Installing and Setting up mod_cluser

Jim Tyrrell
February 18, 2011

This white paper covers the setting up of mod_cluster, the Apache Http Server, JBoss Enterprise Application Platform (EAP), on a single computer. This will give you the ability to validate you have all the components working before expanding out or adding in the complexities of the network, firewalls, and more. These issues are far outside of a white paper to quickly get you up and running. At the end of this white paper you will have EAP configured to replicate data across nodes in a cluster, have mod_cluster setup and installed in the Apache HTTP Server, and have the Apache HTTP Server installed and working. At the end of all of these steps SELinux can also be enabled in enforcing mode to monitor and secure your Red Hat Enterprise Linux instance.

Contents

1. What is mod_cluster

mod_cluster is used to load balance using the Apache Http Server web requests to any number of underlying EAP instances.

2. Why would you use mod_cluster

mod_cluster is used to spread out the load of dynamic application web requests across several different instances of EAP. This allows you to spread out the load across many different JBoss EAP instances providing for failover. Using mod_cluster also enables higher numbers of end users served as it can spread out the load across many different instances. By utilizing several JBoss EAP instances you can also facilitate rolling application upgrades to maintain greater application instance uptime. Mod_cluster is superior to other previous clustering/load balancing technologies as it supports several things. These include dynamic registration of new EAP instance once they are configured to reach out to mod_cluster running in the Apache Http Server. The load is dynamically routed/load balanced across all the nodes via a bytes sent/received algorithm. Most importantly mod_cluster is the way forward for JBoss to deliver software load balancing.

3. Current Limitations of mod_cluster

Currently mod_cluster is only supported in the context of a Enterprise Web Service (EWS) subscription, and requires higher then the Apache Http Server 2.2.8 if you are going to use it without the context of the EWS subscription, if you do this you will be running HTTPD and mod_cluster totally unsupported. At this point this is a gap that is currently being worked by Red Hat to align these version numbers.

4. Installation and Setup Steps

4.1. Introduction

Overview

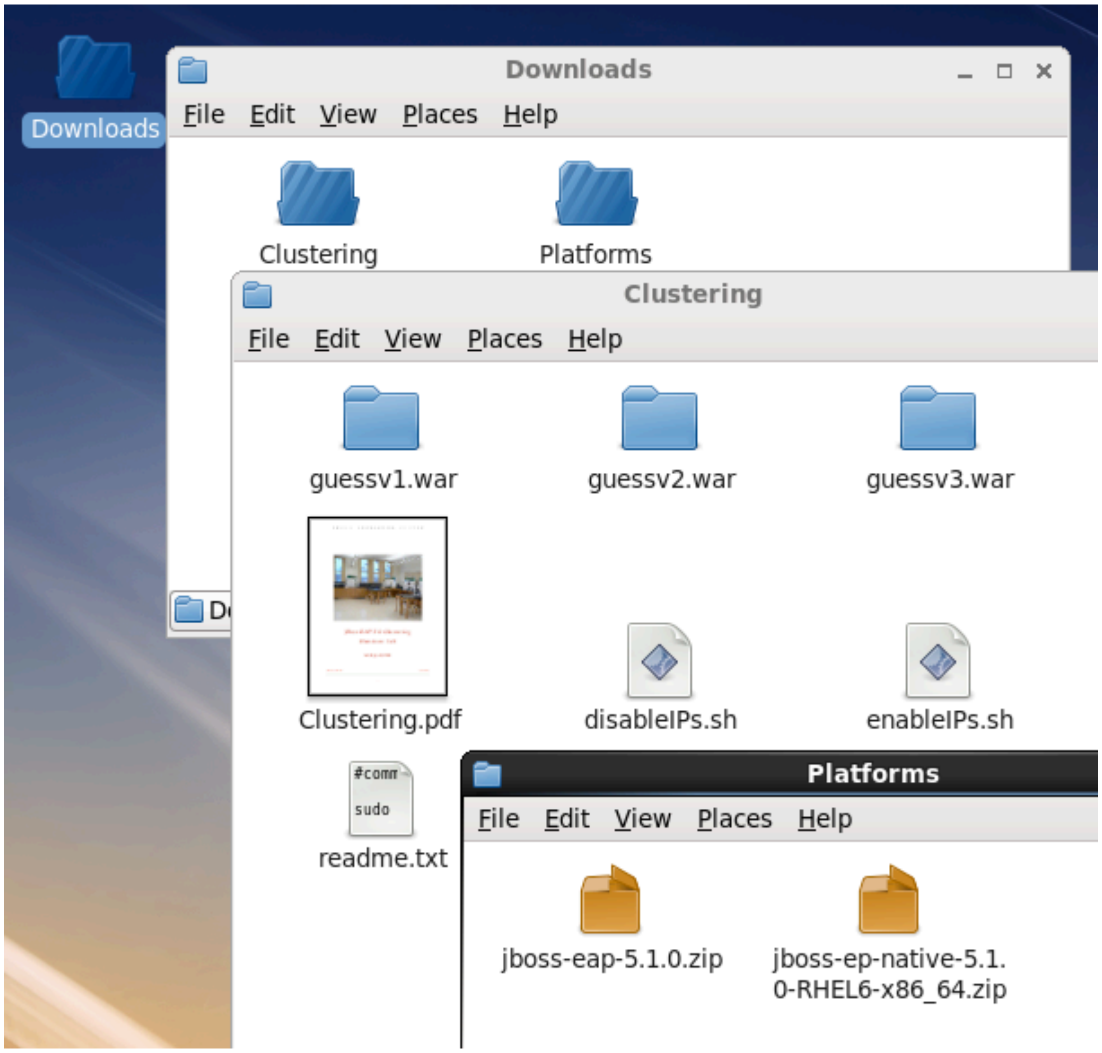
Clustering with JBoss EAPSELinux is a simple way to give your website more availability and capacity for your end users. This is done simply be creating/installing several instances of JBoss and then load balancing them via some sort of proxy. In this white paper we will use the Apache Http Server as the port 80 web server. We will then install mod_cluster to dynamically discover the nodes that are able to handle the load. An appendix will walk you through using/configuring SELinux, however, the main flow of the white paper will turn off SELinux. The scope of SELinux is far outside of the scope of this white paper, but is included via the appendix to get you thinking about SELinux implications. Once everything is installed we will explore turning on and off various instances, application deployment, rolling updates etc.

This white paper is written specifically for Red Hat EnterpriSELinux 6.0 (RHEL), and should work with minor path changes for future versions of RHEL. For the JBoss components, this white paper was created with EAP 5.1 and EAP 5.1 native components. As future versions of these are released this white paper should work with minor path changes reflecting those updated binaries. For other operating supported operating systems the below steps should essentially be unchanged.

Included FIles

If you are getting this white paper as part of a Red Hat delivered lab, all of the required files should be on the Desktop in various folders. If you are downloading this the recommended structure ${Desktop}/Downloads/Platforms for your two JBoss files Application Platform 5.1.0 Binary, and JBoss EAP 5.1. Native Components for RHEL 6, x86_64. It would be recommended to have this guide in a ${Desktop}/Downloads/Clustering along with the sample war files, two scripts for enabling IP Address, and a readme.txt that has the various commands available for cutting and pasting.

You can see this structure below:



System Expectations

With this white paper it is expected that you have a computer with RHEL 6. It is expected you will have the environment PATH set to include a JDK 6.0 to use for these white papers. It is also a good idea to have JAVA_HOME set to your JDK that you plan on using. If you are in a Red Hat provided lab these settings are already done for you. Please make sure you do this before running any of the white papers. Two examples of what these settings might look like is below:

```
PATH=${Some Path}/jdk1.6.0_20/bin:${Some Path}/ant/apache-ant-1.8.1: ${More
Path Info}
JAVA_HOME=${Some Path}jdk1.6.0_20
```

To verify that this is correct you will have to look at these values on your system One simple way to check the JDK version that you have is to run:

```
java -version
```

to see which one is in your path, and it should be a JDK 6 version to run this white paper.
Please note if you are using your own computer having an existing CLASSPATH environment variable set may cause odd issues with jar class loading, it is recommended to have this empty and not set. Please make sure to back up this value for when the whitepaper is over. You are welcome to not do this, however weird things may happen when you are running through the white papers if you do not have an empty CLASSPATH variable.

What is Expected of You
This white paper is intended for self directed study, and is being delivered as a courtesy to our customers, If you are having issues, for other users forums should be available to assist you with any questions. Please know that all care was made in creating this user guide, but all screen shots and steps along the way might be off by just a little so please be patient with any issues, and feel free to raise them in the forums, or at http://jira.jboss.com/whitepapers

4.2. Check List

Check List
Sometimes you just need a quick list of the steps to do something, as it is something you do every so often, but you are not sure of all of the steps. If you need complete handholding, that is what the following lab chapters deliver, however if you know the gist of what you need to do, this check list is provided to help you get going.

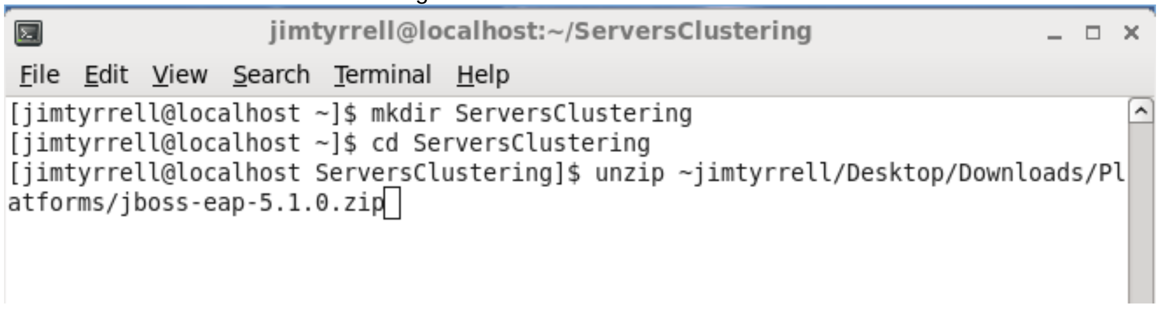
1. Get Required Software
2. Unzip EAP
3. Unzip Native Components
4. Copy all to node1
5. Make configuration file changes to node1
6. Make sure node1 starts
7. Copy node1 to node2, node3, node4
8. Make sure apache starts
9. Install mod_cluster components
10. Make sure apache still starts
11. Verify installation

4.3. Install and Configure EAP

Get the File
In the ${USER_HOME}Downloads/Platforms directory you will find the EAP installer, it platform agnostic and it should look something like this:
```
jboss-eap-5.1.0.zip
```

Just Unzip and Go
Installing the EAP is very very simple, and has the following high level steps:
Create a ServersClustering directory in the user home directory, make this unique
Unzip the contents of the file above into that directory

```
mkdir ~student/ServersClustering
cd ~student/ServersClustering
unzip ~student/Desktop/Downloads/Platforms/jboss-eap-5.1.0.zip
```

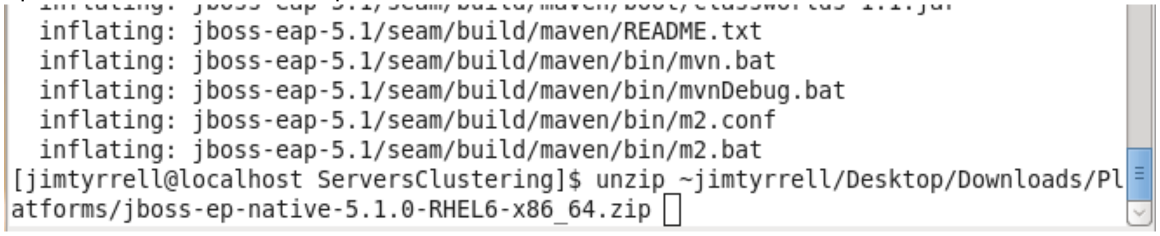
Your command/s should look something like this:

jimtyrrell@localhost:~/ServersClustering
File Edit View Search Terminal Help
```
[jimtyrrell@localhost ~]$ mkdir ServersClustering
[jimtyrrell@localhost ~]$ cd ServersClustering
[jimtyrrell@localhost ServersClustering]$ unzip ~jimtyrrell/Desktop/Downloads/Pl
atforms/jboss-eap-5.1.0.zip
```

Make sure you hit enter after the unzip command and wait for it to finish.
That is it, now JBoss Enterprise Application Platform is installed and ready to use. In a few seconds you have installed a full JEE container.

Apache Portable Runtime

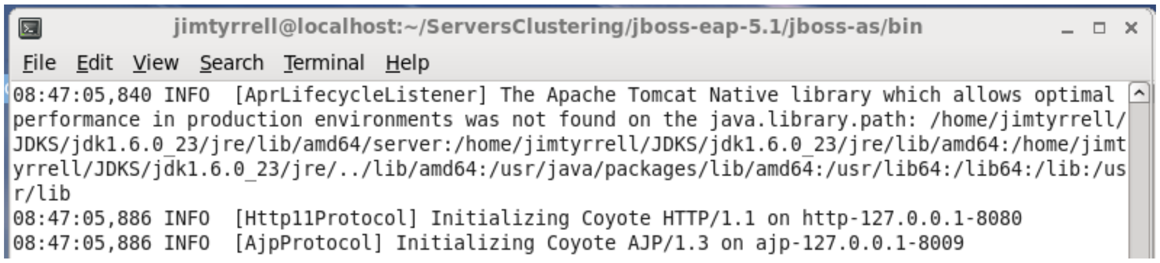
The next step is to install the APR or Apache Portable Runtime into the Container. This is not specifically needed for setting up clustering, but performance and getting the most out of your available resources is something customers typically need/want when setting up clustering. In order to install the APR, you just need to unzip the included file from the Clustering folder.
The command to do this is:
```
unzip ~student/Desktop/Downloads/Platforms/jboss-ep-native-5.1.0-RHEL6-
x86_64.zip
```

And it would look like this, the unzip is finished from the prior step, and you are now ready to unzip the Apache Portable Runtime Components:



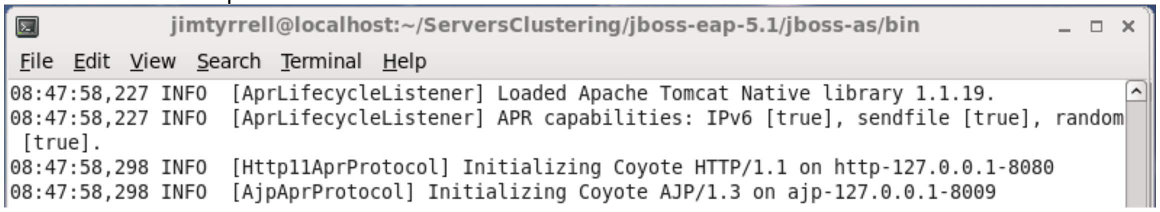
```
  inflating: jboss-eap-5.1/seam/build/maven/README.txt
  inflating: jboss-eap-5.1/seam/build/maven/bin/mvn.bat
  inflating: jboss-eap-5.1/seam/build/maven/bin/mvnDebug.bat
  inflating: jboss-eap-5.1/seam/build/maven/bin/m2.conf
  inflating: jboss-eap-5.1/seam/build/maven/bin/m2.bat
[jimtyrrell@localhost ServersClustering]$ unzip ~jimtyrrell/Desktop/Downloads/Pl
atforms/jboss-ep-native-5.1.0-RHEL6-x86_64.zip
```

