

Introducción

- Un cluster de servidores de aplicaciones es un conjunto de dos o más servidores de aplicaciones configurados de forma que proveen:
 - Alta disponibilidad (HA High Availability).
 - Tolerancia a fallos (FailOver).
 - Mejoras de rendimiento (Performance).
 - Mejoras de escalabilidad (Simplifica el crecimiento).
- La mayoría de los servidores de aplicaciones permiten configurar clusters.

Cluster en WildFly (I)

- El servidor de aplicaciones WildFly posee dos modos de cluster:
 - Modo domain.
 - Modo standalone.
- En cualquier de los dos modos todas las aplicaciones que se despliegan en un cluster de Jboss deben tener la etiqueta `<distributable/>` en el fichero `web.xml`.

Cluster en WildFly (II)

- En el modo domain el cluster se configura como un nodo maestro y uno o más nodos esclavos.
- El nodo maestro:
 - Realiza toda la configuración de recursos (datasources).
 - Despliega las aplicaciones
- Permite, una vez creado el maestro, añadir de forma sencilla nuevos nodos al cluster.
- Su configuración inicial es complicada.

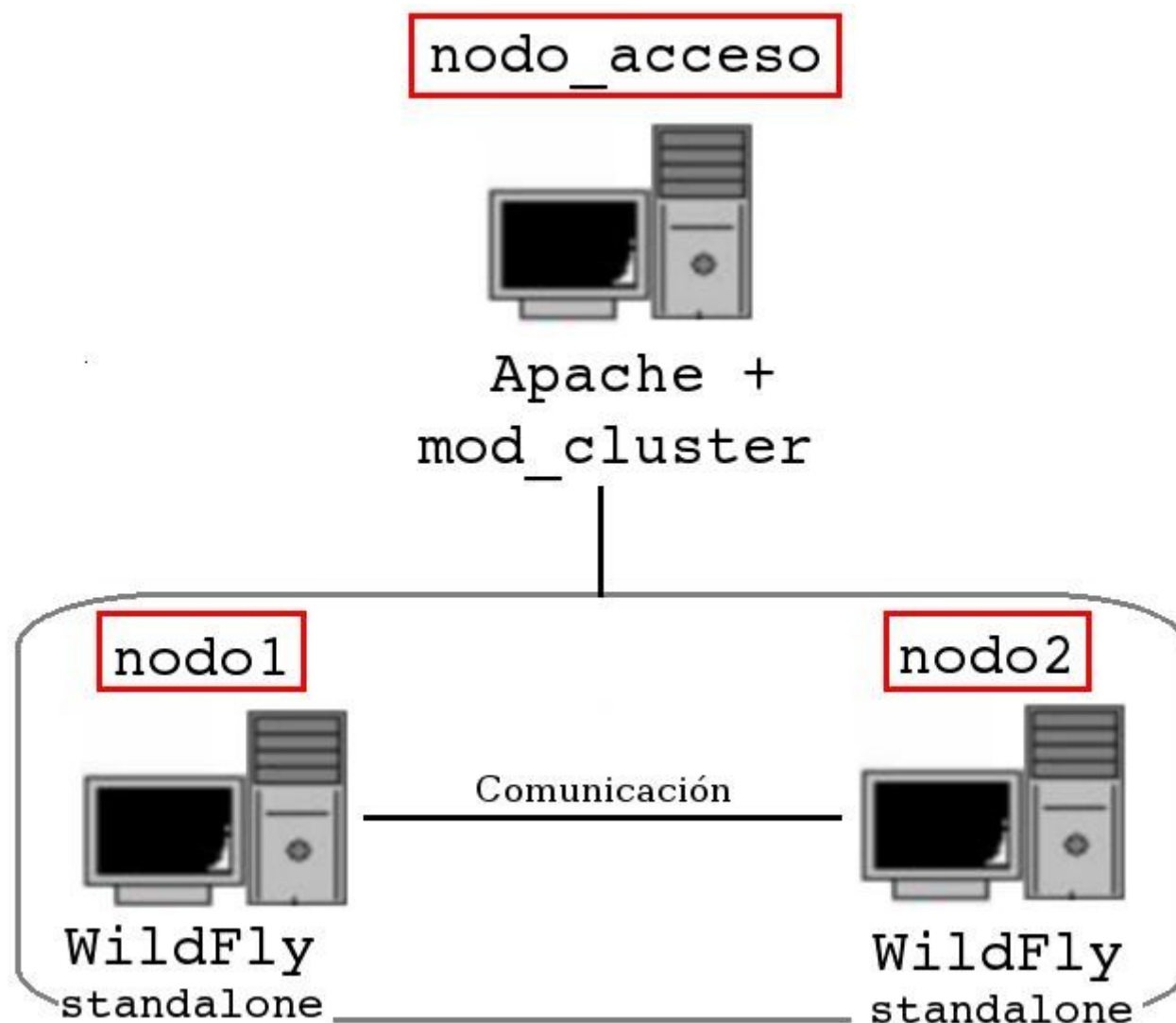
Cluster en WildFly (III)

- En el modo standalone el cluster se configura como de nodos iguales.
- Es necesario realizar en todos los nodos:
 - La configuración de recursos (datasources).
 - El despliegue de aplicaciones.
- Su configuración inicial es sencilla.
- Su gestión posterior es más complicada.

Estructura de un cluster (I)

- De forma general, la estructura de un cluster de servidores de aplicaciones esta formada por:
 - Dos o más nodos que ejecutan el servidor de aplicaciones y que forman un cluster de servidores de aplicaciones.
- Un servidor web Apache a través del cual se accede a los recursos del servidor de aplicaciones.
 - El servidor web puede ser también un cluster de servidores web si se necesita alta disponibilidad.
 - Envía las solicitudes a uno u otro servidor de aplicaciones.

