

Welcome to...

JBoss Clustering
featuring JBoss 6 EAP / 7.x.x

Cav Edwards

Swarm Software – Research Laboratories

Table of Contents

1 Overview.....	3
2 Domain Controller Configuration.....	4
2.1 Adding Management Users.....	4
2.2 Host.XML in the Domain-Controller.....	4
2.2.1 Setting the Domain Controller virtual host name.....	5
2.2.2 Specifying which host becomes the domain controller.....	6
2.2.3 Make the Domain Controller visible outside localhost.....	6
2.3 Domain.xml in the domain controller.....	6
2.3.1 Server Groups and clusters.....	6
3 Managed-Server HOST Configuration.....	6
3.1 Management Password at each host.....	7
3.2 Host.xml in the Managed-Server host.....	7
3.2.1 Managed-Server Virtual Hostname.....	7
3.2.2 Specifying the Managed server Host.....	7
3.2.3 Ensuring the Managed-Server Host can talk to the Domain-Controller.....	7
3.3 Establishing the Cluster.....	8
4 Managed Servers and Clusters.....	8
4.1 Deploying Applications to a Cluster.....	9
5 JBoss Profiles.....	10
5.1 Profile Configuration in a Domain.....	11
6 Apache HA Configuration.....	11
6.1 Httpd.conf.....	11
6.2 Setting the listen port for Apache.....	11
6.3 Registering mod_cluster with Apache.....	11
6.4 Mod_Cluster configuration.....	11
6.4.1 Listen 10.25.2.22:10001.....	12
6.4.2 MemManagerFile /var/cache/httpd.....	12
6.4.3 VirtualHost Listen 10.25.2.22:10001.....	12
6.4.3.1 Directory.....	12
6.4.3.1.1 Order allow,deny.....	12
6.4.4 Location /mod_cluster-manager.....	12
6.4.5 ManagerBalancerName clusterA.....	12
7 Sticky sessions.....	13
8 Testing the Cluster.....	13

1 Overview

This document explains how to setup and configure JBoss Clustering. It also includes Apache HA clustering for JBoss.

2 Domain Controller Configuration

2.1 Adding Management Users

For each host you would like to take part in clustering you need to specify a SEPARATE user. Within the Domain Controllers bin folder, run add-user.(sh or cmd)

Specify you would like to add a management user. The username MUST MATCH the Managed Server host name you wish to take part in the cluster. This host name will be specified in the configuration of managed server host below. It is not the physical hostname but a domain-controller hostname, a unique name allocated to each physical host that will host managed-servers on behalf of the domain-controller.

NOTE: host name in JBoss clustering terminology is a name you assign to a machine that is capable of running JBoss and NOT a hostname in TCP/IP and networking terms.

```

BLLAPP1
[root@bllapp1 bin]# ./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 : hostNameOfManagedServerHost
Password :
Re-enter Password :
About to add user 'hostNameOfManagedServerHost' for realm 'ManagementRealm'
Is this correct yes/no? y
Added user 'hostNameOfManagedServerHost' to file '/usr/local/jboss6_EAP6/jboss-eap-6.0/standalone/configuration/mgmt-users.properties'
Added user 'hostNameOfManagedServerHost' to file '/usr/local/jboss6_EAP6/jboss-eap-6.0/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
To represent the user add the following to the server-identities definition <secret value="amJvc3M=" />
[root@bllapp1 bin]#

```

Note the base64 representation of the password from above. It will be used to configure the managed server host machine later.