Now you have the first step for the install completed. You may ask yourself how do you know if you have the APR installed. Well at startup time the server will output a message letting you know, without the APR looks like this:


jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/bin
File Edit View Search Terminal Help

```
08:47:05,840 INFO  [AprLifecycleListener] The Apache Tomcat Native library which allows optimal
performance in production environments was not found on the java.library.path: /home/jimtyrrell/
JDKS/jdk1.6.0_23/jre/lib/amd64/server:/home/jimtyrrell/JDKS/jdk1.6.0_23/jre/lib/amd64:/home/jimt
yrrell/JDKS/jdk1.6.0_23/jre/../lib/amd64:/usr/java/packages/lib/amd64:/usr/lib64:/lib64:/lib:/us
r/lib
08:47:05,886 INFO  [Http11Protocol] Initializing Coyote HTTP/1.1 on http-127.0.0.1-8080
08:47:05,886 INFO  [AjpProtocol] Initializing Coyote AJP/1.3 on ajp-127.0.0.1-8009
```

With the APR the output will look like this:


jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/bin
File Edit View Search Terminal Help

```
08:47:58,227 INFO  [AprLifecycleListener] Loaded Apache Tomcat Native library 1.1.19.
08:47:58,227 INFO  [AprLifecycleListener] APR capabilities: IPv6 [true], sendfile [true], random
 [true].
08:47:58,298 INFO  [Http11AprProtocol] Initializing Coyote HTTP/1.1 on http-127.0.0.1-8080
08:47:58,298 INFO  [AjpAprProtocol] Initializing Coyote AJP/1.3 on ajp-127.0.0.1-8009
```

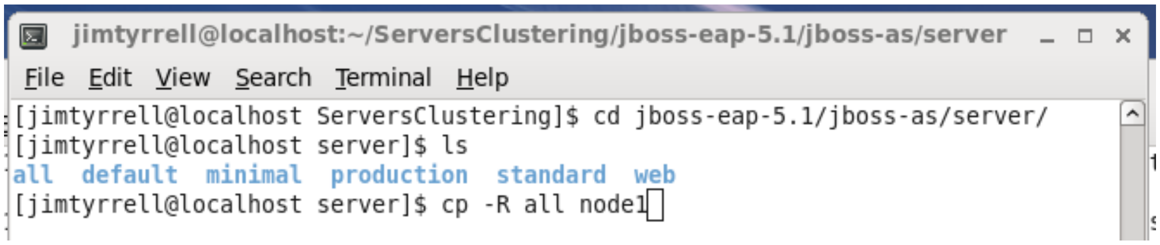
Scrolling up you can view that the Apache Portable Runtime (APR) was loaded successfully. It should also be noted that the APR is specific to each Operating System, and you would have to get the correct one.

That is it for unzipping files and having all the files you will need on your local file system as required for the next steps in the lab. If you are confused or something does not feel right, please feel free to raise your hand.

Copy a Server Config

All changes we plan to make to our server configuration will be a derivative of the all configuration that ships with JBoss. The all configuration is left unchanged so that you have a known good and working configuration that you can always use to copy to make a new instance if something goes sideways. If you are not familiar JBoss makes it very easy to create specialized configurations, and ships with several out of the box including: all, production, default, etc. We will simply change to the correct directory and execute a copy.
```
cd ~student/ServersClustering/jboss-eap-5.1/jboss-as/server/
cp -R all node1
```

As shown below:



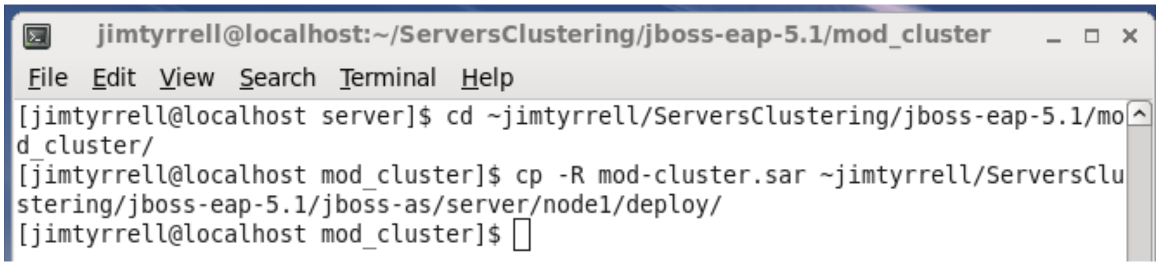
```
[jimtyrrell@localhost ServersClustering]$ cd jboss-eap-5.1/jboss-as/server/
[jimtyrrell@localhost server]$ ls
all  default  minimal  production  standard  web
[jimtyrrell@localhost server]$ cp -R all node1
```

Configuration File Changes

JBoss is just a simple set of files that can be changed. Once these changes are made you can zip up or copy that configuration and make it available as golden image. We will use that feature a little later on. First we need to copy the mod_cluster.sar file which was delivered in the main jboss-eap-5.1.0.zip in the mod_cluster directory, this sar needs to be copied as it is a value added feature not delivered with the core of the product. You have a choice with JBoss, and mod_cluster is the recommended software load balancer supported by JBoss. This is not shipped by default to save space in JVM memory, threads, etc to give you as light a JEE container as possible.
To start you need to copy the sar as shown:

```
cd ~student/ServerClustering/jboss-eap-5.0/mod_cluster
cp -R mod-cluster.sar ~student/ServersClustering/jboss-eap-5.1/jboss-as/server/
node1/deploy
```

As shown:

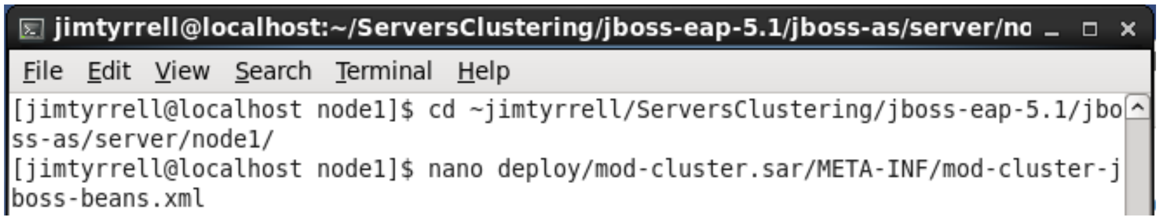


```
[jimtyrrell@localhost server]$ cd ~jimtyrrell/ServersClustering/jboss-eap-5.1/mo
d_cluster/
[jimtyrrell@localhost mod_cluster]$ cp -R mod-cluster.sar ~jimtyrrell/ServersClu
stering/jboss-eap-5.1/jboss-as/server/node1/deploy/
[jimtyrrell@localhost mod_cluster]$
```

Next we need to edit the mod-cluster-jboss-beans.xml file, you can use your favorite editor for this vi or nano. If you know vi your all set, if not nano is right up your alley.

```
cd ~student/ServerClustering/jboss-eap-5.0/jboss-as/server/node1
nano deploy/mod-cluster.sar/META-INF/mod-cluster-jboss-beans.xml
```

As shown:

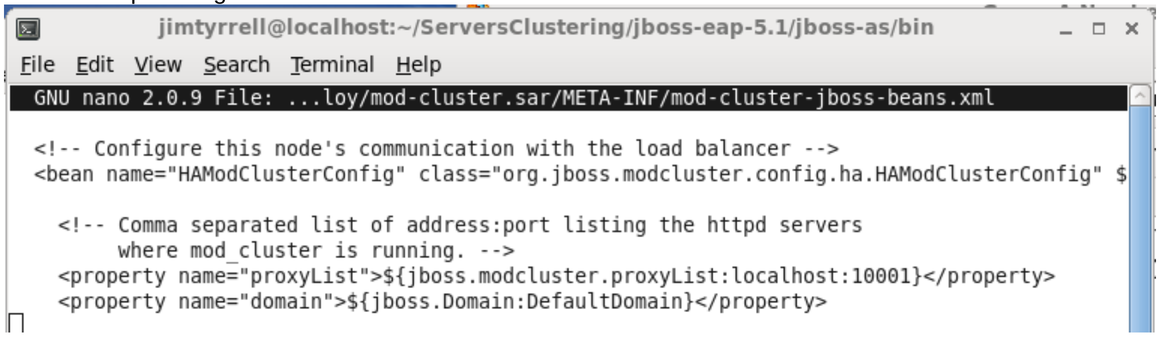


```
[jimtyrrell@localhost node1]$ cd ~jimtyrrell/ServersClustering/jboss-eap-5.1/jbo
ss-as/server/node1/
[jimtyrrell@localhost node1]$ nano deploy/mod-cluster.sar/META-INF/mod-cluster-j
boss-beans.xml
```

Next you need to edit and/or add two lines in the file, scroll down until you find the entry for ...“proxyList”.. Those two lines will look like this, note line wrapping in this document show these as three lines, in the config file they need to be two, and respect the normal rules of XML markup in regards to line wraps and whitespace:

```
<property name="proxyList">${jboss.modcluster.proxyList:localhost:10001}</
property>         <property name="domain">${jboss.Domain:DefaultDomain}</
property>
```

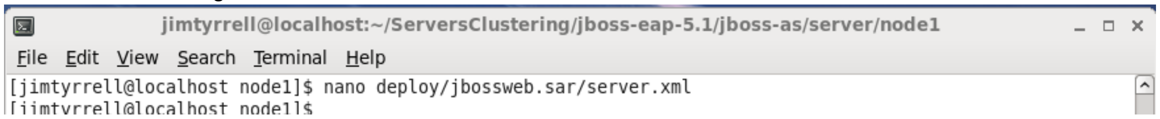
So it ends up looking like this:



```
jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/bin
File  Edit  View  Search  Terminal  Help
  GNU nano 2.0.9 File: ...loy/mod-cluster.sar/META-INF/mod-cluster-jboss-beans.xml

  <!-- Configure this node's communication with the load balancer -->
  <bean name="HAModClusterConfig" class="org.jboss.modcluster.config.ha.HAModClusterConfig" $

    <!-- Comma separated list of address:port listing the httpd servers
         where mod_cluster is running. -->
    <property name="proxyList">${jboss.modcluster.proxyList:localhost:10001}</property>
    <property name="domain">${jboss.Domain:DefaultDomain}</property>
```

Make sure you save the file.
Next you will have to edit the server.xml file in the jbossweb.sar file, that command will look something like this, again use which ever editor you are most comfortable with.
nano deploy/jbossweb.sar/server.xml

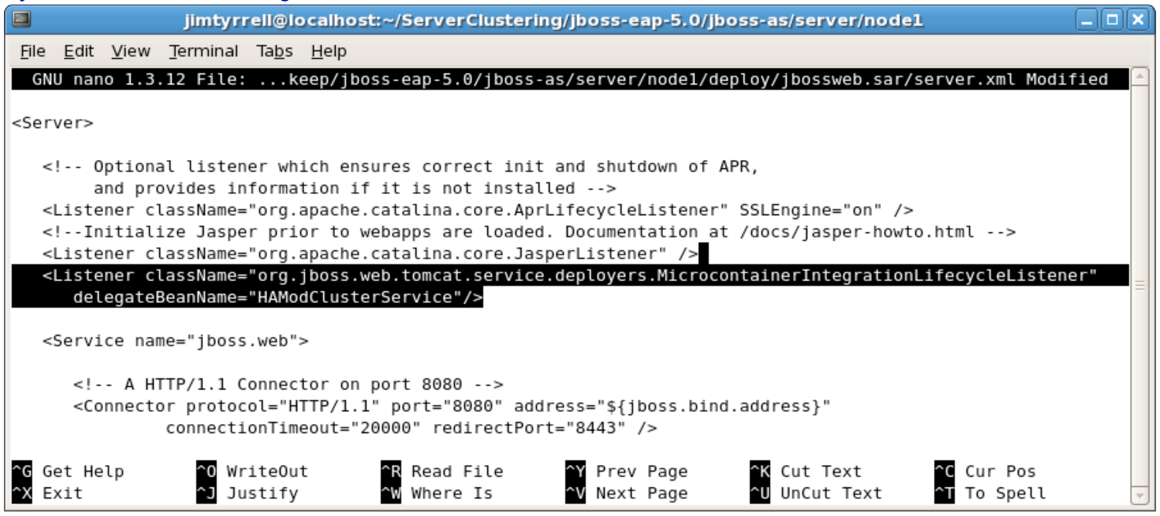
It will look something like this:



```
jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/server/node1
File  Edit  View  Search  Terminal  Help
[jimtyrrell@localhost node1]$ nano deploy/jbossweb.sar/server.xml
[jimtyrrell@localhost node1]$
```

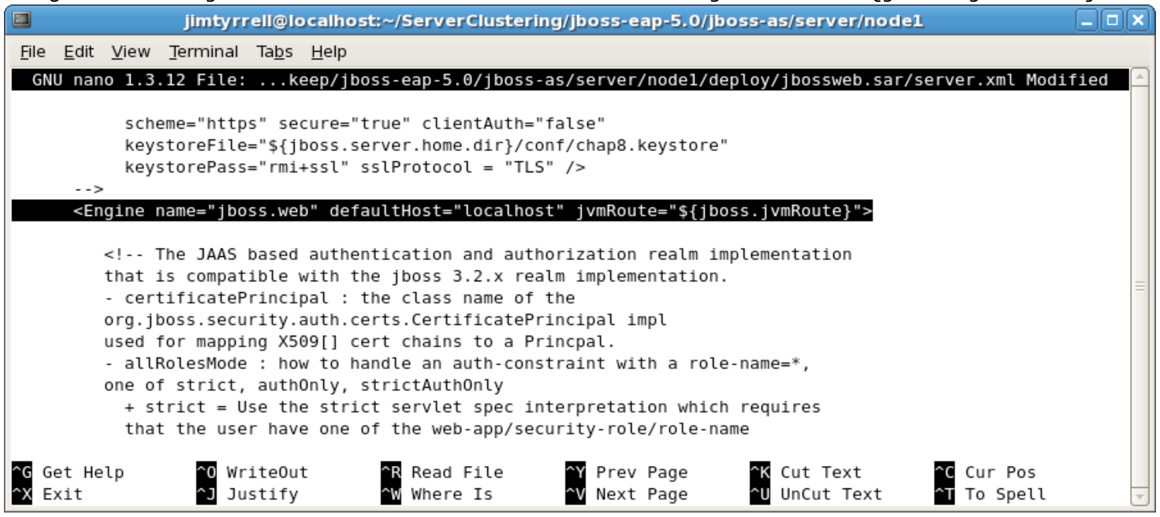
You then need to add/edit two lines in this file, the first is to add a new Listener near the other listeners at the top of the file and the second is to add a jvmRoute to the existing Engine Component. The jvmRoute is used by the load balancer to do or support sticky sessions, which is this idea that once a request is mapped to a specific EAP instance, it will map future requests to that node. This is done for efficiency.

```
<Listener
className="org.jboss.web.tomcat.service.deployers.MicrocontainerIntegrationLifecycleListener" delegateBeanName="HAModClusterService"/>
```



```
jimtyrrell@localhost:~/ServerClustering/jboss-eap-5.0/jboss-as/server/node1
File  Edit  View  Terminal  Tabs  Help
  GNU nano 1.3.12 File: ...keep/jboss-eap-5.0/jboss-as/server/node1/deploy/jbossweb.sar/server.xml Modified

<Server>

   <!-- Optional listener which ensures correct init and shutdown of APR,
        and provides information if it is not installed -->
   <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
   <!--Initialize Jasper prior to webapps are loaded. Documentation at /docs/jasper-howto.html -->
   <Listener className="org.apache.catalina.core.JasperListener" />
   <Listener className="org.jboss.web.tomcat.service.deployers.MicrocontainerIntegrationLifecycleListener"
      delegateBeanName="HAModClusterService"/>

   <Service name="jboss.web">

      <!-- A HTTP/1.1 Connector on port 8080 -->
      <Connector protocol="HTTP/1.1" port="8080" address="${jboss.bind.address}"
               connectionTimeout="20000" redirectPort="8443" />

^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

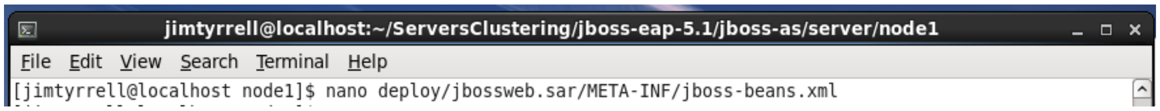
```
<Engine name="jboss.web" defaultHost="localhost" jvmRoute="${jboss.jvmRoute}">
```


jimtyrrell@localhost:~/ServerClustering/jboss-eap-5.0/jboss-as/server/node1
File Edit View Terminal Tabs Help