Estructura de un cluster (II)



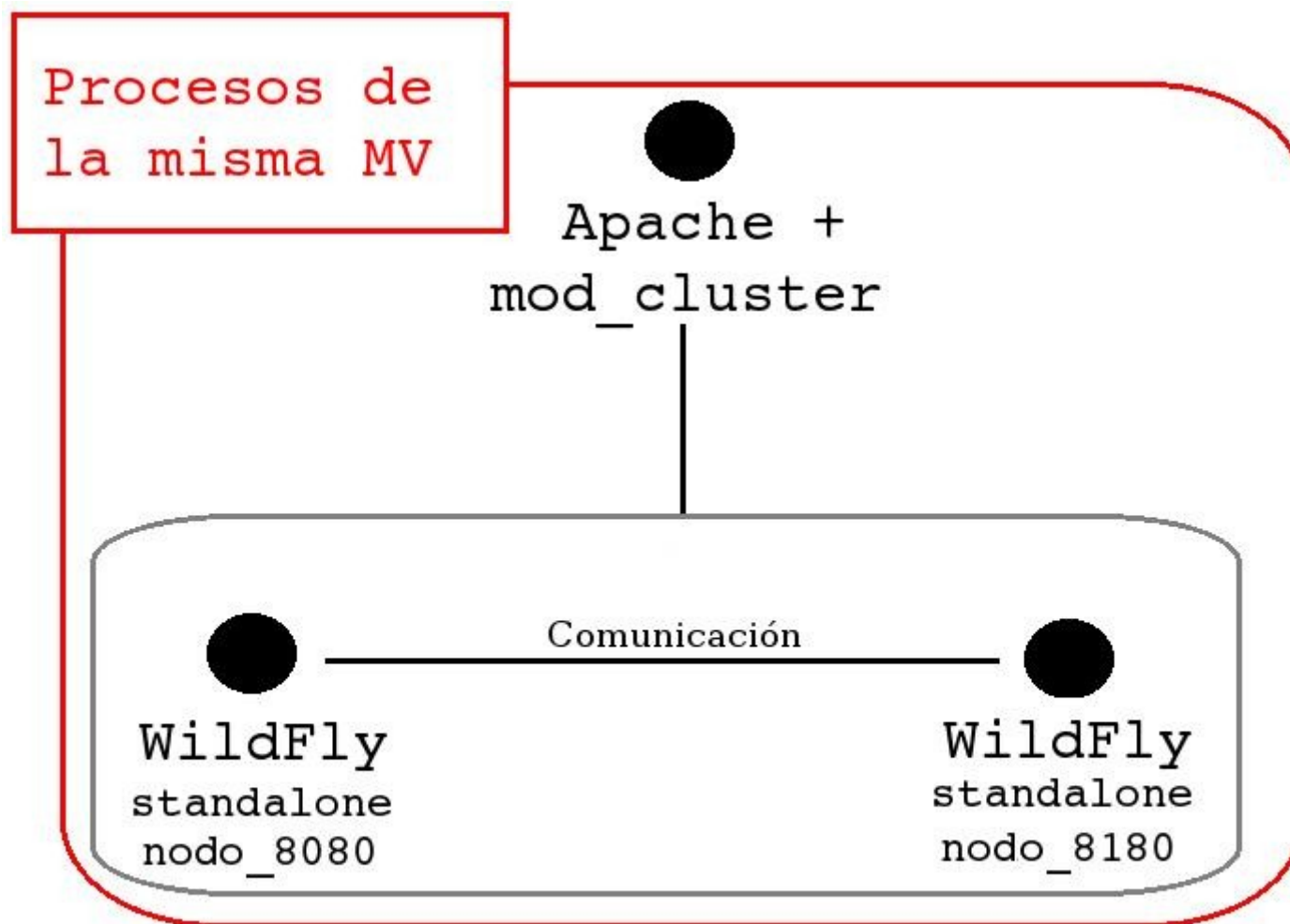
Estructura de un cluster (III)

- Podemos ver dos ordenadores (nodo1 y nodo2) que ejecutan un cluster standalone de WildFly.
 - Los nodos del cluster se comunican entre ellos cuando despliegan aplicaciones “distribuibles” (con la etiqueta “<distributable/>”).
- Se puede acceder **individualmente** a cada nodo del cluster a través del puerto de escucha del propio servidor de aplicaciones mediante las URLs:
 - `http://nodo1:8080/aplicacion_distribuible`
 - `http://nodo2:8080/aplicacion_distribuible`

Estructura de un cluster (IV)

- Existe un tercer ordenador (nodo_acceso) que ejecuta un servidor web Apache.
 - Necesita el módulo mod_cluster para poder acceder a uno u otro nodo del cluster.
- Para acceder al cluster de WildFly se debe acceder a través del puerto 80 del servidor web Apache.
 - El módulo mod_cluster decide a qué nodo realiza la petición.
- El acceso se realiza con la URL:
 - http://nodo_acceso/aplicacion_distribuable
- El usuario no sabe a qué nodo está accediendo.

Escenario simplificado (I)



Escenario simplificado (II)

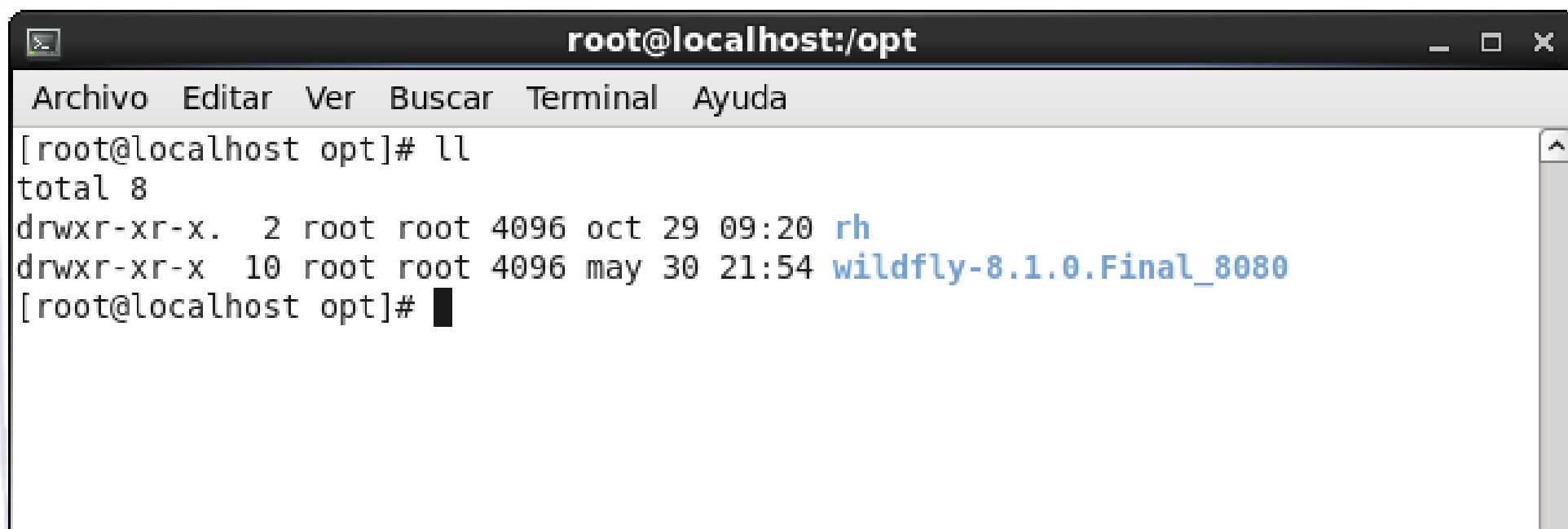
- Vamos a utilizar un único ordenador (máquina virtual) para simular el cluster.
- Utilizaremos dos servidores de aplicaciones WildFly:
 - Un servidor de aplicaciones utilizará por defecto el conjunto de puertos 8080 (nodo_8080).
 - El otro servidor de aplicaciones utilizará por defecto el conjunto de puertos 8180 (nodo_8180).
 - Todos sus puertos (management, ajp, etc.) estarán desplazados 100 unidades.
- Además, configuraremos un servidor Apache para que se comuniquen con ambos nodos.