2.2 Host.XML in the Domain-Controller

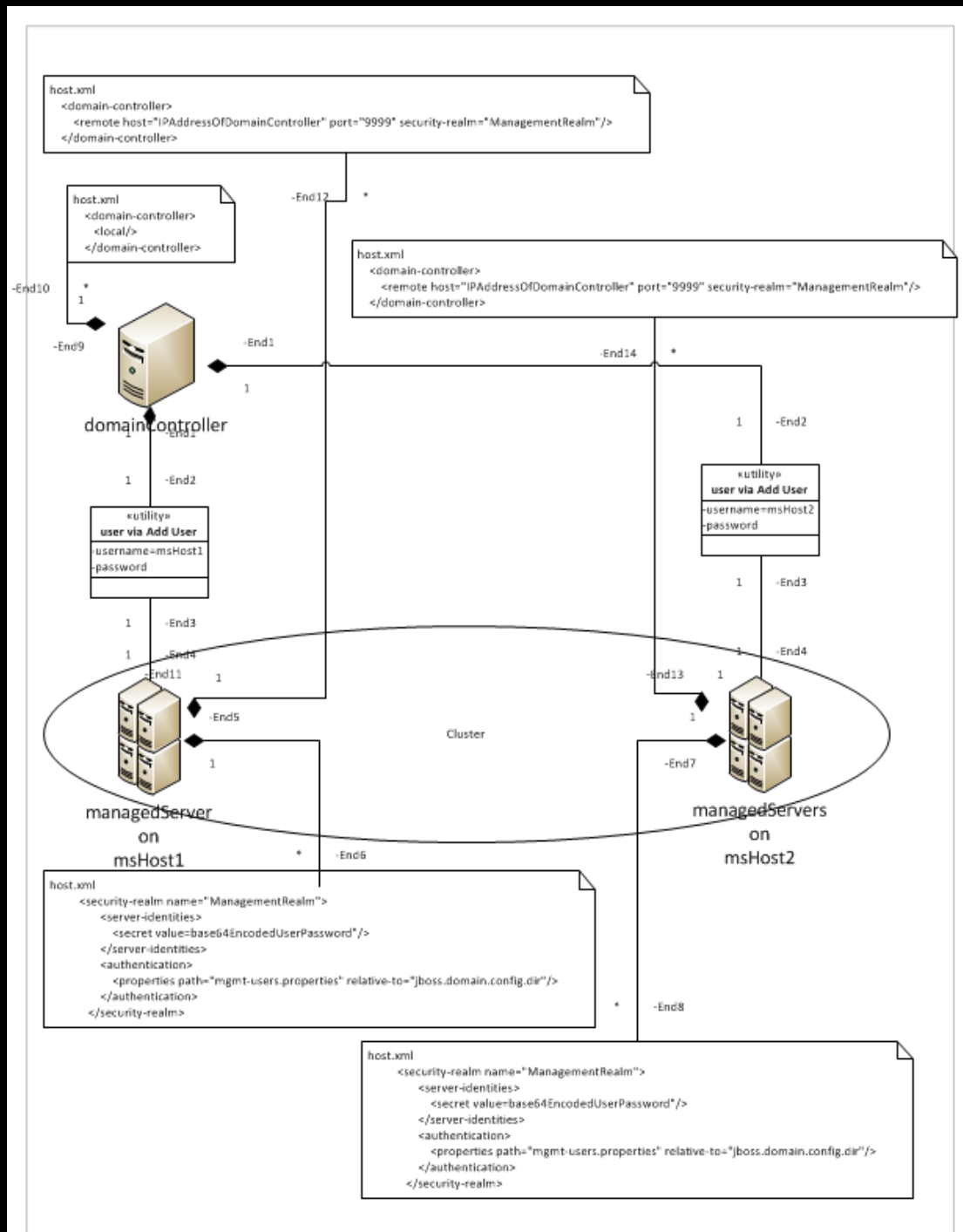
Host.xml at the domain controller determines how the domain establishes itself. For the domain controller the host.xml file determines that it is to administer the communications throughout the domain.

Consider the following diagram.

There exists a Domain Controller, through which deployments are managed.

There exist multiple hosts on which Managed-Servers run.

The domain controller and each host has/have a host.xml file through which the clustered estate is established.



2.2.1 Setting the Domain Controller virtual host name

Within the Domain Controller <JBoss Install path>/domain/configuration/host.xml file.

Check for this line at the top:

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

The hostname is master in this case. It can be any name you would like but it must be unique. We will be setting one in the managed server host too.

Specify that this is indeed the domain controller with the following XML, which should already be set by the install.