```
  GNU nano 1.3.12 File: ...keep/jboss-eap-5.0/jboss-as/server/node1/deploy/jbossweb.sar/server.xml Modified

           scheme="https" secure="true" clientAuth="false"
           keystoreFile="${jboss.server.home.dir}/conf/chap8.keystore"
           keystorePass="rmi+ssl" sslProtocol = "TLS" />
      -->
      <Engine name="jboss.web" defaultHost="localhost" jvmRoute="${jboss.jvmRoute}">

         <!-- The JAAS based authentication and authorization realm implementation
         that is compatible with the jboss 3.2.x realm implementation.
         - certificatePrincipal : the class name of the
         org.jboss.security.auth.certs.CertificatePrincipal impl
         used for mapping X509[] cert chains to a Princpal.
         - allRolesMode : how to handle an auth-constraint with a role-name=*,
         one of strict, authOnly, strictAuthOnly
           + strict = Use the strict servlet spec interpretation which requires
           that the user have one of the web-app/security-role/role-name

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell
```

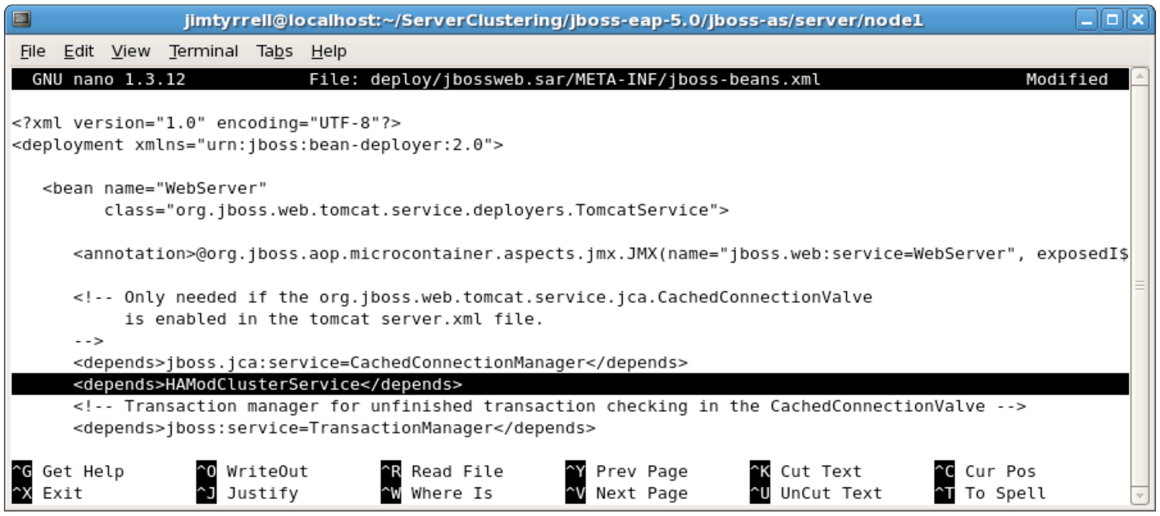
After making those two changes make sure you save out the file.
The next step is to edit the jboss-beans.xml file:

```
nano deploy/jbossweb.sar/META-INF/jboss-beans.xml
```


jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/server/node1
File Edit View Search Terminal Help

```
[jimtyrrell@localhost node1]$ nano deploy/jbossweb.sar/META-INF/jboss-beans.xml
```

```
and add in this entry:
<depends>HAModClusterService</depends>
It should look like this:
```


jimtyrrell@localhost:~/ServerClustering/jboss-eap-5.0/jboss-as/server/node1
File Edit View Terminal Tabs Help

```
  GNU nano 1.3.12              File: deploy/jbossweb.sar/META-INF/jboss-beans.xml                  Modified

<?xml version="1.0" encoding="UTF-8"?>
<deployment xmlns="urn:jboss:bean-deployer:2.0">

   <bean name="WebServer"
         class="org.jboss.web.tomcat.service.deployers.TomcatService">

      <annotation>@org.jboss.aop.microcontainer.aspects.jmx.JMX(name="jboss.web:service=WebServer", exposedI$

      <!-- Only needed if the org.jboss.web.tomcat.service.jca.CachedConnectionValve
           is enabled in the tomcat server.xml file.
      -->
      <depends>jboss.jca:service=CachedConnectionManager</depends>
      <depends>HAModClusterService</depends>
      <!-- Transaction manager for unfinished transaction checking in the CachedConnectionValve -->
      <depends>jboss:service=TransactionManager</depends>

^G Get Help      ^O WriteOut      ^R Read File     ^Y Prev Page     ^K Cut Text      ^C Cur Pos
^X Exit          ^J Justify       ^W Where Is      ^V Next Page     ^U UnCut Text    ^T To Spell
```

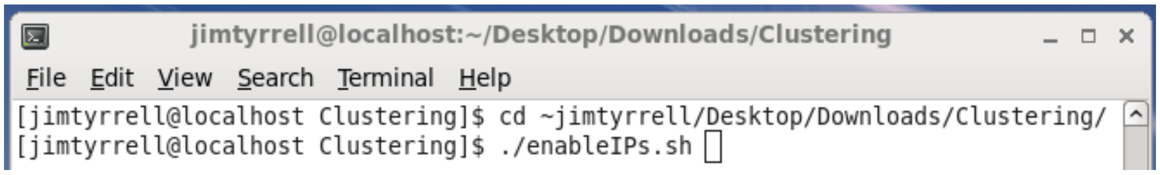
Make sure you save the file.

Moment of Truth

Now it is time to see if all of the changes you made were successful. In other words can you start up the JBoss instance. Before you do that you need to create a few ip address and multicast address on your local box. You can open up the readme.txt in the ~student/Desktop/Downloads/Clustering/ directory and you will find several commands you need to run to turn on these ip address. Or even easier you can just run the command:

```
./enableIPs.sh
```

as shown:


jimtyrrell@localhost:~/Desktop/Downloads/Clustering
File Edit View Search Terminal Help

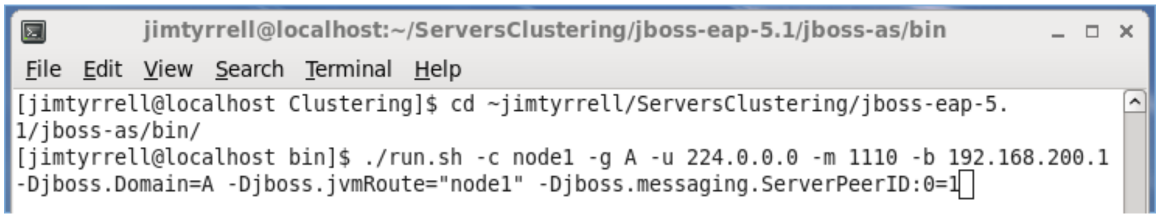
```
[jimtyrrell@localhost Clustering]$ cd ~jimtyrrell/Desktop/Downloads/Clustering/
[jimtyrrell@localhost Clustering]$ ./enableIPs.sh
```

It will ask you for your password as you should be in a lab added to the sudoers file. If you are on your own system you will have to ask your administrator how to turn on these IP Address.
The next step is to change to the bin directory and start up node1. The readme has lines for starting each of the four nodes we will start eventually. Grab the first one and lets start up the server:

```
cd ~student/ServerClustering/jboss-eap-5.0/jboss-as/bin
./run.sh -c node1 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.1 -Djboss.Domain=A -
Djboss.jvmRoute="node1"
```

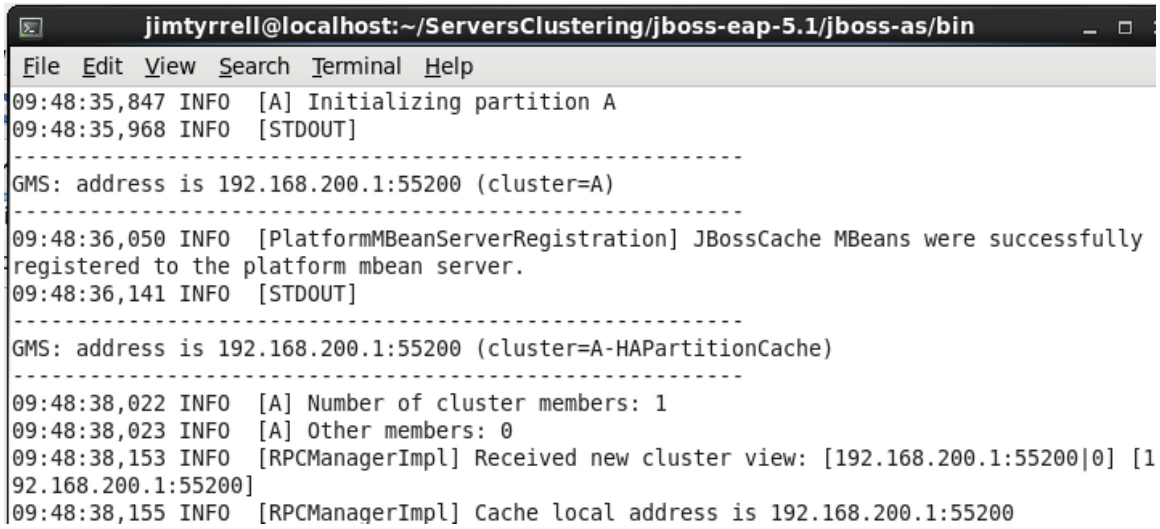
-Djboss.messaging.ServerPeerID:0=1
You may be wondering about all those options above, lets break them down:
-c is for configuration
-g is for group name in clustering
-u is for unicast address
-m is for multicast port address
-b is for IP Address
-Djboss.Domain is a unique domain used in mod_cluster
-Djboss.jvmRoute is used to uniquely identify a worker node
-Djboss.messagin.ServerPeerID:0 is used to inject a unique number into JBoss Messaging

As shown:


jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/bin
File Edit View Search Terminal Help

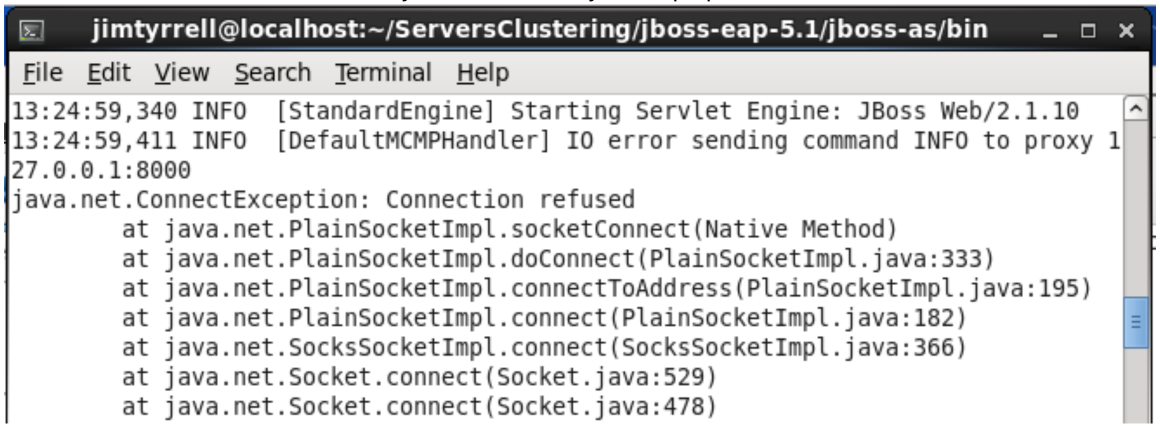
```
[jimtyrrell@localhost Clustering]$ cd ~jimtyrrell/ServersClustering/jboss-eap-5.
1/jboss-as/bin/
[jimtyrrell@localhost bin]$ ./run.sh -c node1 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.1
-Djboss.Domain=A -Djboss.jvmRoute="node1" -Djboss.messaging.ServerPeerID:0=1
```

Hit enter and wait for the server to come up. Remember to scroll up and see that the APR was installed, was it?
A few things to note, you should see that this server is a member of a cluster of one as shown:


jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/bin
File Edit View Search Terminal Help

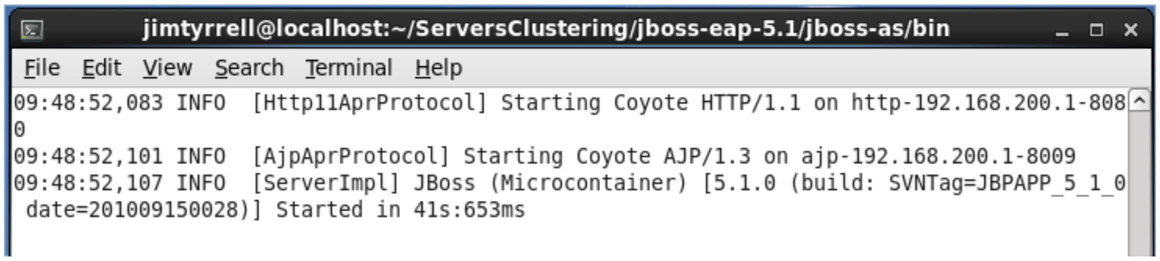
```
09:48:35,847 INFO  [A] Initializing partition A
09:48:35,968 INFO  [STDOUT]
---------------------------------------------------------
GMS: address is 192.168.200.1:55200 (cluster=A)
---------------------------------------------------------
09:48:36,050 INFO  [PlatformMBeanServerRegistration] JBossCache MBeans were successfully
registered to the platform mbean server.
09:48:36,141 INFO  [STDOUT]
---------------------------------------------------------
GMS: address is 192.168.200.1:55200 (cluster=A-HAPartitionCache)
---------------------------------------------------------
09:48:38,022 INFO  [A] Number of cluster members: 1
09:48:38,023 INFO  [A] Other members: 0
09:48:38,153 INFO  [RPCManagerImpl] Received new cluster view: [192.168.200.1:55200|0] [1
92.168.200.1:55200]
09:48:38,155 INFO  [RPCManagerImpl] Cache local address is 192.168.200.1:55200
```

You will see an error like this, it is okay as we have not yet setup apache:


jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/bin
File Edit View Search Terminal Help