Escenario simplificado (III)

- En nuestro escenario simplificado debemos realizar las siguientes tareas:
 - Instalar y configurar el WildFly al que llamaremos `nodo_8080`. Desplegar aplicación distribuable y comprobar su funcionamiento individual.
 - Instalar y configurar el WildFly al que llamaremos `nodo_8180`. Desplegar aplicación distribuable y comprobar su funcionamiento individual.
 - Comprobar adicionalmente la comunicación entre ambos nodos.
 - Instalar y configurar el servidor web Apache y el módulo `mod_cluster`. Comprobar el correcto funcionamiento del cluster.

Modo standalone nodo_8080 (I)

- Descomprimir el wildfly-8.1.0 en la carpeta /opt y renombrarlo para diferenciarlo del que instalaremos después.

A terminal window titled 'root@localhost:/opt' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the command 'll' being executed, resulting in a directory listing for the /opt directory. The listing shows two entries: a directory 'rh' and a file 'wildfly-8.1.0.Final_8080'.

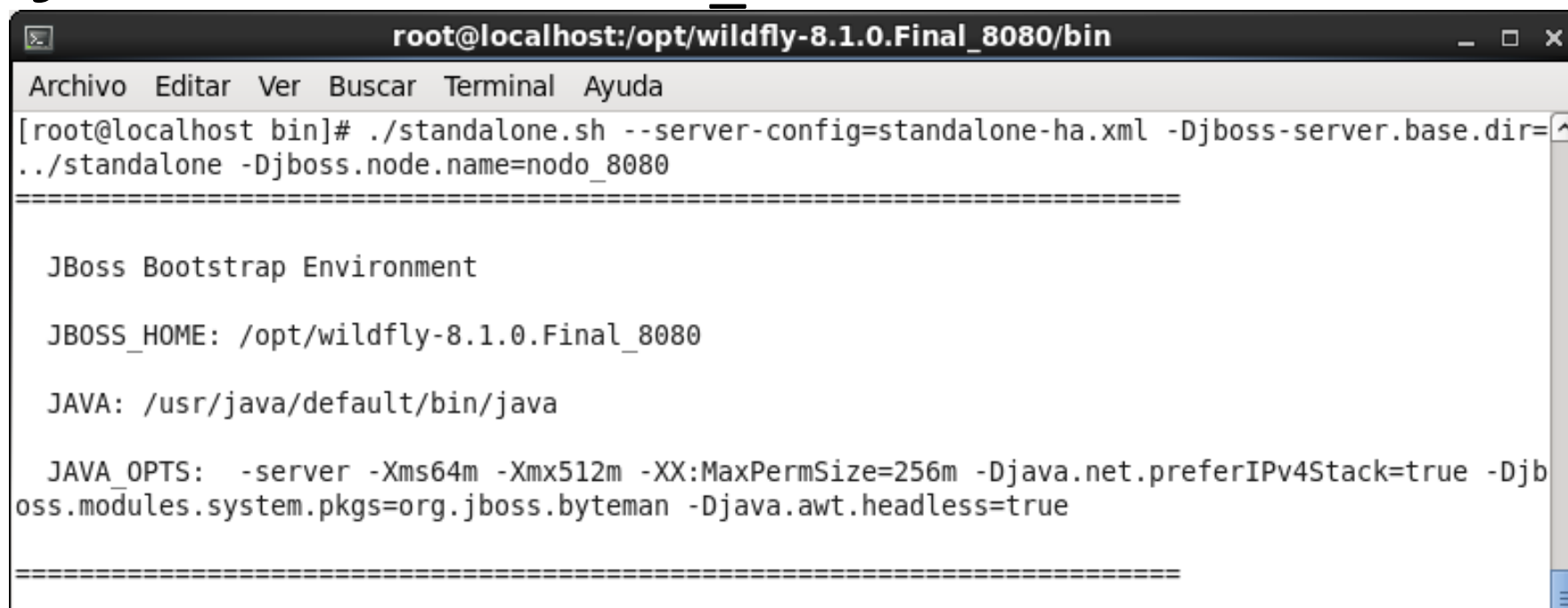
```
root@localhost:/opt
Archivo  Editar  Ver     Buscar  Terminal  Ayuda
[root@localhost opt]# ll
total 8
drwxr-xr-x.  2 root root 4096 oct 29 09:20 rh
drwxr-xr-x 10 root root 4096 may 30 21:54 wildfly-8.1.0.Final_8080
[root@localhost opt]#
```

Modo standalone nodo_8080 (II)

- Lanzar un terminal y ejecutar el comando:

```
cd /opt/wildfly-8.1.0.Final_8080/bin
```

```
./standalone.sh --server-config=standalone-  
ha.xml -Djboss.server.base.dir=../standalone  
-Djboss.node.name=nodo_8080
```



