774?_sscc=t
7. https://developer.jboss.org/wiki/JBossAS7UserGuide
8. https://developer.jboss.org/message/718901
9. https://docs.jboss.org/author/display/AS72/Domain+Setup