```
13:24:59,340 INFO  [StandardEngine] Starting Servlet Engine: JBoss Web/2.1.10
13:24:59,411 INFO  [DefaultMCMPHandler] IO error sending command INFO to proxy 1
27.0.0.1:8000
java.net.ConnectException: Connection refused
        at java.net.PlainSocketImpl.socketConnect(Native Method)
        at java.net.PlainSocketImpl.doConnect(PlainSocketImpl.java:333)
        at java.net.PlainSocketImpl.connectToAddress(PlainSocketImpl.java:195)
        at java.net.PlainSocketImpl.connect(PlainSocketImpl.java:182)
        at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:366)
        at java.net.Socket.connect(Socket.java:529)
        at java.net.Socket.connect(Socket.java:478)
```

The above error is okay, any other errors are not acceptable, if you have any other errors please raise your hand.
When you see this message the server has finished starting:
......Started in .....
As shown:

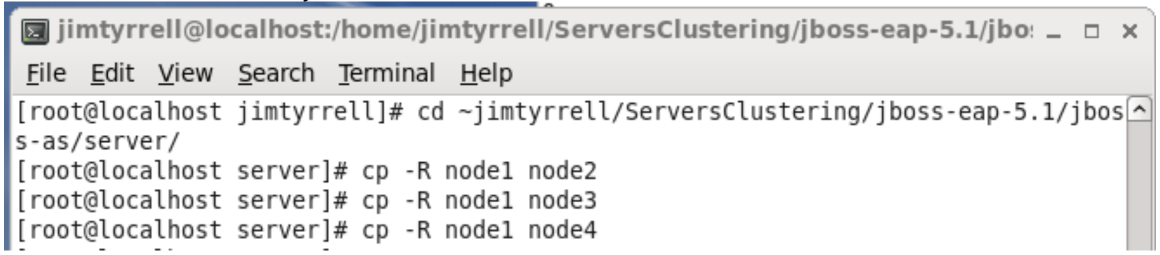

jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jboss-as/bin
File Edit View Search Terminal Help

```
09:48:52,083 INFO  [Http11AprProtocol] Starting Coyote HTTP/1.1 on http-192.168.200.1-808
0
09:48:52,101 INFO  [AjpAprProtocol] Starting Coyote AJP/1.3 on ajp-192.168.200.1-8009
09:48:52,107 INFO  [ServerImpl] JBoss (Microcontainer) [5.1.0 (build: SVNTag=JBPAPP_5_1_0
 date=201009150028)] Started in 41s:653ms
```

You have now started node1, congratulations.If you remember how you copied “all” into “node1” earlier, if you have this server running correctly, it is time to do that to create node2, 3, and 4.
Simple cd to the server directory and execute:

```
cd ~student/ServersClustering/jboss-eap-5.1/jboss-as/server/
cp -R node1 node2
cp -R node1 node3
cp -R node1 node4
```

It should look like this when you are done:

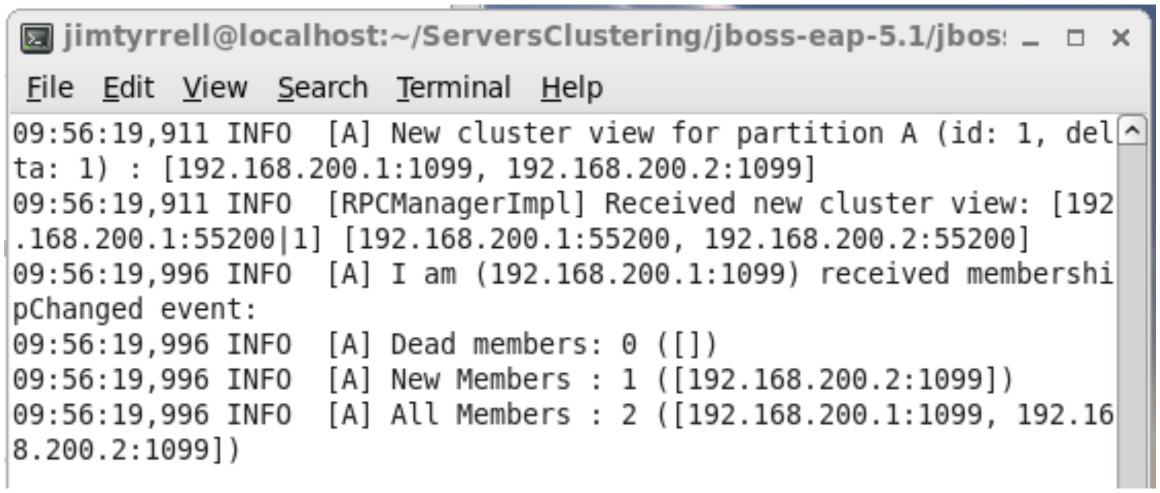

jimtyrrell@localhost:/home/jimtyrrell/ServersClustering/jboss-eap-5.1/jbo:
File Edit View Search Terminal Help

```
[root@localhost jimtyrrell]# cd ~jimtyrrell/ServersClustering/jboss-eap-5.1/jbos
s-as/server/
[root@localhost server]# cp -R node1 node2
[root@localhost server]# cp -R node1 node3
[root@localhost server]# cp -R node1 node4
```

Start up node2 using the second startup command from the readme file:

```
cd ~student/ServerClustering/jboss-eap-5.0/jboss-as/bin
./run.sh -c node2 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.2 -Djboss.Domain=A -
Djboss.jvmRoute="node2" -Djboss.messaging.ServerPeerID:0=2
```

Make sure it starts without any errors, other then the one noted above. Also note that it joined a cluster, with messages in both windows that looked like this, your first server you started will look like this:


jimtyrrell@localhost:~/ServersClustering/jboss-eap-5.1/jbos
File Edit View Search Terminal Help

```
09:56:19,911 INFO  [A] New cluster view for partition A (id: 1, del
ta: 1) : [192.168.200.1:1099, 192.168.200.2:1099]
09:56:19,911 INFO  [RPCManagerImpl] Received new cluster view: [192
.168.200.1:55200|1] [192.168.200.1:55200, 192.168.200.2:55200]
09:56:19,996 INFO  [A] I am (192.168.200.1:1099) received membershi
pChanged event:
09:56:19,996 INFO  [A] Dead members: 0 ([])
09:56:19,996 INFO  [A] New Members : 1 ([192.168.200.2:1099])
09:56:19,996 INFO  [A] All Members : 2 ([192.168.200.1:1099, 192.16
8.200.2:1099])
```

As we shut down and play with these instances these messages will let you know when a machine has left the cluster.

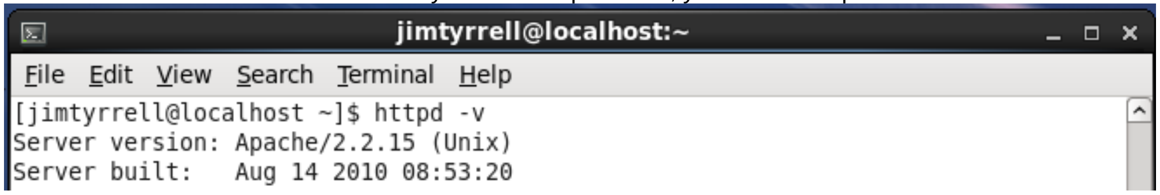
Congratulations you now have two servers that are clustered together, but you do not have any load balancing for web content going on between them. That we will cover in the next lab.

4.4. Install Apache HTTPD

Install Apache HTTPD

The Apache Http Server should already be installed and available on your RHEL 6 instance if this is a Red Hat run lab. If not make sure you have either RHEL 6 with the Apache Http Server installed, a supported Enterprise Web Server for a fully supported configuration for other operating systems, or at miniunum a the Apache Http Server 2.2.8 or higher installed. You will also need the appropriate native components downloaded and available, in this lab we have already done that in Lab Number 1.

At the command line to view the currently installed httpd server, you can run httpd -v as shown:

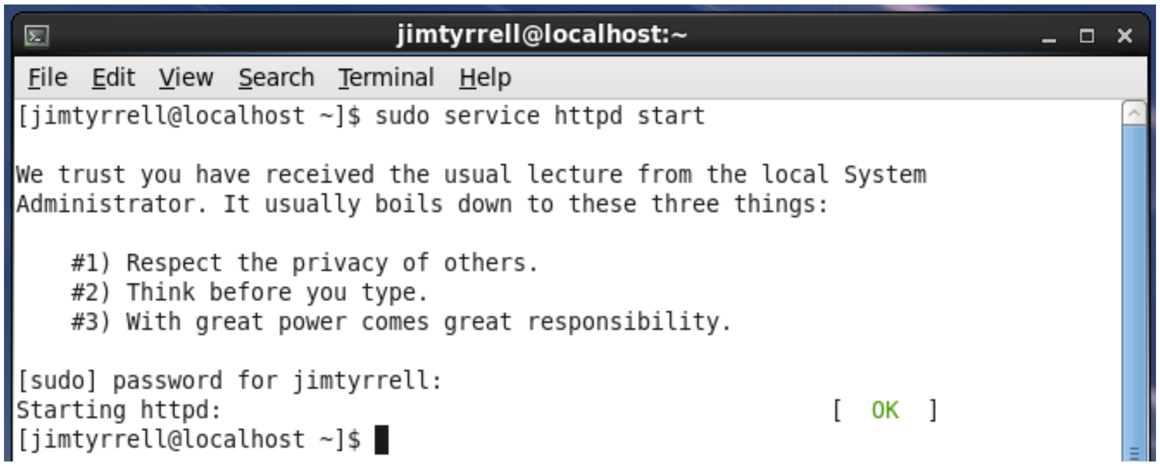


```
[jimtyrrell@localhost ~]$ httpd -v
Server version: Apache/2.2.15 (Unix)
Server built:   Aug 14 2010 08:53:20
```

If you open a web browser on an instance of Red Hat EnterpriSELinux 6 without doing anything, you will probably find that the httpd server has not been stared. To start it run:

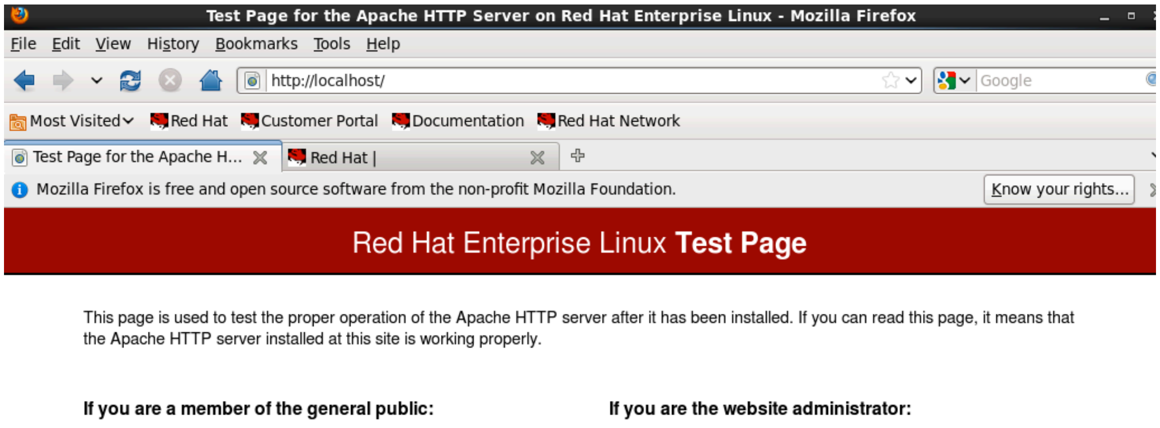
```
sudo service httpd start
```

As shown:



```
[jimtyrrell@localhost ~]$ sudo service httpd start

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for jimtyrrell:
Starting httpd:                                            [  OK  ]
[jimtyrrell@localhost ~]$ 
```

You should open a web browser and make sure that you can see the web server via http://localhost as shown:



Test Page for the Apache HTTP Server on Red Hat Enterprise Linux - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/
Google
Most Visited
Red Hat
Customer Portal
Documentation
Red Hat Network
Test Page for the Apache H...
Red Hat |
Know your rights...

Mozilla Firefox is free and open source software from the non-profit Mozilla Foundation.

Red Hat Enterprise Linux **Test Page**

This page is used to test the proper operation of the Apache HTTP server after it has been installed. If you can read this page, it means that the Apache HTTP server installed at this site is working properly.

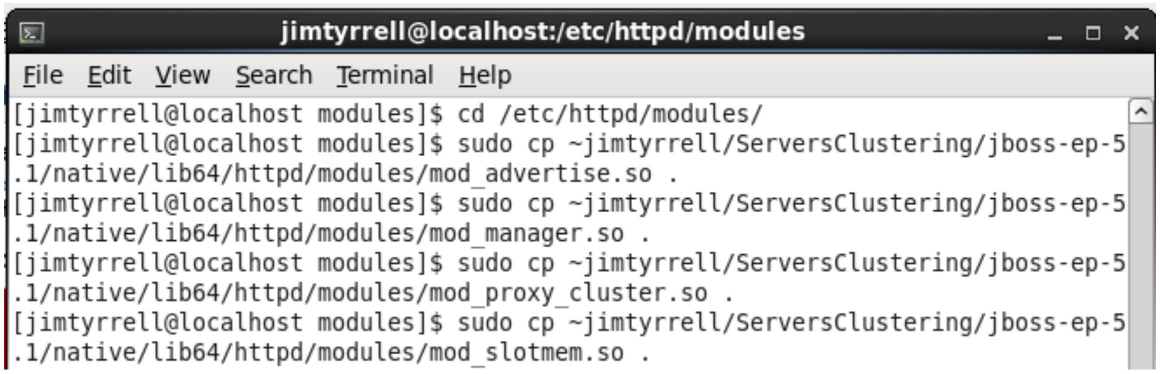
If you are a member of the general public:

If you are the website administrator:

Copy/Install Modules

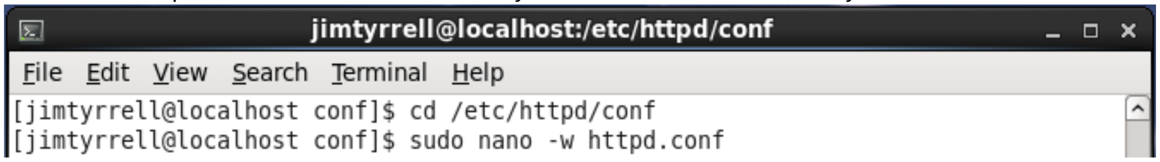
Now it is time to install mod_cluster so it can act as a proxy sending out requests to the installed/discovered JBoss Enterprise Application Platform Instances.
The first step is to copy the four native files that are needed into the apache modules folder.