The screenshot shows a terminal window titled "root@localhost:/opt/wildfly-8.1.0.Final_8080/bin". The terminal has a menu bar with "Archivo", "Editar", "Ver", "Buscar", "Terminal", and "Ayuda". The command prompt is "[root@localhost bin]#". The command entered is `./standalone.sh --server-config=standalone-ha.xml -Djboss.server.base.dir=../standalone -Djboss.node.name=nodo_8080`. The output shows the JBoss Bootstrap Environment with the following variables: `JBOSSE_HOME: /opt/wildfly-8.1.0.Final_8080`, `JAVA: /usr/java/default/bin/java`, and `JAVA_OPTS: -server -Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true`. The terminal output is separated by lines of equals signs.

```
root@localhost:/opt/wildfly-8.1.0.Final_8080/bin
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
[root@localhost bin]# ./standalone.sh --server-config=standalone-ha.xml -Djboss.server.base.dir=
../standalone -Djboss.node.name=nodo_8080
=====

JBoss Bootstrap Environment

JBOSSE_HOME: /opt/wildfly-8.1.0.Final_8080

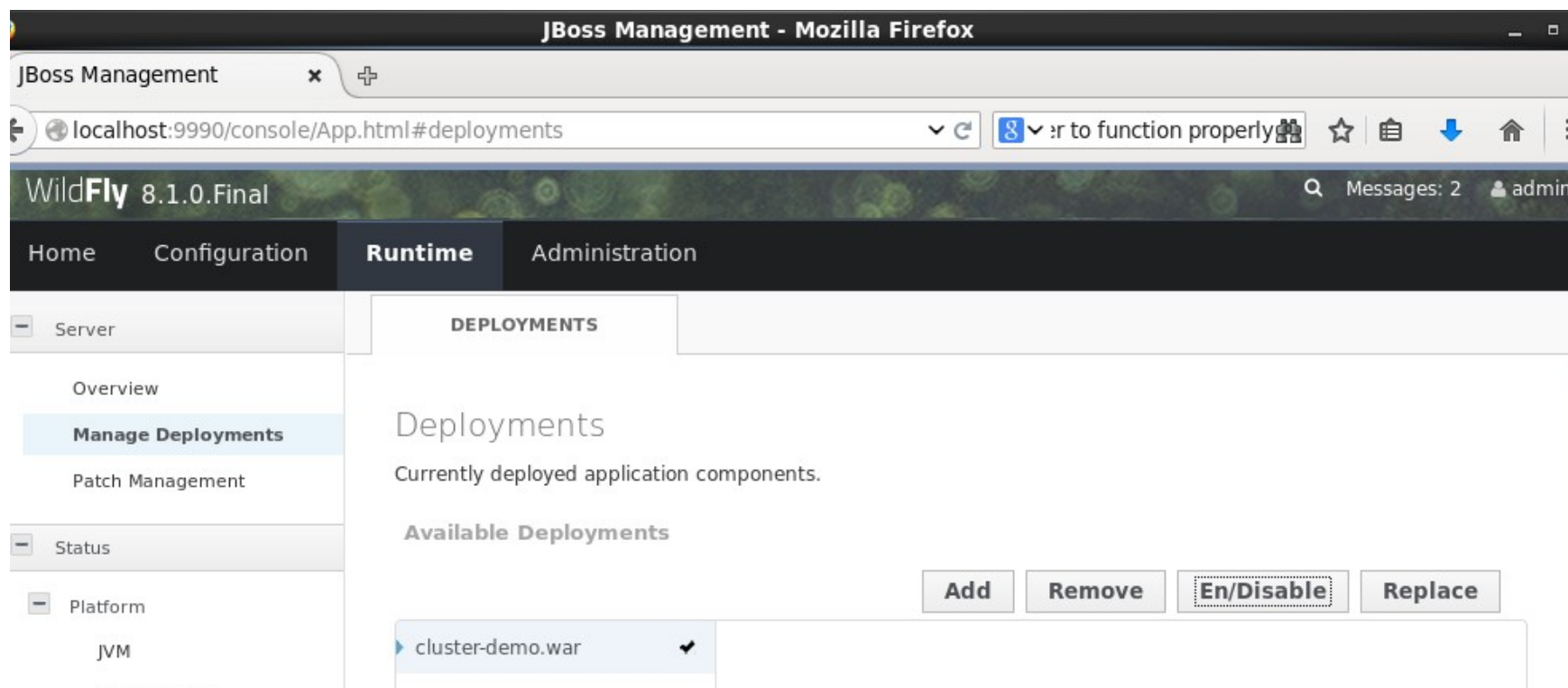
JAVA: /usr/java/default/bin/java

JAVA_OPTS:  -server -Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Djb
oss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true
=====
```

Modo standalone nodo_8080 (III)

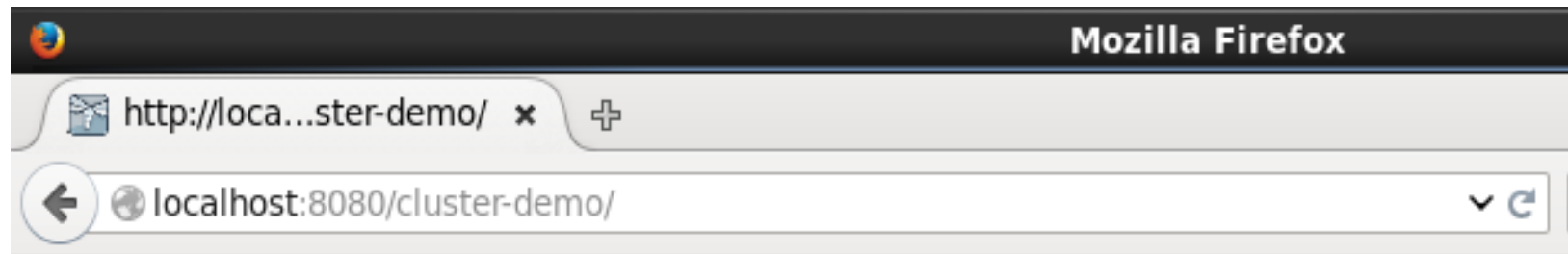
- Desplegar la aplicación distribuable cluster-demo.war.

NOTA: Debe crearse un usuario para acceder al Administration Console.



Modo standalone nodo_8080 (IV)

- Acceder a la aplicación a través del puerto 8080.