2.2.2 Specifying which host becomes the domain controller

The domain controller stanza must have local within it to indicate this is the domain controller.

```
<domain-controller>
  <local/>
  <!-- Alternative remote domain controller configuration with a host and port -->
  <!-- <remote host="${jboss.domain.master.address}" port="${jboss.domain.master.port:9999}" security-realm="ManagementRealm"/>
-->
  <!-- <remote host="${jboss.domain.master.address}" port="${jboss.domain.master.port:9999}" username="jboss" security-
realm="ManagementRealm"/> -->
</domain-controller>
```

2.2.3 Make the Domain Controller visible outside localhost

Now we need to make sure the management, public and insecure/IIOP sockets are bound to the public IP address for the host so we can administer the node remotely.

```
<interfaces>
  <interface name="management">
    <inet-address value="${jboss.bind.address.management:IPAddressOfThisDomainController}"/>
  </interface>
  <interface name="public">
    <inet-address value="${jboss.bind.address: IPAddressOfThisDomainController }"/>
  </interface>
  <interface name="unsecure">
    <!-- Used for IIOP sockets in the standard configuration.
    To secure JacORB you need to setup SSL -->
    <inet-address value="${jboss.bind.address.unsecure: IPAddressOfThisDomainController }"/>
  </interface>
</interfaces>
```

2.3 Domain.xml in the domain controller

Domain.xml specifies the configuration for the domain-controller.

2.3.1 Server Groups and clusters

A server group determines what clusters exist in the domain.

```
<server-groups>
  <server-group name="clusterA" profile="full">
    <jvm name="default">
      <heap size="1303m" max-size="1303m"/>
      <permgen max-size="256m"/>
    </jvm>
    <socket-binding-group ref="full-sockets"/>
  </server-group>
  <server-group name="clusterB" profile="full">
    <jvm name="default">
      <heap size="1303m" max-size="1303m"/>
      <permgen max-size="256m"/>
    </jvm>
    <socket-binding-group ref="full-sockets"/>
  </server-group>
  <server-group name="clusterC" profile="full-ha">
    <jvm name="default">
      <heap size="1303m" max-size="1303m"/>
      <permgen max-size="256m"/>
    </jvm>
    <socket-binding-group ref="full-ha-sockets"/>
  </server-group>
</server-groups>
```

In the example above three clusters are specified, clusterA, clusterB and clusterC.

3 Managed-Server HOST Configuration

Now we configure the managed-server host machine.

3.1 Management Password at each host

We must specify the base64 encoded form of the password we obtained from configuring the domain controller for this host, See : [2.1 Adding Management Users](#).

```
<management>
  <security-realms>
    <security-realm name="ManagementRealm">
      <server-identities>
        <secret value="amJvc3M=" />
      </server-identities>
      <authentication>
        <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>
      </authentication>
    </security-realm>
  <!--
    <security-realm name="ManagementRealm">
      <authentication>
        <local default-user="$local" />
        <properties path="mgmt-users.properties" relative-to="jboss.domain.config.dir"/>
      </authentication>
    </security-realm>
  -->
```

3.2 Host.xml in the Managed-Server host

Each managed-server host needs a host.xml file that defines the servers that are to run at this host and which cluster / server-group from the domain-controller the server is to run under.

3.2.1 Managed-Server Virtual Hostname

Within <JBoss Install path>/domain/configuration/host.xml file. We must give the managed server host a unique name.

```
<host name="host1" xmlns="urn:jboss:domain:1.3">
```

NOTE: Remember, this hostname must be the same as the management user added to the domain controller.

3.2.2 Specifying the Managed server Host

Now we ensure this managed server host talks to the domain controller.

This then makes this host a client of the domain controller.

```
<domain-controller>
  <remote host="IPAddressOfDomainController" port="9999" security-realm="ManagementRealm"/>
  <!-- <local/> -->
</domain-controller>
```

3.2.3 Ensuring the Managed-Server Host can talk to the Domain-Controller

Finally we must ensure that the domain controller can communicate with the managed server host by changing the management, public and unsecure/IIOP sockets addresses.

```
<interfaces>
  <interface name="management">
    <inet-address value="{jboss.bind.address.management: IPAddressOfThisManagedServerHost}"/>
  </interface>
  <interface name="public">
    <inet-address value="{jboss.bind.address: IPAddressOfThisManagedServerHost }"/>
  </interface>
  <interface name="unsecure">
    <!-- Used for IIOP sockets in the standard configuration.
         To secure JacORB you need to setup SSL -->
    <inet-address value="{jboss.bind.address.unsecure: IPAddressOfThisManagedServerHost }"/>
  </interface>
</interfaces>
```

When these steps are complete, start the domain controller. Then start the managed server hosts.