```
cd /etc/httpd/modules/
sudo cp ~student/ServersClustering/jboss-ep-5.1/native/lib64/httpd/modules/
mod_advertise.so .
sudo cp ~student/ServersClustering/jboss-ep-5.1/native/lib64/httpd/modules/
mod_manager.so .
sudo cp ~student/ServersClustering/jboss-ep-5.1/native/lib64/httpd/modules/
mod_proxy_cluster.so .
sudo cp ~student/ServersClustering/jboss-ep-5.1/native/lib64/httpd/modules/
mod_slotmem.so .
```

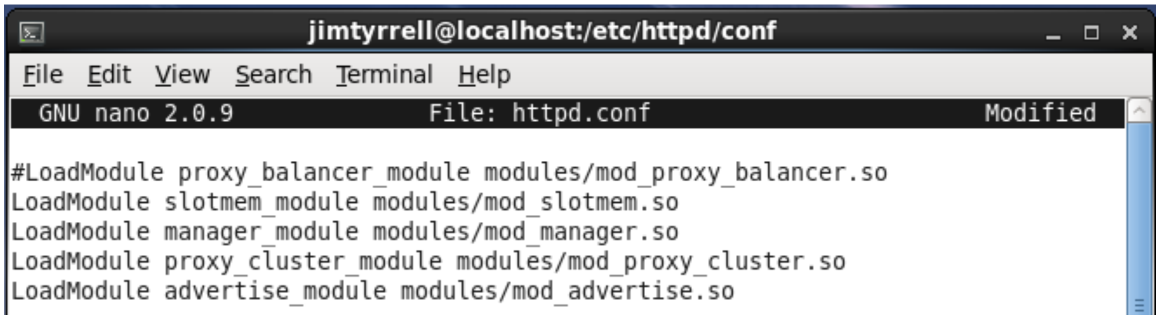

```
[jimtyrrell@localhost modules]$ cd /etc/httpd/modules/
[jimtyrrell@localhost modules]$ sudo cp ~jimtyrrell/ServersClustering/jboss-ep-5
.1/native/lib64/httpd/modules/mod_advertise.so .
[jimtyrrell@localhost modules]$ sudo cp ~jimtyrrell/ServersClustering/jboss-ep-5
.1/native/lib64/httpd/modules/mod_manager.so .
[jimtyrrell@localhost modules]$ sudo cp ~jimtyrrell/ServersClustering/jboss-ep-5
.1/native/lib64/httpd/modules/mod_proxy_cluster.so .
[jimtyrrell@localhost modules]$ sudo cp ~jimtyrrell/ServersClustering/jboss-ep-5
.1/native/lib64/httpd/modules/mod_slotmem.so .
```

httpd.conf Changes

Next edit the httpd.cnf file as shown, make sure you use sudo to edit the file as you can see below:

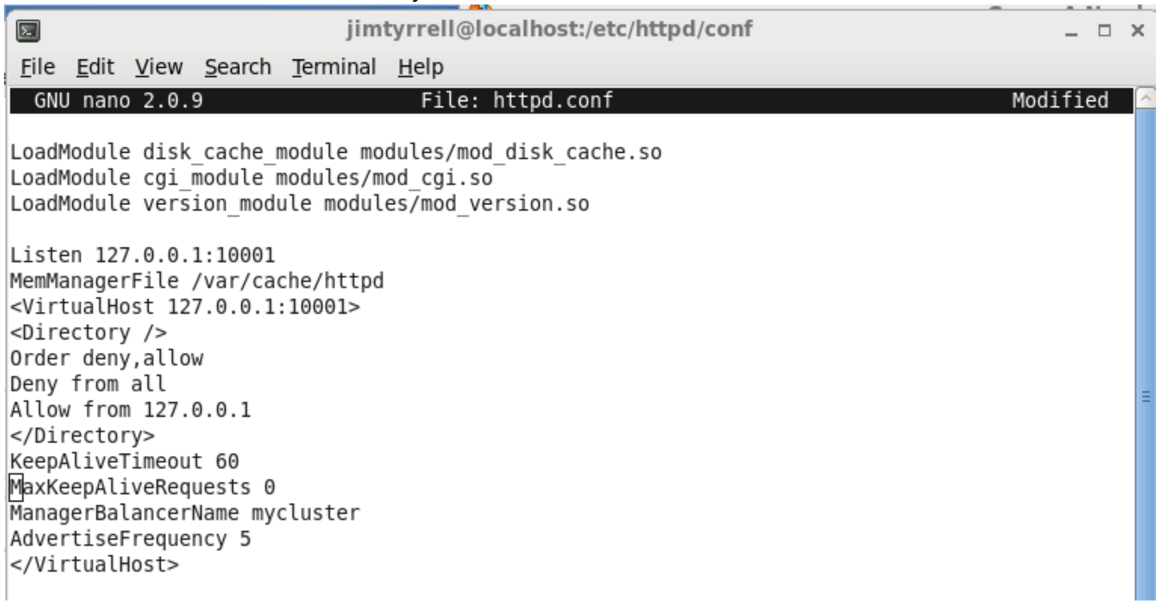

```
[jimtyrrell@localhost conf]$ cd /etc/httpd/conf
[jimtyrrell@localhost conf]$ sudo nano -w httpd.conf
```

```
Comment out an existing line, and add in four new lines:
#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so
```


```
  GNU nano 2.0.9              File: httpd.conf                    Modified

#LoadModule proxy_balancer_module modules/mod_proxy_balancer.so
LoadModule slotmem_module modules/mod_slotmem.so
LoadModule manager_module modules/mod_manager.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module modules/mod_advertise.so
```

```
In the same file add in the below section as shown:
Listen 127.0.0.1:10001
MemManagerFile /var/cache/httpd
<VirtualHost 127.0.0.1:10001>
   <Directory />
       Order deny,allow
       Deny from all
       Allow from 127.0.0.1
   </Directory>
   KeepAliveTimeout 60
   MaxKeepAliveRequests 0
   ManagerBalancerName mycluster
   AdvertiseFrequency 5
</VirtualHost>
<Location /mod_cluster-manager>
   SetHandler mod_cluster-manager
   Order deny,allow
   Deny from all
   Allow from 127.0.0.1
</Location>
```

At the end of the LoadModule Section you can add the above as shown:

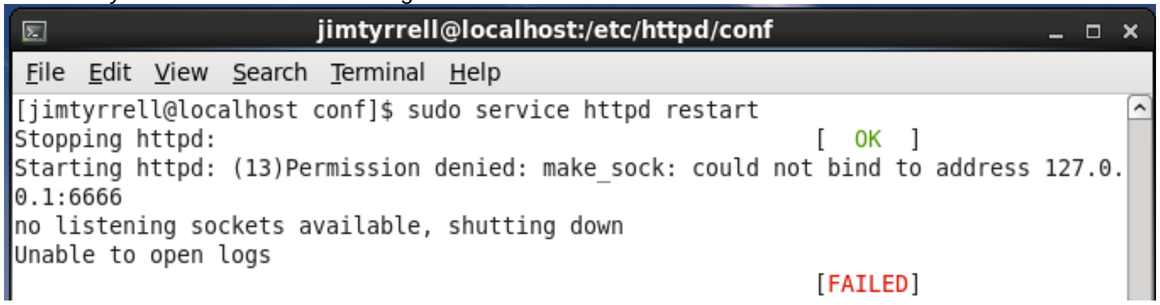

jimtyrrell@localhost:/etc/httpd/conf
File Edit View Search Terminal Help

```
  GNU nano 2.0.9                  File: httpd.conf                      Modified

LoadModule disk_cache_module modules/mod_disk_cache.so
LoadModule cgi_module modules/mod_cgi.so
LoadModule version_module modules/mod_version.so

Listen 127.0.0.1:10001
MemManagerFile /var/cache/httpd
<VirtualHost 127.0.0.1:10001>
<Directory />
Order deny,allow
Deny from all
Allow from 127.0.0.1
</Directory>
KeepAliveTimeout 60
MaxKeepAliveRequests 0
ManagerBalancerName mycluster
AdvertiseFrequency 5
</VirtualHost>
```

Start httpd

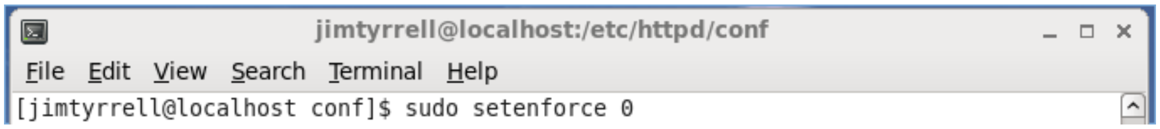
Now you should run:
sudo service httpd restart
You will note that you will get an error, as by default SELinux is enabled in Red Hat Enterprise Linux 6. For the purposes of this lab, we are going to disable this, by putting it into permissive mode. In SELinux enforcing mode, denied access is both blocked and logged. In permissive mode, denied access is not blocked but is logged. setenforce will only change the state until next reboot or until the mode is changed setenforce again, however, in a real production setting you would not want to do this. Appendix A will walk you through how to enable SELinux. There are many reasons for using SELinux that are far outside the scope of this Lab.

The error you will see looks something like this:


jimtyrrell@localhost:/etc/httpd/conf
File Edit View Search Terminal Help

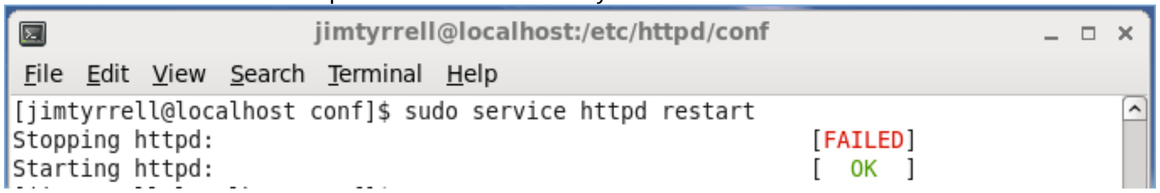
```
[jimtyrrell@localhost conf]$ sudo service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd: (13)Permission denied: make_sock: could not bind to address 127.0.
0.1:6666
no listening sockets available, shutting down
Unable to open logs
                                                           [FAILED]
```

In order to fix this turn off SELinux by running the command:
sudo setenforce 0
as shown:


jimtyrrell@localhost:/etc/httpd/conf
File Edit View Search Terminal Help

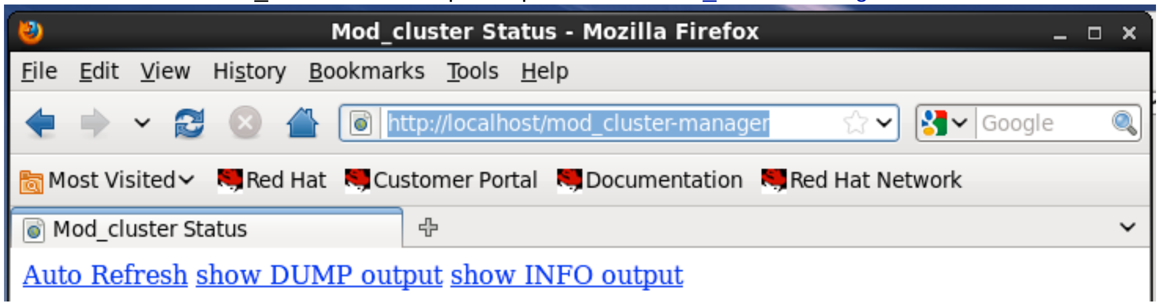
```
[jimtyrrell@localhost conf]$ sudo setenforce 0
```

Now rerun the sudo service httpd restart command and you should have success as shown:



```
[jimtyrrell@localhost conf]$ sudo service httpd restart
Stopping httpd:                                            [FAILED]
Starting httpd:                                            [  OK  ]
```

Now check out the mod_cluster console open http://localhost/mod_cluster-manager as shown:



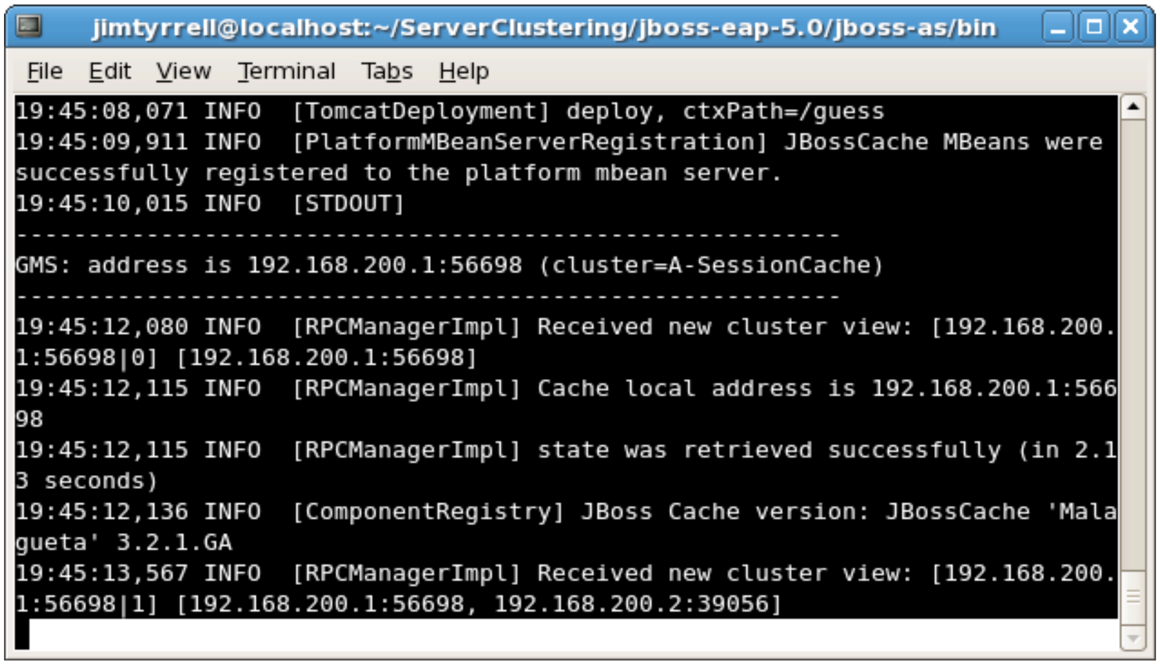
Auto Refresh show DUMP output show INFO output

4.5. Deploy an Application

Caution: For the ease of the lab we will be using the farm directory to replicate war files for us, however under no circumstances should you ever use that method in production.
~student/Desktop/Downloads/Clustering

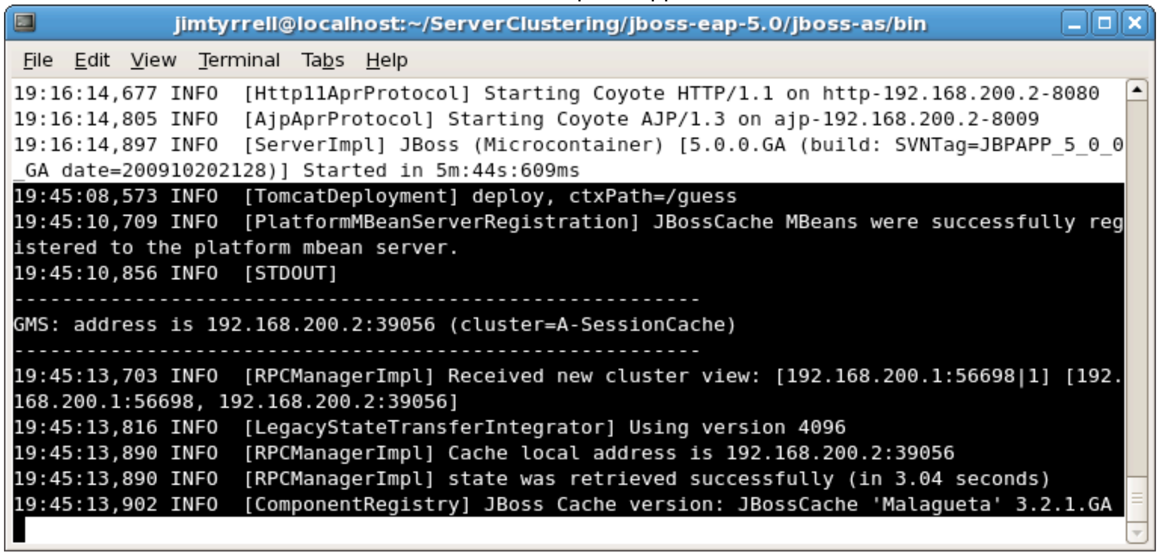
Deploy

In your ~student/Desktop/Downloads/Clustering/session-demo-wars folder are there war files guessv1.war, guessv2.war and guessv3.war. You just need to copy the guessv1.war file into the farm directory of node1.
cp -R guessv1.war ~student/ServersClustering/jboss-eap-5.1/jboss-as/server/node1/farm/guess.war
You can see that node1 deployed the application and also copied it via the farm service to node2 in the next screenshot:



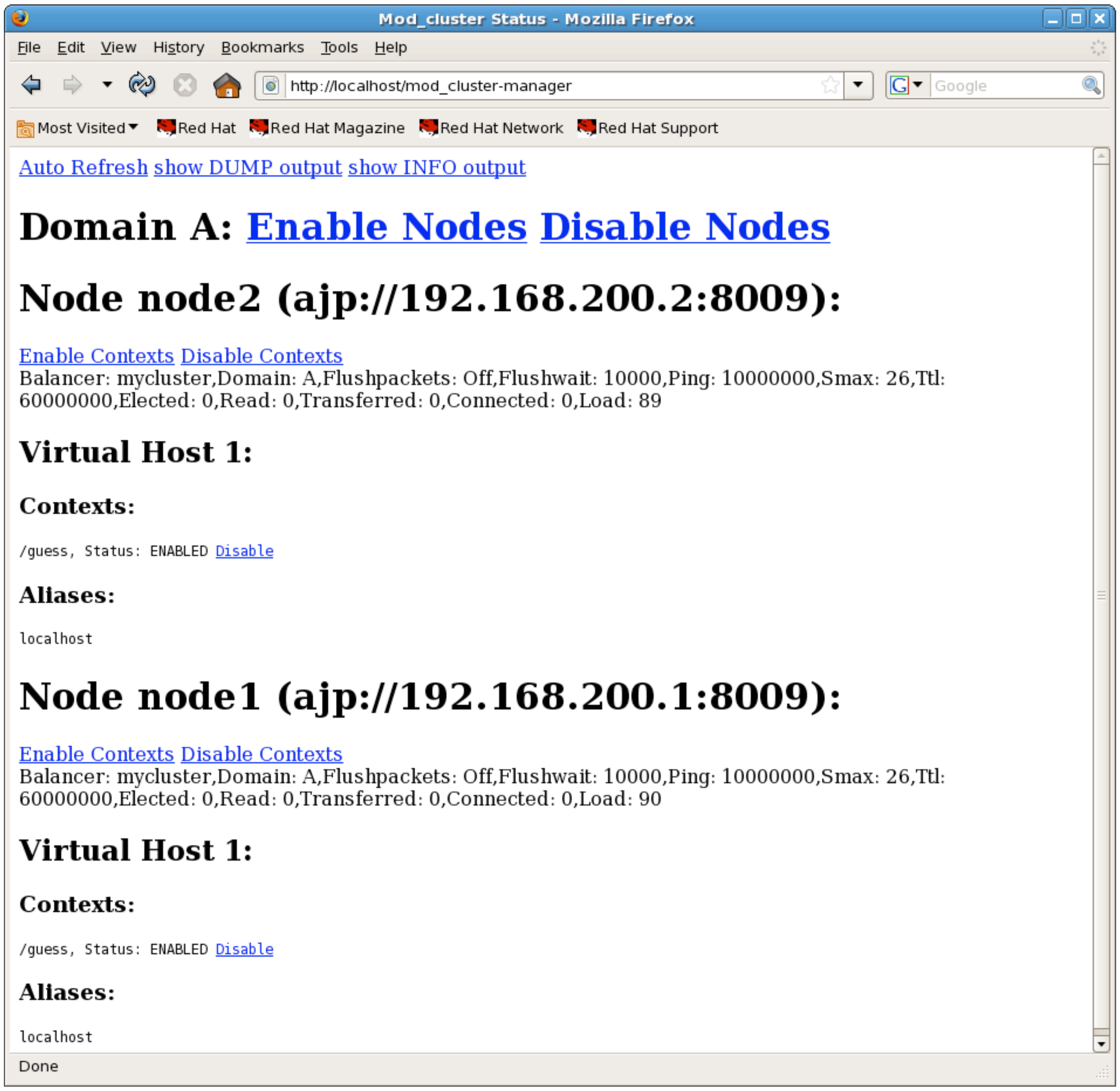
```
19:45:08,071 INFO  [TomcatDeployment] deploy, ctxPath=/guess
19:45:09,911 INFO  [PlatformMBeanServerRegistration] JBossCache MBeans were
successfully registered to the platform mbean server.
19:45:10,015 INFO  [STDOUT]
---------------------------------------------------------
GMS: address is 192.168.200.1:56698 (cluster=A-SessionCache)
---------------------------------------------------------
19:45:12,080 INFO  [RPCManagerImpl] Received new cluster view: [192.168.200.
1:56698|0] [192.168.200.1:56698]
19:45:12,115 INFO  [RPCManagerImpl] Cache local address is 192.168.200.1:566
98
19:45:12,115 INFO  [RPCManagerImpl] state was retrieved successfully (in 2.1
3 seconds)
19:45:12,136 INFO  [ComponentRegistry] JBoss Cache version: JBossCache 'Mala
gueta' 3.2.1.GA
19:45:13,567 INFO  [RPCManagerImpl] Received new cluster view: [192.168.200.
1:56698|1] [192.168.200.1:56698, 192.168.200.2:39056]
```

You can see that node2 received the file and started up the application:


jimtyrrell@localhost:~/ServerClustering/jboss-eap-5.0/jboss-as/bin
File Edit View Terminal Tabs Help