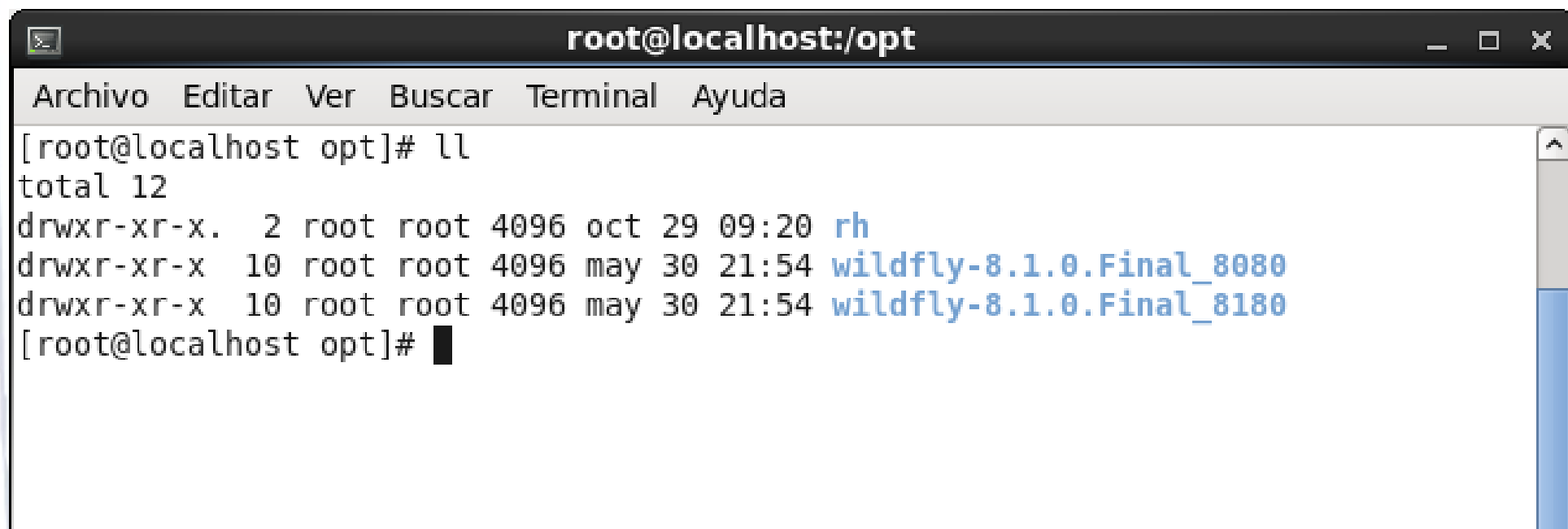


Hello World!

Hello! The time is now Wed Nov 12 01:40:09 CET 2014

Modo standalone nodo_8180 (I)

- Dejar ejecutándose el nodo_8080.
- Descomprimir el wildfly-8.1.0 en la carpeta /opt y renombrarlo para diferenciarlo del nodo_8080.

A terminal window titled 'root@localhost:/opt' with a menu bar containing 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal shows the command 'll' being executed, resulting in a directory listing of three files in the /opt directory. The files are 'rh', 'wildfly-8.1.0.Final_8080', and 'wildfly-8.1.0.Final_8180'. The permissions for all three files are 'drwxr-xr-x'. The file 'rh' has a size of 4096 bytes and was last modified on October 29 at 09:20. The other two files, 'wildfly-8.1.0.Final_8080' and 'wildfly-8.1.0.Final_8180', also have a size of 4096 bytes and were last modified on May 30 at 21:54. The terminal prompt is '[root@localhost opt]#'.

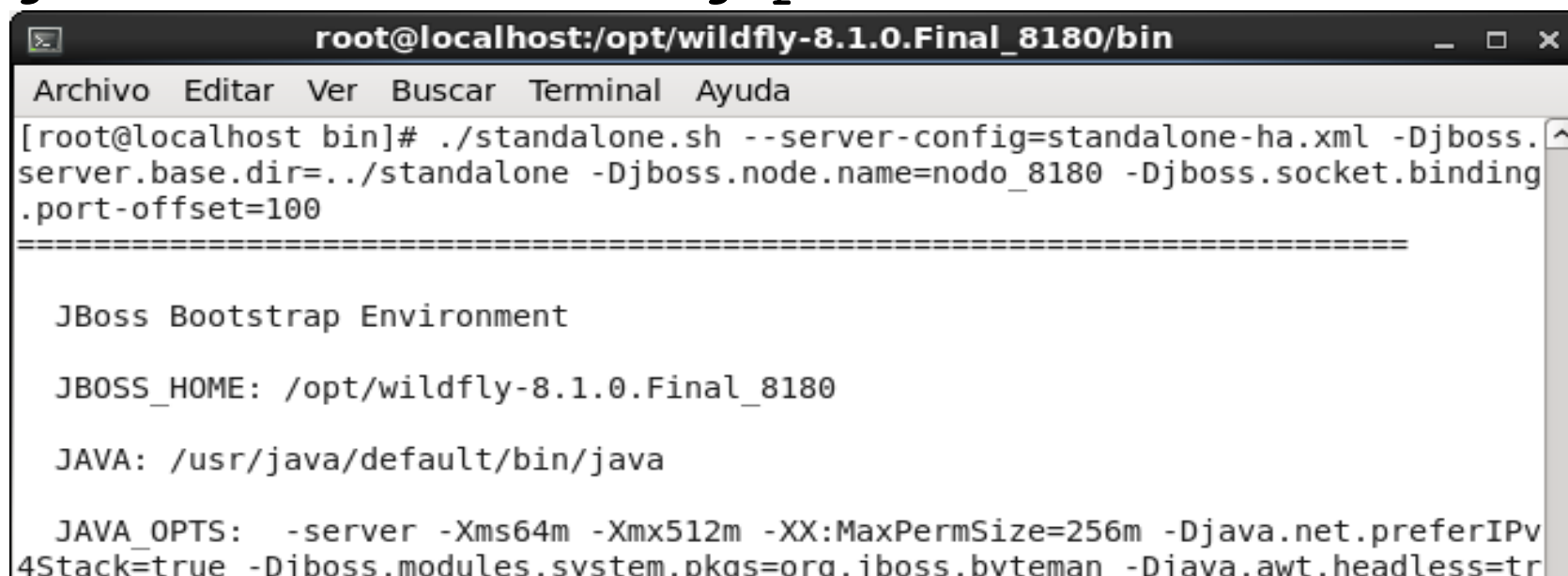
```
root@localhost:/opt
Archivo  Editar  Ver     Buscar  Terminal  Ayuda
[root@localhost opt]# ll
total 12
drwxr-xr-x.  2 root root 4096 oct 29 09:20 rh
drwxr-xr-x  10 root root 4096 may 30 21:54 wildfly-8.1.0.Final_8080
drwxr-xr-x  10 root root 4096 may 30 21:54 wildfly-8.1.0.Final_8180
[root@localhost opt]#
```