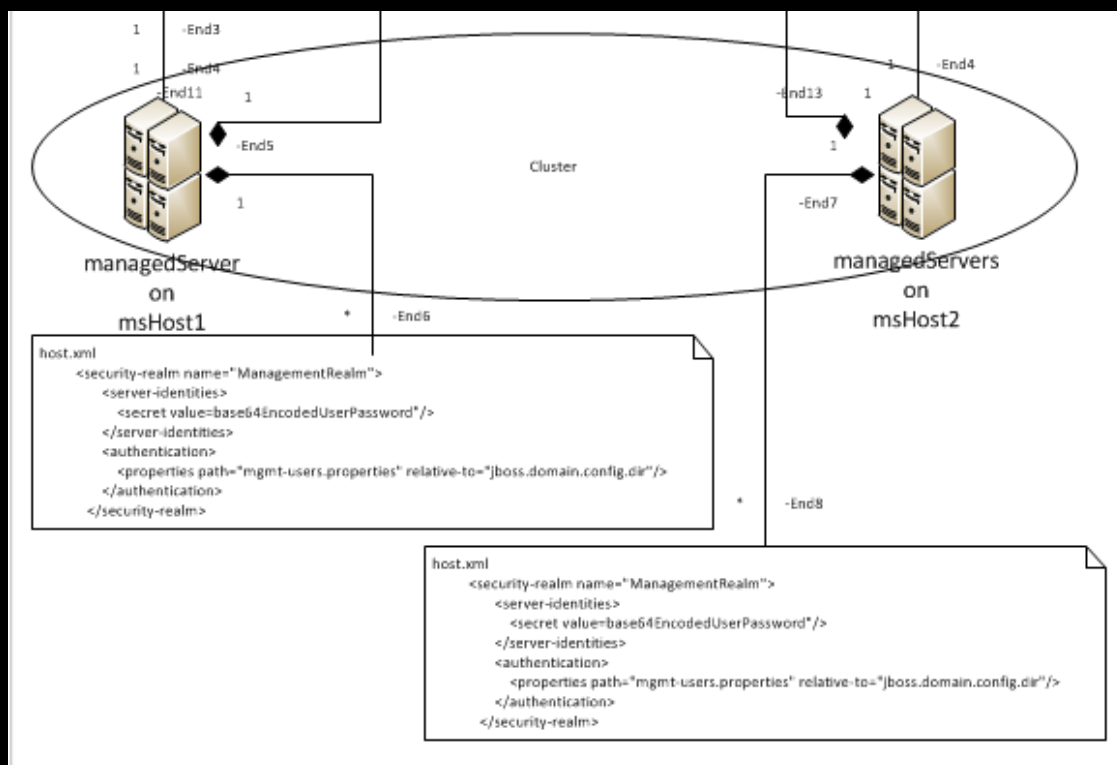
3.3 Establishing the Cluster

You should see the following when the domain becomes established.

```
[Host Controller] 01:54:16,645 INFO [org.jboss.as.domain] (slave-request-threads
- 1) JBAS010918: Registered remote slave host "host1", JBoss EAP 6.0.0.GA (AS 7.1.
2.Final-redhat-1)
```

4 Managed Servers and Clusters

Consider the following diagram. Each managed-server host needs a host.xml file that defines the servers that are to run at this host and which cluster / server-group from the domain-controller the server is to run under.