```
19:16:14,677 INFO  [Http11AprProtocol] Starting Coyote HTTP/1.1 on http-192.168.200.2-8080
19:16:14,805 INFO  [AjpAprProtocol] Starting Coyote AJP/1.3 on ajp-192.168.200.2-8009
19:16:14,897 INFO  [ServerImpl] JBoss (Microcontainer) [5.0.0.GA (build: SVNTag=JBPAPP_5_0_0
_GA date=200910202128)] Started in 5m:44s:609ms
19:45:08,573 INFO  [TomcatDeployment] deploy, ctxPath=/guess
19:45:10,709 INFO  [PlatformMBeanServerRegistration] JBossCache MBeans were successfully reg
istered to the platform mbean server.
19:45:10,856 INFO  [STDOUT]
---------------------------------------------------------
GMS: address is 192.168.200.2:39056 (cluster=A-SessionCache)
---------------------------------------------------------
19:45:13,703 INFO  [RPCManagerImpl] Received new cluster view: [192.168.200.1:56698|1] [192.
168.200.1:56698, 192.168.200.2:39056]
19:45:13,816 INFO  [LegacyStateTransferIntegrator] Using version 4096
19:45:13,890 INFO  [RPCManagerImpl] Cache local address is 192.168.200.2:39056
19:45:13,890 INFO  [RPCManagerImpl] state was retrieved successfully (in 3.04 seconds)
19:45:13,902 INFO  [ComponentRegistry] JBoss Cache version: JBossCache 'Malagueta' 3.2.1.GA
```

Now refresh your browser window, and you should see the two nodes each with a new /guess context as shown below:


Mod_cluster Status - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/mod_cluster-manager
Google
Most Visited
Red Hat
Red Hat Magazine
Red Hat Network
Red Hat Support
Done

Auto Refresh show DUMP output show INFO output

Domain A: Enable Nodes Disable Nodes

Node node2 (ajp://192.168.200.2:8009):

Enable Contexts Disable Contexts
Balancer: mycluster,Domain: A,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 26,Ttl: 60000000,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 89

Virtual Host 1:

Contexts:

```
/guess, Status: ENABLED Disable
```

Aliases:

```
localhost
```

Node node1 (ajp://192.168.200.1:8009):

Enable Contexts Disable Contexts
Balancer: mycluster,Domain: A,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 26,Ttl: 60000000,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 90

Virtual Host 1:

Contexts:

```
/guess, Status: ENABLED Disable
```

Aliases:

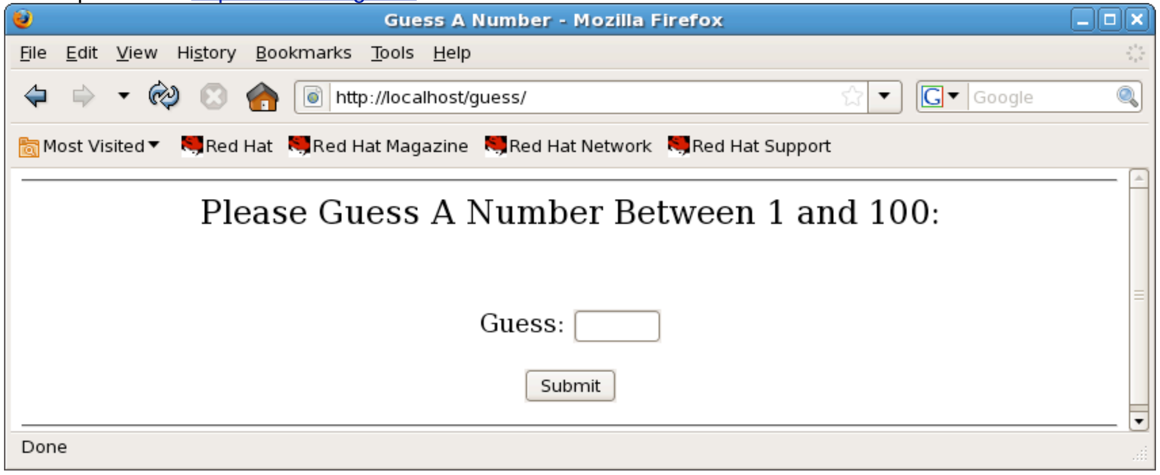
```
localhost
```

Congratulations you now have the applications deployed and you are now ready to see the load balancing in action. If your web browser does not look like this, please raise your hand.

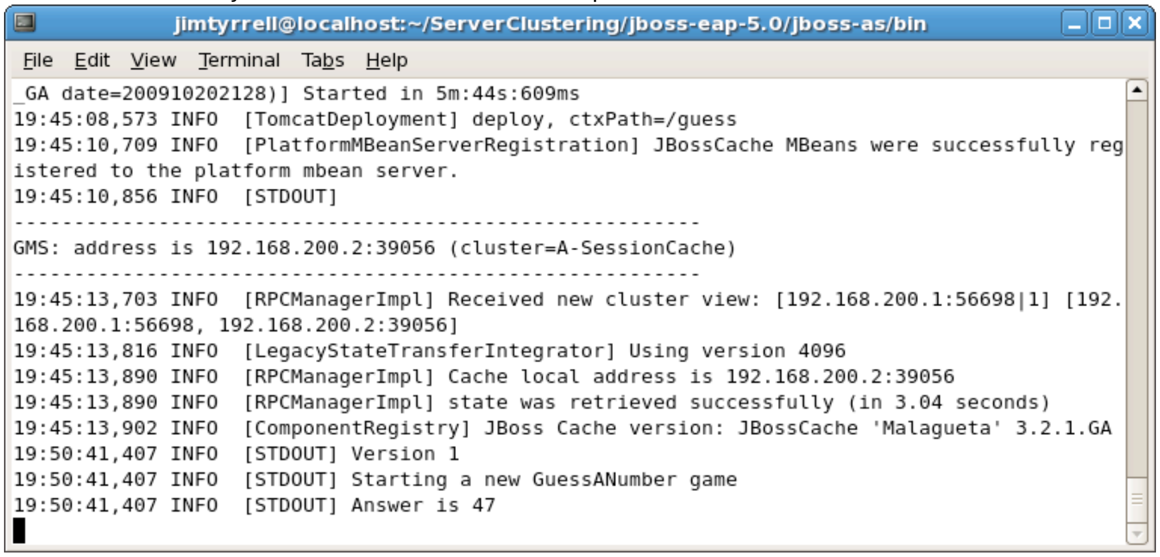
4.6. Bounce Servers and Explore What Happens

Test Load Balancing

Load up the URL http://localhost/guess as shown below:

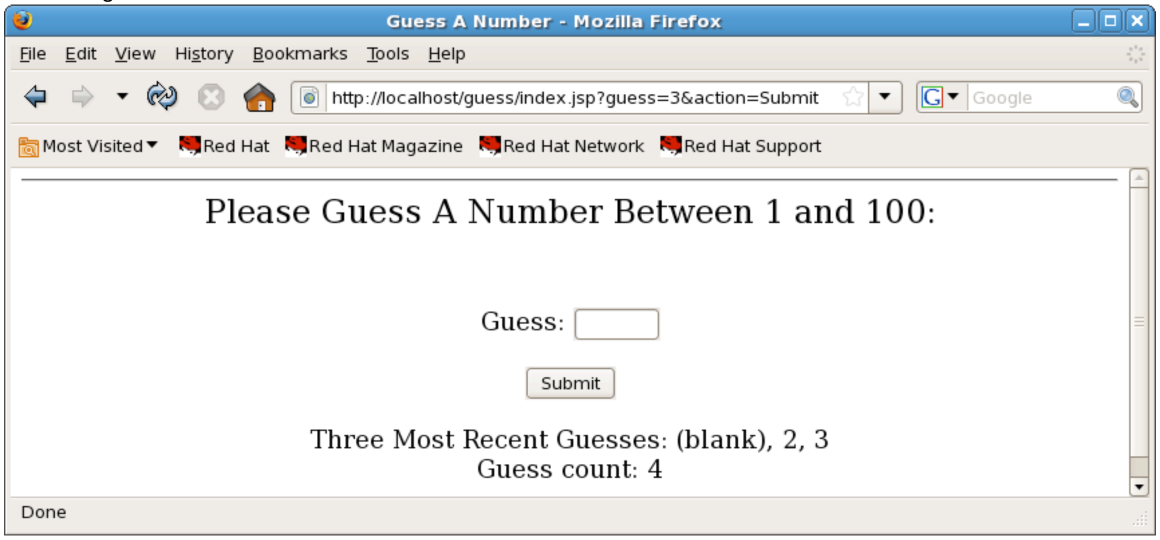

Guess A Number - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/guess/
Most Visited Red Hat Red Hat Magazine Red Hat Network Red Hat Support
Please Guess A Number Between 1 and 100:
Guess:
Submit
Done

Also note that one of your node terminals has some output that looks like this:


jimtyrrell@localhost:~/ServerClustering/jboss-eap-5.0/jboss-as/bin
File Edit View Terminal Tabs Help