Modo standalone nodo_8180 (II)

- Lanzar un terminal y ejecutar el comando:

```
cd /opt/wildfly-8.1.0.Final_8180/bin
```

```
./standalone.sh --server-config=standalone-  
ha.xml -Djboss.server.base.dir=../standalone  
-Djboss.node.name=nodo_8180  
-Djboss.socket.binding.port-offset=100
```



The screenshot shows a terminal window titled 'root@localhost:/opt/wildfly-8.1.0.Final_8180/bin'. The terminal has a menu bar with 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The command entered is `./standalone.sh --server-config=standalone-ha.xml -Djboss.server.base.dir=../standalone -Djboss.node.name=nodo_8180 -Djboss.socket.binding.port-offset=100`. The output shows the JBoss Bootstrap Environment with the following variables: `JBOSS_HOME: /opt/wildfly-8.1.0.Final_8180`, `JAVA: /usr/java/default/bin/java`, and `JAVA_OPTS: -server -Xms64m -Xmx512m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true`.

Modo standalone nodo_8180 (III)

- Desplegar la aplicación distribuable cluster-demo.war.

NOTA: Debe crearse un usuario para acceder al Administration Console.

JBoss Management - Mozilla Firefox

localhost:10090/console/App.html#deployments

WildFly 8.1.0.Final

Home Configuration **Runtime** Administration

Server

Overview

Manage Deployments

Patch Management

Status

Platform

JVM

Environment

DEPLOYMENTS

Deployments

Currently deployed application components.

Available Deployments

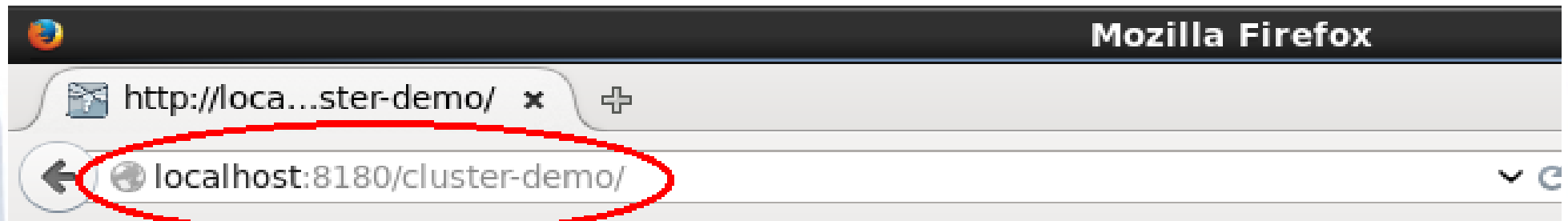
cluster-demo.war

Add Remove En/Disable Replace

Desplazamiento 100 en el puerto!!

Modo standalone nodo_8180 (IV)

- Acceder a la aplicación a través del puerto 8180.



Hello World!

Hello! The time is now Wed Nov 12 01:43:48 CET 2014

Modo standalone nodo_8180 (V)

- Comprobar en la terminal del nodo_8080 como ha “encontrado” al otro nodo que forma el cluster de WildFly.

NOTA: La comunicación no ocurre hasta que no se despliega la misma aplicación distribuible en ambos nodos.

```
web context: /cluster-demo
01:54:30,615 INFO  [org.jboss.as.server] (Controller Boot Thread) JBAS018559: Deployed "cluster-demo.war" (runtime-name : "cluster-demo.war")
01:54:30,637 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015961: Http management interface listening on http://127.0.0.1:9990/management
01:54:30,637 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin console listening on http://127.0.0.1:9990
01:54:30,638 INFO  [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.1.0.Final "Kenny" started in 10151ms - Started 289 of 400 services (179 services are lazy, passive or on-demand)
01:56:03,874 INFO  [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (Incoming-1,shared=udp) ISPN000094: Received new cluster view: [nodo_8080/web|1] (2) [nodo_8080/web, nodo_8180/web]
```

Apache + mod_cluster (I)

- Instalar el servidor de apache (si no está instalado) con el comando:

```
yum install httpd
```

- Instalar el módulo mod_cluster.

```
yum install mod_cluster-1.2.6-3.el6.i686.rpm
```


Apache + mod_cluster (II)

- Configurar el fichero `/etc/httpd/conf/httpd.conf`:
 - Seleccionar el puerto para el monitor del `mod_cluster`, por ejemplo el puerto 10001.
 - A través de este monitor se puede observar los nodos de WildFly reconocidos por el apache.
 - Configurar parámetros de comunicación entre el servidor web Apache y los nodos de WildFly.
 - Es importante el parámetro `AdvertiseSecurityKey` porque el valor que le asignemos deberemos añadirlo en los WildFly.
 - Es necesario eliminar la carga del módulo `mod_proxy_balancer` para poder utilizar el `mod_cluster`.