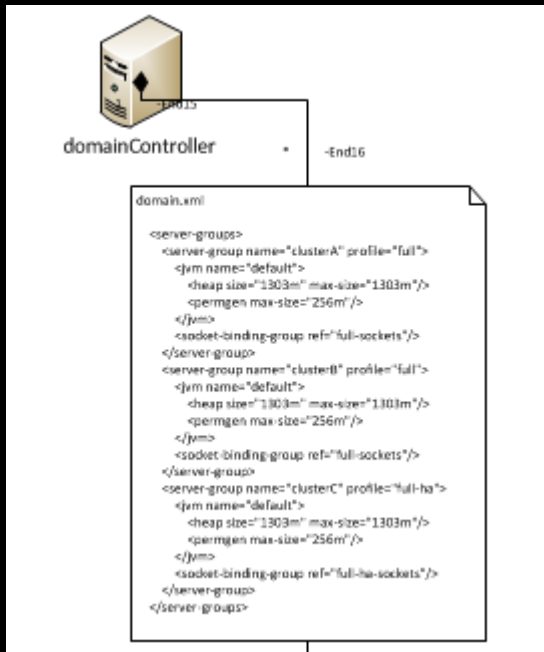
```
<servers>
  <server name="ms1" group="clusterA" auto-start="true">
    <!-- Remote JPDA debugging for a specific server -->
    <jvm name="default">
      <jvm-options>
        <option value="-Xrunjwp:transport=dt_socket,address=8787,server=y,suspend=n"/>
      </jvm-options>
    </jvm>
  </server>
  <server name="ms2" group="clusterA" auto-start="true">
    <!-- server-two avoids port conflicts by incrementing the ports in
    the default socket-group declared in the server-group -->
    <socket-bindings port-offset="150"/>
  </server>
  <server name="ms3" group="clusterC" auto-start="true">
    <!-- server-three avoids port conflicts by incrementing the ports in
    the default socket-group declared in the server-group -->
    <socket-bindings port-offset="250"/>
  </server>
</servers>
</host>
```

In the above example, three managed-servers exist. Ms1 is in cluster clusterA. Ms2 is in cluster clusterA. Ms3 is in cluster clusterC.

Hosts are capable of running managed-servers.

Managed-servers are the servers that do the work on behalf of the domain. Applications are normally deployed to managed-servers, grouped into clusters. These clusters are seen by the domain controller as administration units. This means that when you deploy your application you will do so to a cluster.

Consider the diagram below.



Clusters are registered for the domain within domain.xml at the Domain Controller.

Managed Servers are registered at each host via the host.xml at the Managed-Server Host.

4.1 Deploying Applications to a Cluster

Applications within a clustered domain will normally be deployed to a cluster.

This is simplified in the administration console of JBoss.

First you deploy the application.

Then you target it to a cluster, 'adding to group'.

The screenshot shows the JBoss Enterprise Application Platform 6.0 console. The main area is titled 'Server Group Deployments' under 'Deployment Content'. A modal window titled 'Select server groups' is open, displaying a table with the following data:

Server Group	Profile	Add to Group
clusterA	full	<input checked="" type="checkbox"/>
clusterB	full	<input type="checkbox"/>
clusterC	full-ha	<input type="checkbox"/>

Below the table, there is a checkbox that is checked and the text 'Enable activemq-rar-5.6-SNAPSHOT.rar'. The modal has 'Save' and 'Cancel' buttons at the bottom.

5 JBoss Profiles

JBoss will make use of profiles to enable multiple configurations at the administration console. A profile can be selected within the console as shown below:

The screenshot shows the JBoss Enterprise Application Platform 6.0 console with the 'Profiles' tab selected. A dropdown menu for 'Profile:' is open, showing the following options: 'full-ha', 'default', 'full', 'full-ha', and 'ha'. The 'full-ha' option is selected and circled in green. The main area displays 'Messaging Destinations' and 'JMS Messaging Provider' settings.

5.1 Profile Configuration in a Domain

To configure resources for the domain, the domain.xml file at the domain-controller must be edited. This can also be achieved via the administration console. Resources need to be added to the correct profile. For HA use of JBoss the full-HA profile should be used.

6 Apache HA Configuration

Apache can act as a load balancer for the JBoss clustered domain. Using the mod_cluster plugin load can be routed round-robin or weighted to a JBoss cluster.

6.1 Httpd.conf

Within HTTPD.CONF all the configuration for mod_cluster is completed.