```
_GA date=200910202128)] Started in 5m:44s:609ms
19:45:08,573 INFO  [TomcatDeployment] deploy, ctxPath=/guess
19:45:10,709 INFO  [PlatformMBeanServerRegistration] JBossCache MBeans were successfully reg
istered to the platform mbean server.
19:45:10,856 INFO  [STDOUT]
---------------------------------------------------------
GMS: address is 192.168.200.2:39056 (cluster=A-SessionCache)
---------------------------------------------------------
19:45:13,703 INFO  [RPCManagerImpl] Received new cluster view: [192.168.200.1:56698|1] [192.
168.200.1:56698, 192.168.200.2:39056]
19:45:13,816 INFO  [LegacyStateTransferIntegrator] Using version 4096
19:45:13,890 INFO  [RPCManagerImpl] Cache local address is 192.168.200.2:39056
19:45:13,890 INFO  [RPCManagerImpl] state was retrieved successfully (in 3.04 seconds)
19:45:13,902 INFO  [ComponentRegistry] JBoss Cache version: JBossCache 'Malagueta' 3.2.1.GA
19:50:41,407 INFO  [STDOUT] Version 1
19:50:41,407 INFO  [STDOUT] Starting a new GuessANumber game
19:50:41,407 INFO  [STDOUT] Answer is 47
```

This is the starting of a new session that is unique to that server, however under the covers this stats has been replicated. Make at least three guesses in the web browser, note the answer is above, not the most secure application is it :). This is also the node that will serve the duration of a users requests until it is shutdown. This is done via sticky sessions. When your are all done your browser window will look something like this:

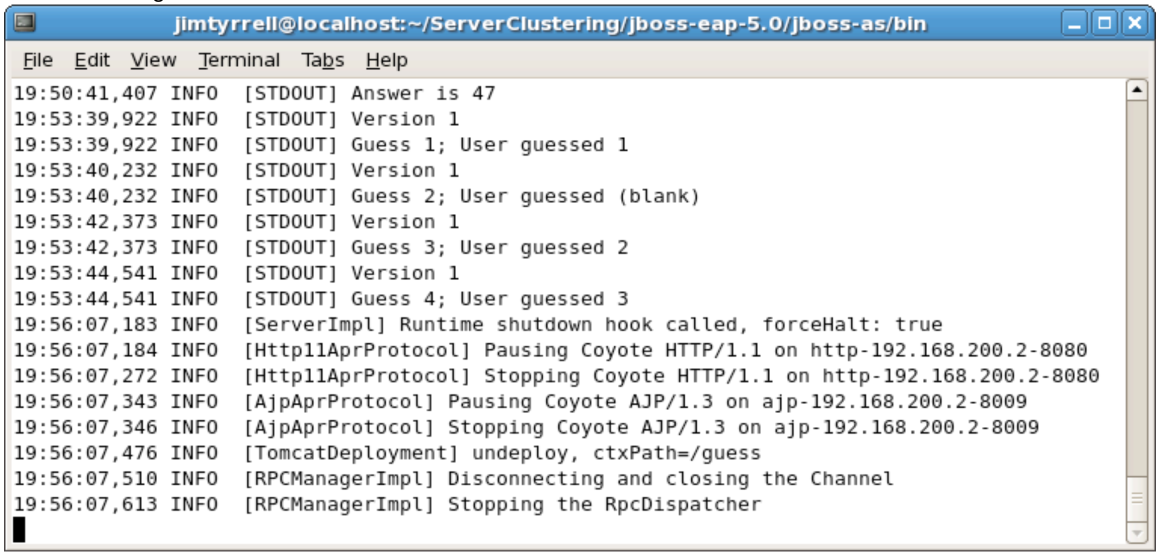

Guess A Number - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/guess/index.jsp?guess=3&action=Submit
Google
Most Visited
Red Hat
Red Hat Magazine
Red Hat Network
Red Hat Support
Submit
Done

Please Guess A Number Between 1 and 100:

Guess:

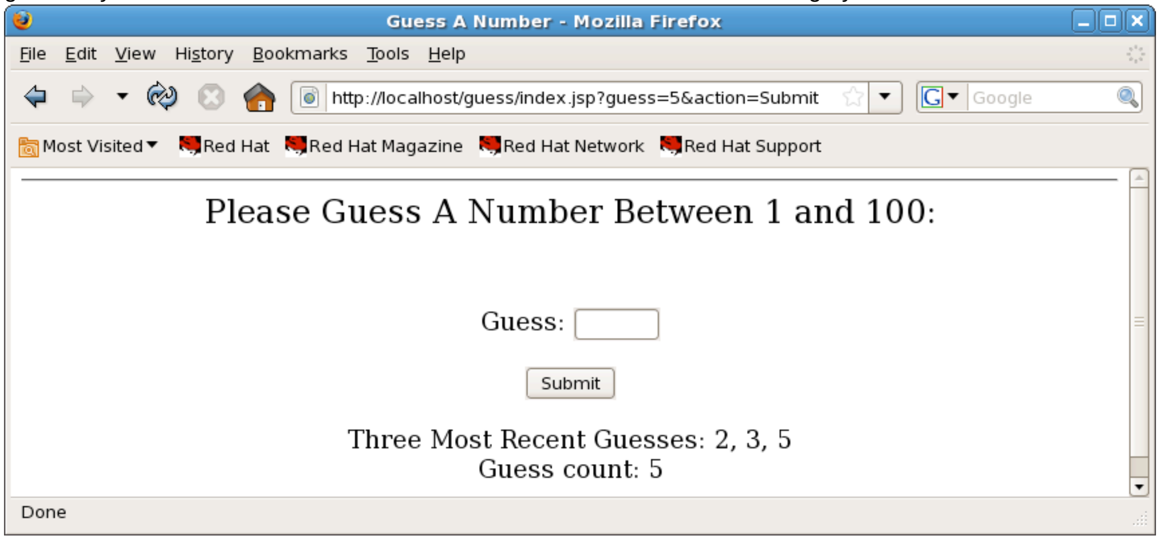
Three Most Recent Guesses: (blank), 2, 3
Guess count: 4

Now kill the node that was echoing the output shown above by press cntrl-c in the terminal window, it should look something like this:


jimtyrrell@localhost:~/ServerClustering/jboss-eap-5.0/jboss-as/bin
File Edit View Terminal Tabs Help

```
19:50:41,407 INFO  [STDOUT] Answer is 47
19:53:39,922 INFO  [STDOUT] Version 1
19:53:39,922 INFO  [STDOUT] Guess 1; User guessed 1
19:53:40,232 INFO  [STDOUT] Version 1
19:53:40,232 INFO  [STDOUT] Guess 2; User guessed (blank)
19:53:42,373 INFO  [STDOUT] Version 1
19:53:42,373 INFO  [STDOUT] Guess 3; User guessed 2
19:53:44,541 INFO  [STDOUT] Version 1
19:53:44,541 INFO  [STDOUT] Guess 4; User guessed 3
19:56:07,183 INFO  [ServerImpl] Runtime shutdown hook called, forceHalt: true
19:56:07,184 INFO  [Http11AprProtocol] Pausing Coyote HTTP/1.1 on http-192.168.200.2-8080
19:56:07,272 INFO  [Http11AprProtocol] Stopping Coyote HTTP/1.1 on http-192.168.200.2-8080
19:56:07,343 INFO  [AjpAprProtocol] Pausing Coyote AJP/1.3 on ajp-192.168.200.2-8009
19:56:07,346 INFO  [AjpAprProtocol] Stopping Coyote AJP/1.3 on ajp-192.168.200.2-8009
19:56:07,476 INFO  [TomcatDeployment] undeploy, ctxPath=/guess
19:56:07,510 INFO  [RPCManagerImpl] Disconnecting and closing the Channel
19:56:07,613 INFO  [RPCManagerImpl] Stopping the RpcDispatcher
█
```

You will now see some updated messages in the other node that was not hosting the work, submit a new guess in your browser window and see that the state was not lost even though you shutdown the server:

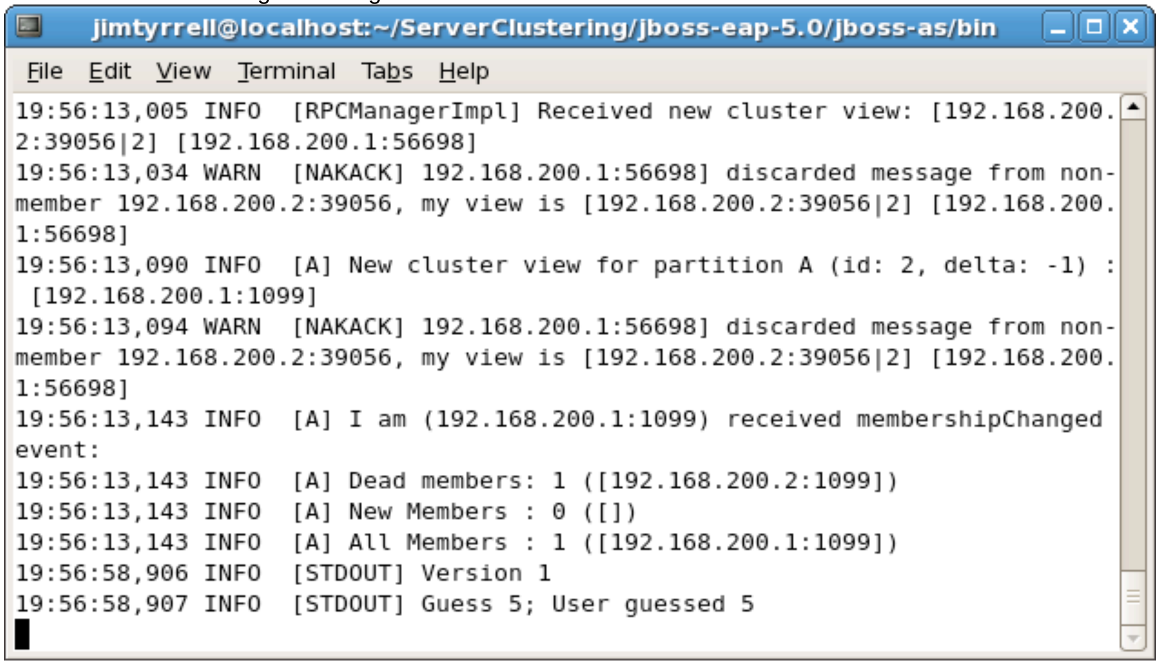

Guess A Number - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/guess/index.jsp?guess=5&action=Submit
Most Visited Red Hat Red Hat Magazine Red Hat Network Red Hat Support
Submit
Done

Please Guess A Number Between 1 and 100:

Guess:

Three Most Recent Guesses: 2, 3, 5
Guess count: 5

and the terminal showing the new guess:


jimtyrrell@localhost:~/ServerClustering/jboss-eap-5.0/jboss-as/bin
File Edit View Terminal Tabs Help

```
19:56:13,005 INFO  [RPCManagerImpl] Received new cluster view: [192.168.200.
2:39056|2] [192.168.200.1:56698]
19:56:13,034 WARN  [NAKACK] 192.168.200.1:56698] discarded message from non-
member 192.168.200.2:39056, my view is [192.168.200.2:39056|2] [192.168.200.
1:56698]
19:56:13,090 INFO  [A] New cluster view for partition A (id: 2, delta: -1) :
 [192.168.200.1:1099]
19:56:13,094 WARN  [NAKACK] 192.168.200.1:56698] discarded message from non-
member 192.168.200.2:39056, my view is [192.168.200.2:39056|2] [192.168.200.
1:56698]
19:56:13,143 INFO  [A] I am (192.168.200.1:1099) received membershipChanged
event:
19:56:13,143 INFO  [A] Dead members: 1 ([192.168.200.2:1099])
19:56:13,143 INFO  [A] New Members : 0 ([])
19:56:13,143 INFO  [A] All Members : 1 ([192.168.200.1:1099])
19:56:58,906 INFO  [STDOUT] Version 1
19:56:58,907 INFO  [STDOUT] Guess 5; User guessed 5
```

Restart by up arrowing to the server start command you just stopped with cntrl-c, wait until you see it has joined the cluster and completely started, then kill the other server.

Now put in another guess and see how the load was automatically sent to the other server.

Extra Credit

A more advanced version of this lab, would be to start up node3 with the third ./run.sh file in the readme.txt and see how it dynamically joins the cluster. When it is fully started feel free to kill the other two nodes.

The steps are outlined below:

1. Start up the third server, notice how the guess.war file was copied for you.
2. Refresh the http://localhost/mod_cluster_manager to see how the third server has joined the cluster.

3. Kill the other one or two nodes once the third server has started (one or two depending on how you have been playing around with this)
4. Submit another guess and see how it has been load balanced
5. Congratulations you have now clustered and load balanced your war file across at least 2 servers, and possibly three if you did the more advanced part of the lab.

Feel free to play around with this some more.

When you are done make sure all three instances are closed down.

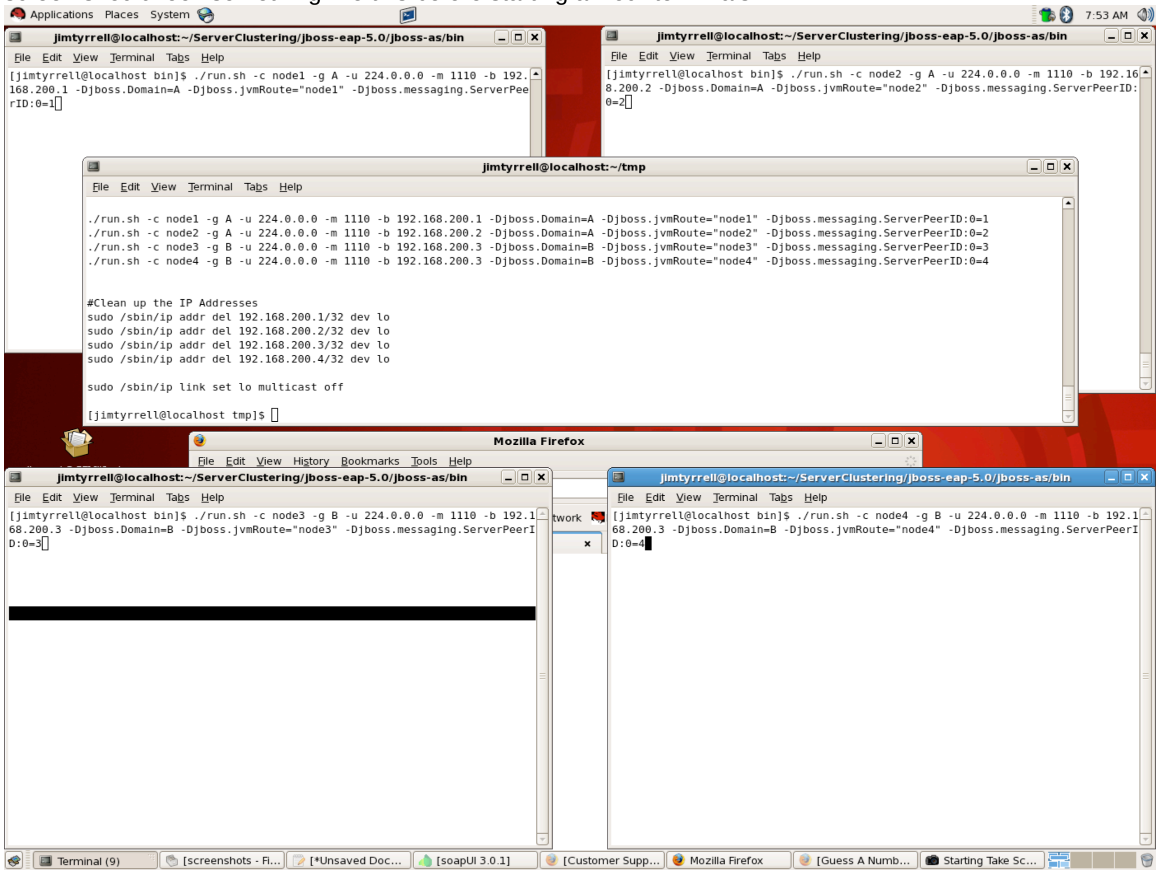
4.7. Rolling Server Restarts

Start Up the Servers

Open up the run.conf file in the jboss-as/bin directory and change the line JAVA_OPTS to have -Xms512m -Xmx512m instead of the defaults. In order to run this white paper/lab on a single computer their is no need to run with the default larger JVM, in order to squeeze several JVMs on one computer, it is easy and recommended to shrink the size of the required space for the JVM/JBoss EAP. If you are running on a machine with 8 GB or more, you do not need to shrink the default memory settings.

Open the readme.txt in the ~student/Desktop/Downloads/Clustering folder

Use the second set of server start commands to start four instances of JBoss, notice there is domain A and B in the startup commands. Also notice the different unique multicast addresses that are in use. Your screen should look something like this before starting all four terminals:



```
[jimtyrrell@localhost bin]$ ./run.sh -c node1 -g A -u 224.0.0.0 -m 1110 -b 192.
168.200.1 -Djboss.Domain=A -Djboss.jvmRoute="node1" -Djboss.messaging.ServerPee
rID:0=1
```

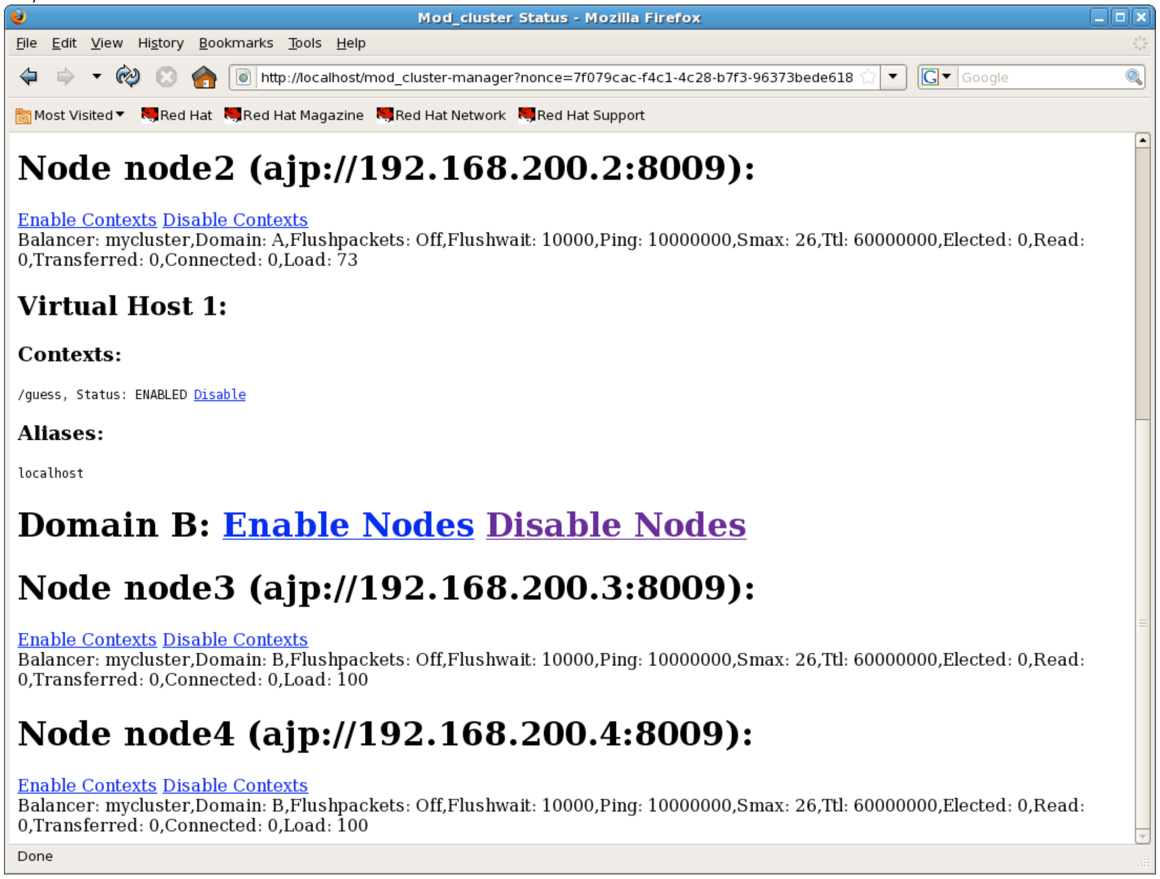
```
[jimtyrrell@localhost bin]$ ./run.sh -c node2 -g A -u 224.0.0.0 -m 1110 -b 192.16
8.200.2 -Djboss.Domain=A -Djboss.jvmRoute="node2" -Djboss.messaging.ServerPeerID:
0=2
```

```
./run.sh -c node1 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.1 -Djboss.Domain=A -Djboss.jvmRoute="node1" -Djboss.messaging.ServerPeerID:0=1
./run.sh -c node2 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.2 -Djboss.Domain=A -Djboss.jvmRoute="node2" -Djboss.messaging.ServerPeerID:0=2
./run.sh -c node3 -g B -u 224.0.0.0 -m 1110 -b 192.168.200.3 -Djboss.Domain=B -Djboss.jvmRoute="node3" -Djboss.messaging.ServerPeerID:0=3
./run.sh -c node4 -g B -u 224.0.0.0 -m 1110 -b 192.168.200.3 -Djboss.Domain=B -Djboss.jvmRoute="node4" -Djboss.messaging.ServerPeerID:0=4


#Clean up the IP Addresses
sudo /sbin/ip addr del 192.168.200.1/32 dev lo
sudo /sbin/ip addr del 192.168.200.2/32 dev lo
sudo /sbin/ip addr del 192.168.200.3/32 dev lo
sudo /sbin/ip addr del 192.168.200.4/32 dev lo

sudo /sbin/ip link set lo multicast off

[jimtyrrell@localhost tmp]$
```

```
[jimtyrrell@localhost bin]$ ./run.sh -c node3 -g B -u 224.0.0.0 -m 1110 -b 192.1
68.200.3 -Djboss.Domain=B -Djboss.jvmRoute="node3" -Djboss.messaging.ServerPeerI
D:0=3
```

```
[jimtyrrell@localhost bin]$ ./run.sh -c node4 -g B -u 224.0.0.0 -m 1110 -b 192.1
68.200.3 -Djboss.Domain=B -Djboss.jvmRoute="node4" -Djboss.messaging.ServerPeerI
D:0=4
```

When you are done your http://localhost/mod_cluster-manger will have Domain A with the /guest application an, it will look about like this:


Mod_cluster Status - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/mod_cluster-manager?nonce=7f079cac-f4c1-4c28-b7f3-96373bede618
Google
Most Visited
Red Hat
Red Hat Magazine
Red Hat Network
Red Hat Support
Done

Node node2 (ajp://192.168.200.2:8009):

Enable Contexts Disable Contexts
Balancer: mycluster,Domain: A,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 26,Ttl: 60000000,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 73

Virtual Host 1:

Contexts:

```
/guess, Status: ENABLED Disable
```

Aliases:

```
localhost
```

Domain B: Enable Nodes Disable Nodes

Node node3 (ajp://192.168.200.3:8009):

Enable Contexts Disable Contexts
Balancer: mycluster,Domain: B,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 26,Ttl: 60000000,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 100

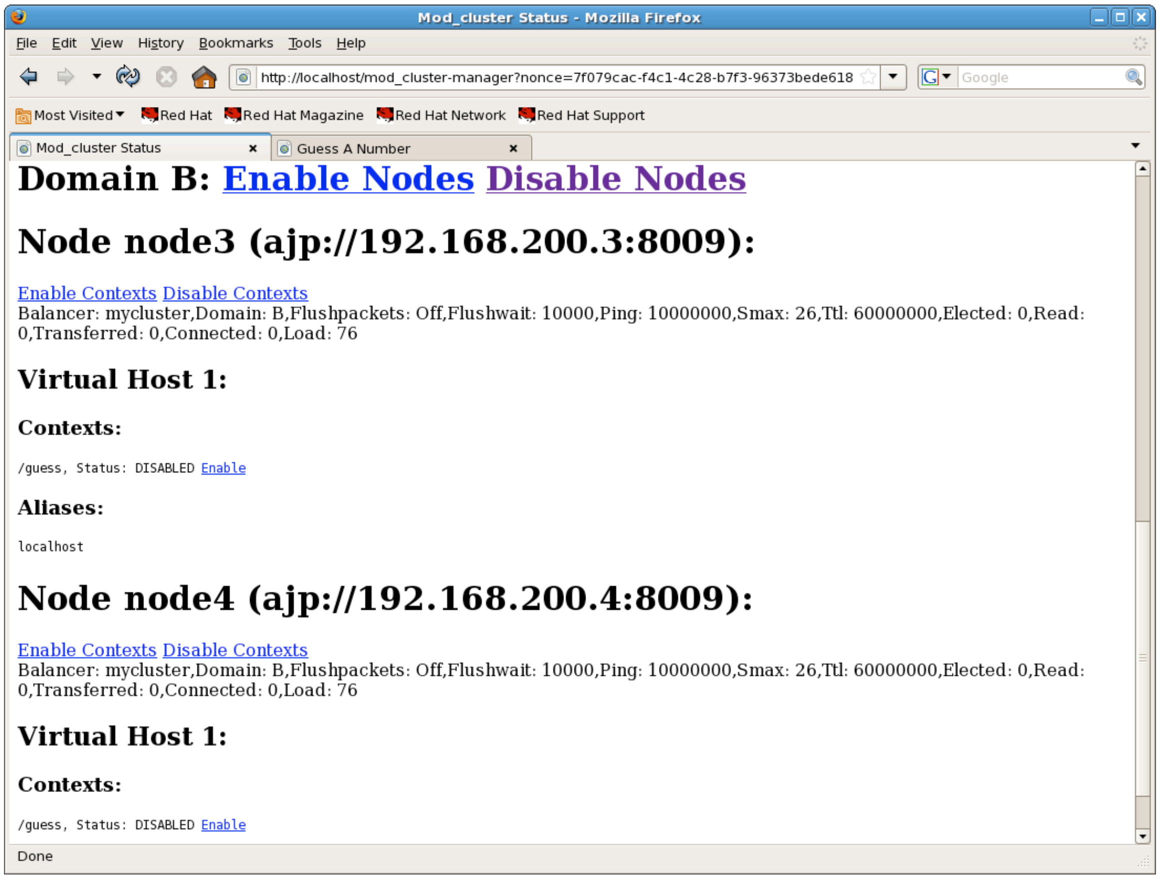
Node node4 (ajp://192.168.200.4:8009):

Enable Contexts Disable Contexts
Balancer: mycluster,Domain: B,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 26,Ttl: 60000000,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 100

Now make some guesses as you did before, shut down the node that is hosting your application as before. Notice that it rolls over to the other node.
Now copy your guessv2.war file from the ${User_Home}/Downloads/Clustering/session-demo-wars into the node3/farm directory.

```
cp -R ~student/Downloads/Clustering/session-demo-wars/guessv2.war ~student/
ServerClustering/jboss-eap-5.0/jboss-as/server/node3/farm
```

Note that it was copied over to node 4 and refresh your mod_cluster-manager page, and click Disable nodes on Domain B. This will stop traffic from going to Domain B. It should look like this, check out the Disabled Contexts:


Mod_cluster Status - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost/mod_cluster-manager?nonce=7f079cac-f4c1-4c28-b7f3-96373bede618
Google
Most Visited
Red Hat
Red Hat Magazine
Red Hat Network
Red Hat Support
Mod_cluster Status
Guess A Number

Domain B: Enable Nodes Disable Nodes

Node node3 (ajp://192.168.200.3:8009):

Enable Contexts Disable Contexts
Balancer: mycluster,Domain: B,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 26,Ttl: 60000000,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 76

Virtual Host 1:

Contexts:

```
/guess, Status: DISABLED Enable
```

Aliases:

```
localhost
```

Node node4 (ajp://192.168.200.4:8009):

Enable Contexts Disable Contexts
Balancer: mycluster,Domain: B,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 26,Ttl: 60000000,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 76

Virtual Host 1:

Contexts:

```
/guess, Status: DISABLED Enable
```

Done

Now enable Domain B, by clicking Enable Nodes. Disable Domain A, by clicking Disable Nodes. Now go to your browser window and submit a new guess. Note the request was sent one of the Domain A Nodes as it was disabled and not stopped. Disabling the A node still allows it to service requests.

Now kill the remaining server in Domain A. The session will failover to Domain B, but you will probably see an exception. Did you? This is because our application is not exactly coded to deal with this scenario. Doing live session migration is something that is generally pretty hard, and your applications need to be written to deal with this case.

Now continue on making guesses in Domain B, kill the server that is hosting the request, notice is fails over to the other node.

Now reverse the process, start up the two nodes in Domain A, leave the shutdown node in Domain B off.

Copy over version 3 of the guess.war file into the farm directory for node1 or 2 in Domain A.

Enable the Domain A in the mod_cluster-Manager, if it still shows disabled. Disable Domain B, and notice the failover and the fail.

You have now shown/demonstrated nearly 100% uptime with JBoss and rolling server migrations. You have now completed the clustering lab

4.8. Conclusion

What you learned

- You installed JBoss for mod_cluster
- You started Apache httpd
- You installed mod_cluster
- You explored the mod_cluster-manager
- You did live application migration and updating

4.9. Appendix: SELinux

SE Linux Considerations

As long as you are using port 10001 and the directive MemManagerFile as documented above, mod_cluster, The Apache Http Server, and JBoss EAP can run on a RHEL 6 system using the standard SELinux policy in enforcing mode without modification.

5. Resources

Various resources are listed below:

A Configuration Helper Utility:
http://clusterconfig.appspot.com/

Knowledge Base Articles for more Information:
https://access.redhat.com/kb/docs/DOC-34508
https://access.redhat.com/kb/docs/DOC-46133

All articles referencing mod_cluster:
https://access.redhat.com/knowledge/searchResults?start=1&col=redhat_kbase&quickSearch=mod_cluster&language=en

http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/5/html/Administration_And_Configuration_Guide/clustering-concepts-arch-balancer.html

http://docs.redhat.com/docs/en-US/JBoss_Enterprise_Application_Platform/5/html/Administration_And_Configuration_Guide/clustering-http.html

5.1.1 Documentation needs to go here, when it is available.

6. Appendix A - Readme.tx

```
#commands to turn multicast and get a few more IP Addresses going

sudo /sbin/ip link set lo multicast on

sudo /sbin/ip addr add 192.168.200.1 dev lo
sudo /sbin/ip addr add 192.168.200.2 dev lo
sudo /sbin/ip addr add 192.168.200.3 dev lo
sudo /sbin/ip addr add 192.168.200.4 dev lo

sudo /sbin/route add -net 224.1.1.1 netmask 240.1.1.1 lo
sudo /sbin/route add -net 224.2.2.2 netmask 240.2.2.2 lo

./run.sh -c node1 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.1 -Djboss.Domain=A -
Djboss.jvmRoute="node1" -Djboss.messaging.ServerPeerID:0=1
```

```
./run.sh -c node2 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.2 -Djboss.Domain=A -
Djboss.jvmRoute="node2" -Djboss.messaging.ServerPeerID:0=2
./run.sh -c node3 -g A -u 224.0.0.0 -m 1110 -b 192.168.200.3 -Djboss.Domain=A -
Djboss.jvmRoute="node3" -Djboss.messaging.ServerPeerID:0=3

./run.sh -c node1 -g A -u 232.1.1.1 -m 1110 -b 192.168.200.1 -Djboss.Domain=A -
Djboss.jvmRoute="node1" -Djboss.messaging.ServerPeerID:0=1
./run.sh -c node2 -g A -u 232.1.1.1 -m 1110 -b 192.168.200.2 -Djboss.Domain=A -
Djboss.jvmRoute="node2" -Djboss.messaging.ServerPeerID:0=2
./run.sh -c node3 -g B -u 232.2.2.2 -m 1110 -b 192.168.200.3 -Djboss.Domain=B -
Djboss.jvmRoute="node3" -Djboss.messaging.ServerPeerID:0=3
./run.sh -c node4 -g B -u 232.2.2.2 -m 1110 -b 192.168.200.4 -Djboss.Domain=B -
Djboss.jvmRoute="node4" -Djboss.messaging.ServerPeerID:0=4


#Clean up the IP Addresses
sudo /sbin/ip addr del 192.168.200.1/32 dev lo
sudo /sbin/ip addr del 192.168.200.2/32 dev lo
sudo /sbin/ip addr del 192.168.200.3/32 dev lo
sudo /sbin/ip addr del 192.168.200.4/32 dev lo
sudo /sbin/ip link set lo multicast off
```