Apache + mod_cluster (III)



```
root@localhost:/etc/httpd/conf
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
# ErrorLog logs/dummy-host.example.com-error_log
# CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>

Listen 127.0.0.1:10001
MemManagerFile /var/cache/httpd

<VirtualHost 127.0.0.1:10001>

    <Directory />
        Order allow,deny
        Allow from all
    </Directory>

    KeepAliveTimeout 60
    MaxKeepAliveRequests 0

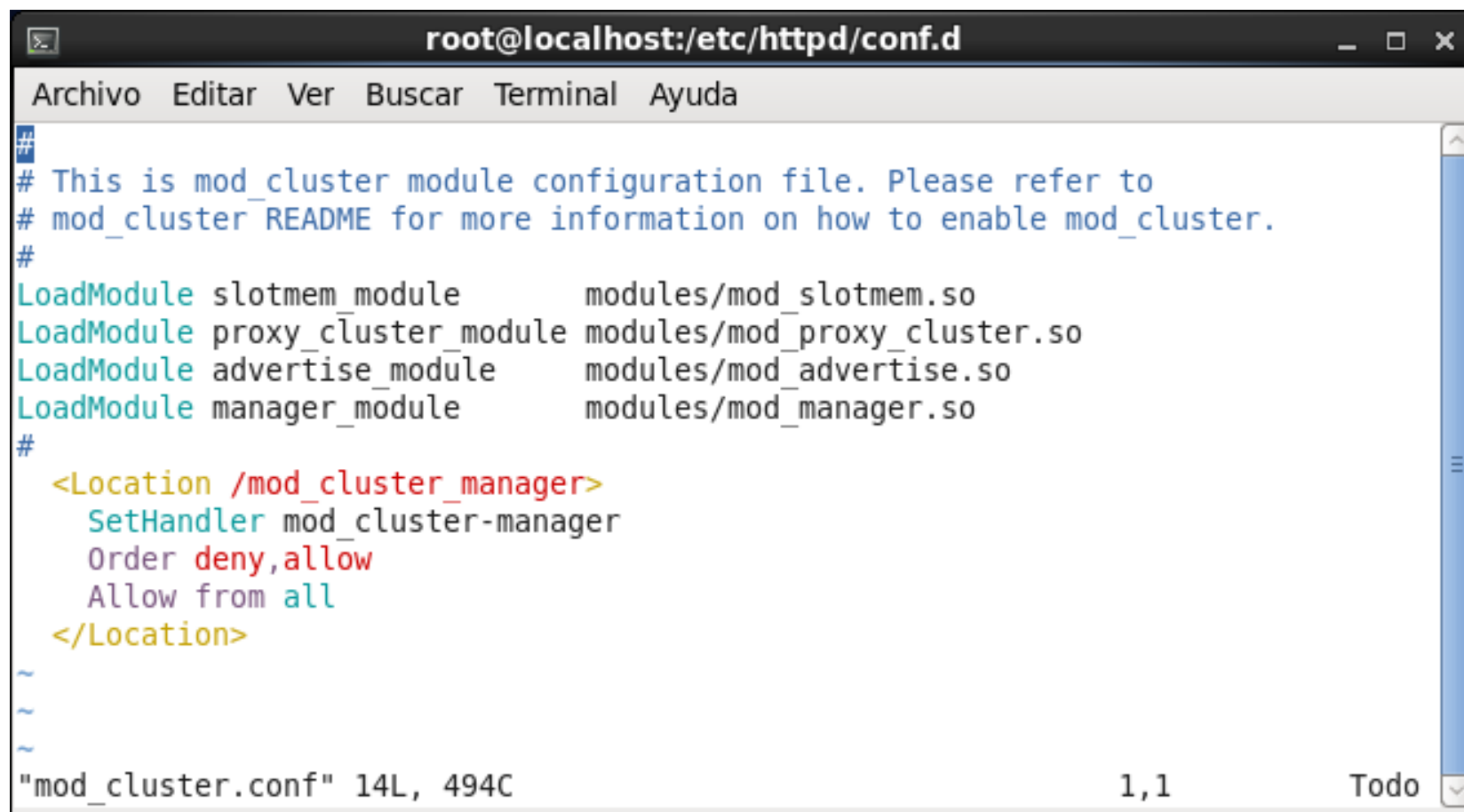
    AdvertiseFrequency 5
    ServerAdvertise On
    EnableMCPMReceive
    AdvertiseSecurityKey secreto

</VirtualHost>

1008,1  Final
```

Apache + mod_cluster (IV)

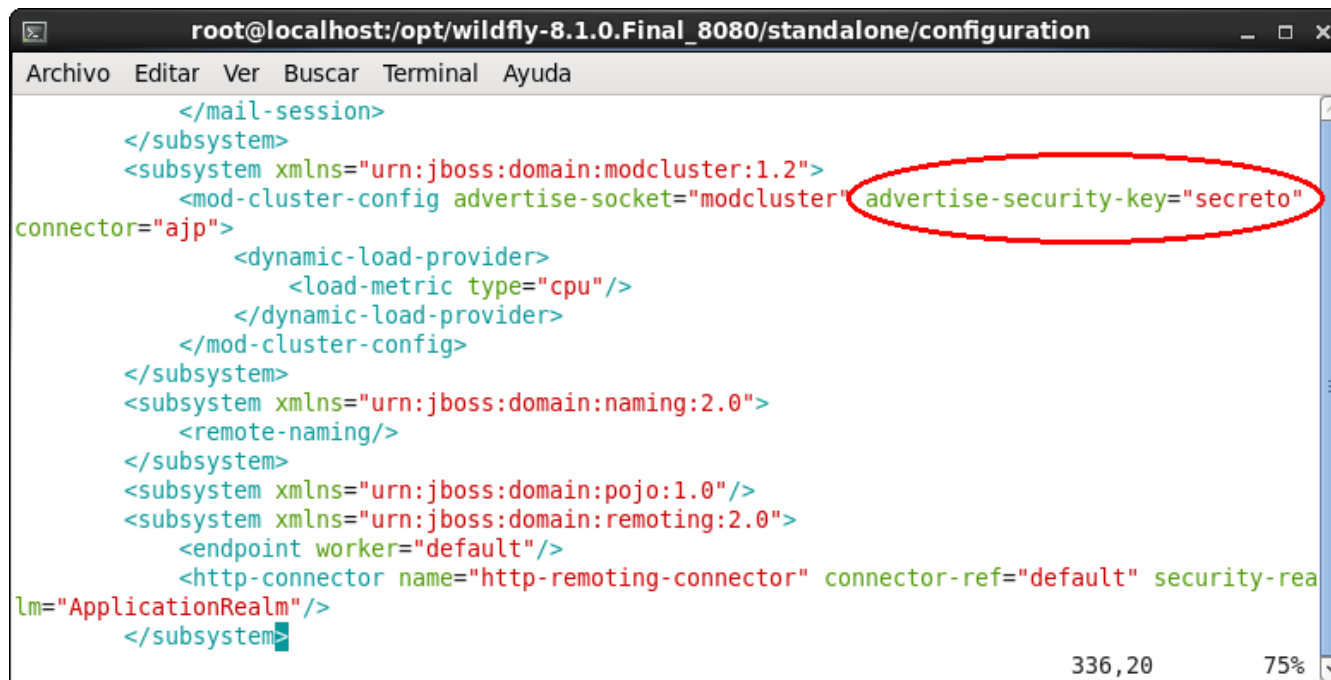
- Editar el fichero `/etc/httpd/conf.d/mod_cluster.conf`



```
root@localhost:/etc/httpd/conf.d
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
#
# This is mod_cluster module configuration file. Please refer to
# mod_cluster README for more information on how to enable mod_cluster.
#
LoadModule slotmem_module      modules/mod_slotmem.so
LoadModule proxy_cluster_module modules/mod_proxy_cluster.so
LoadModule advertise_module    modules/mod_advertise.so
LoadModule manager_module      modules/mod_manager.so
#
<Location /mod_cluster_manager>
    SetHandler mod_cluster-manager
    Order deny,allow
    Allow from all
</Location>
~
~
~
"mod_cluster.conf" 14L, 494C      1,1      Todo
```