6.2 Setting the listen port for Apache

The listen line, when added to <apache install dir>\httpd.conf enables apache to bind to port 80 and listen for traffic from client web browsers.

```
# Apache listen port
Listen 80
```

6.3 Registering mod_cluster with Apache

Add the following lines to httpd.conf to register mod_cluster with apache.

```
# JBoss - commented out for JBoss config
# LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
LoadModule proxy_connect_module modules/mod_proxy_connect.so
LoadModule cache_module modules/mod_cache.so
LoadModule suexec_module modules/mod_suexec.so
LoadModule disk_cache_module modules/mod_disk_cache.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule version_module modules/mod_version.so
```

The existing proxy balancer needs to be removed from Apache. Comment out the line.

```
# JBoss - commented out for JBoss config
# LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
```

6.4 Mod_Cluster configuration

Consider the following configuration information, within httpd.conf at the clustered apache web server.

```
# Apache listen port
Listen 80

# This Listen port is for the mod_cluster-manager, where you can see the status of mod_cluster.
# Port 10001 is not a reserved port, so this prevents problems with SELinux.
Listen 10.25.2.22:10001
# This directive only applies to Red Hat Enterprise Linux. It prevents the temporary
# files from being written to /etc/httpd/logs/ which is not an appropriate location.
MemManagerFile /var/cache/httpd

<VirtualHost Listen 10.25.2.22:10001>
  <Directory />
    # server hosts taking part in the cluster
    Order allow,deny
    Allow from 10.25.2.53
```

```

    Allow from 10.25.2.54
    # we dont want any servers than the ones we specify
    # taking part in the cluster
    #   Allow from all
    # we dont want to deny all, prevents allows from above if we do
    #   Deny from all
</Directory>

# This directive allows you to view mod_cluster status at URL http://10.25.2.22:10001/mod_cluster-manager
<Location /mod_cluster-manager>
    SetHandler mod_cluster-manager
    Order allow,deny
    # allow access for the mod cluster management console
    Allow from 192.168.
    Allow from 172.
    #   Deny from all
</Location>

KeepAliveTimeout 60
MaxKeepAliveRequests 0

ManagerBalancerName clusterA
AdvertiseFrequency 5

</VirtualHost>

```

6.4.1 Listen 10.25.2.22:10001

This setting This Listen port is for the mod_cluster-manager, where you can see the status of mod_cluster. It must be set to the public IPAddress for the cluster-manager machine. This then binds the manager to that address making it visible from the wider network.

6.4.2 MemManagerFile /var/cache/httpd

This directive only applies to Red Hat Enterprise Linux. It prevents the temporary files from being written to /etc/httpd/logs/ which is not an appropriate location.

6.4.3 VirtualHost Listen 10.25.2.22:10001

This directive instructs mod_cluster to create a virtual host listening on the public IPAddress for the machine. This is the address that requests from apache will get routed to. Eventually mod_cluster will load-balance to its registered JBoss nodes via this configuration.

6.4.3.1 Directory

This directive identifies the domain-hosts that managed-servers physically exist on.

6.4.3.1.1 Order allow,deny

This directive sets up allow and deny IPAddresses to restrict access to only the managed-server hosts you wish to take part in the mod_cluster load-balanced cluster.

6.4.4 Location /mod_cluster-manager

This directive allows you to view mod_cluster status at URL http://<publicIPAddress>:10001/mod_cluster-manager.

6.4.5 ManagerBalancerName clusterA

This directive instructs mod_cluster to target http requests coming into Apache to a JBoss cluster called clusterA

Mod_cluster Administration Page

You can access the mod_cluster console at: http://<mod_cluster IP : mod_cluster port>/mod_cluster-manager

7 Sticky sessions

Sticky sessions can be enabled or disabled with the following configuration within the modcluster subsystem within domain.xml file.

```
<subsystem xmlns="urn:jboss:domain:modcluster:1.1">
  <mod-cluster-config advertise-socket="modcluster" connector="ajp" sticky-session="true">
    <dynamic-load-provider>
      <load-metric type="busyness"/>
    </dynamic-load-provider>
  </mod-cluster-config>
</subsystem>
```

8 Testing the Cluster

To test the new cluster, a browser needs to be pointed to the Apache IP address and the listen port specified in [6.2 Setting the listen port for Apache](#).