Apache + mod_cluster (V)

- Añadir en los ficheros standalone-ha.xml de los WildFly, en la sección mod-cluster-config el parámetro advertise-security-key="secreto" (donde "secreto" es el valor del parámetro AdvertiseSecurityKey del apache)



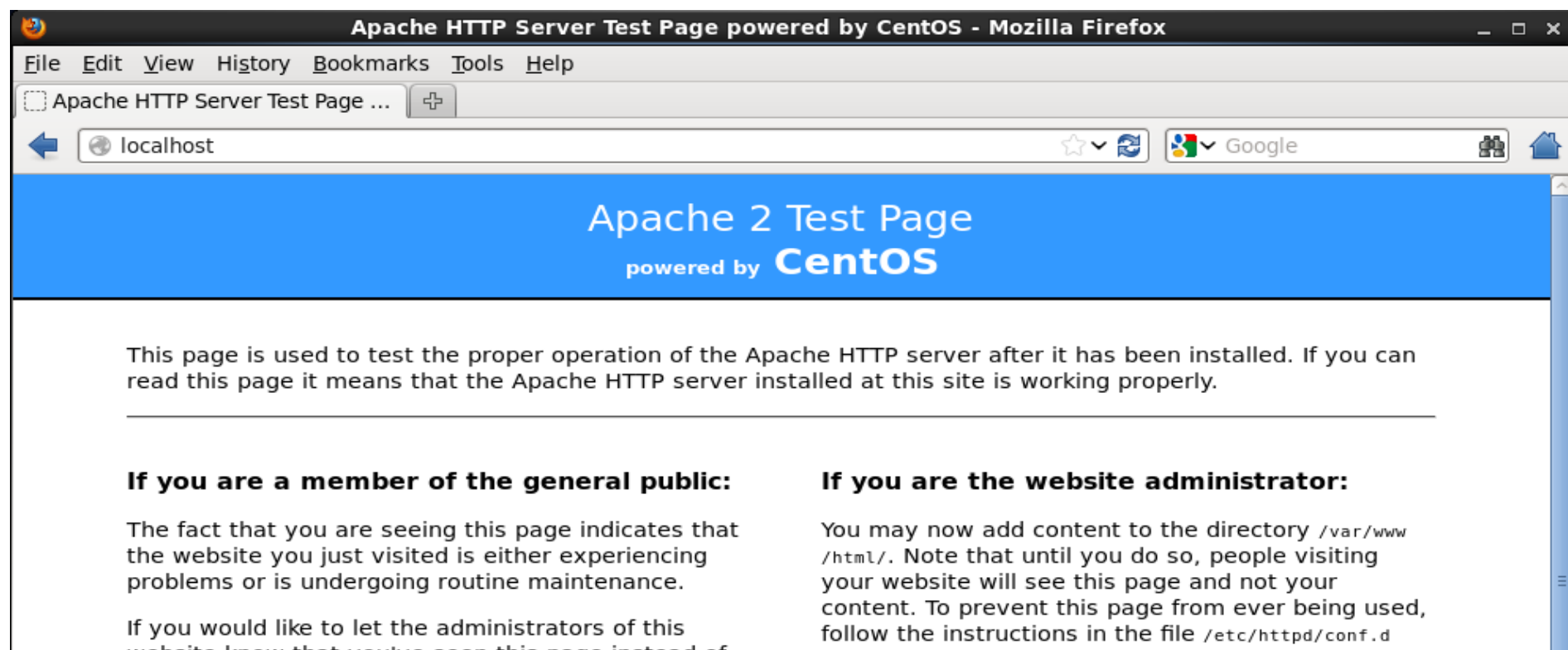
```
root@localhost:/opt/wildfly-8.1.0.Final_8080/standalone/configuration
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
</mail-session>
</subsystem>
<subsystem xmlns="urn:jboss:domain:modcluster:1.2">
  <mod-cluster-config advertise-socket="modcluster" advertise-security-key="secreto"
connector="ajp">
    <dynamic-load-provider>
      <load-metric type="cpu"/>
    </dynamic-load-provider>
  </mod-cluster-config>
</subsystem>
<subsystem xmlns="urn:jboss:domain:naming:2.0">
  <remote-naming/>
</subsystem>
<subsystem xmlns="urn:jboss:domain:pojo:1.0"/>
<subsystem xmlns="urn:jboss:domain:remoting:2.0">
  <endpoint worker="default"/>
  <http-connector name="http-remoting-connector" connector-ref="default" security-realm="ApplicationRealm"/>
</subsystem>
```

Apache + mod_cluster (VI)

- Ejecutar el servidor apache con el comando:

```
service httpd start
```

- Comprobar que funciona.



Comprobación del cluster (I)

- Reiniciar los dos nodos de WildFly.
- Comprobar que Apache reconoce los dos nodos de WildFly.

Mod_cluster Status - Mozilla Firefox

localhost:10001/mod_cluster_manager

localhost:10001/mod_cluster_manager

mod_cluster/1.2.6.Final

[Auto Refresh](#) [show DUMP output](#) [show INFO output](#)

Node nodo_8080 (ajp://127.0.0.1:8009):

[Enable Contexts](#) [Disable Contexts](#)

Balancer: mycluster,LBGroup: ,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 1,Ttl: 60000000,Status: OK,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 97

Virtual Host 1:

Contexts:

/cluster-demo, Status: ENABLED Request: 0 [Disable](#)

Aliases:

default-host
localhost

Node nodo_8180 (ajp://127.0.0.1:8109):

[Enable Contexts](#) [Disable Contexts](#)

Balancer: mycluster,LBGroup: ,Flushpackets: Off,Flushwait: 10000,Ping: 10000000,Smax: 1,Ttl: 60000000,Status: OK,Elected: 0,Read: 0,Transferred: 0,Connected: 0,Load: 97

Virtual Host 1:

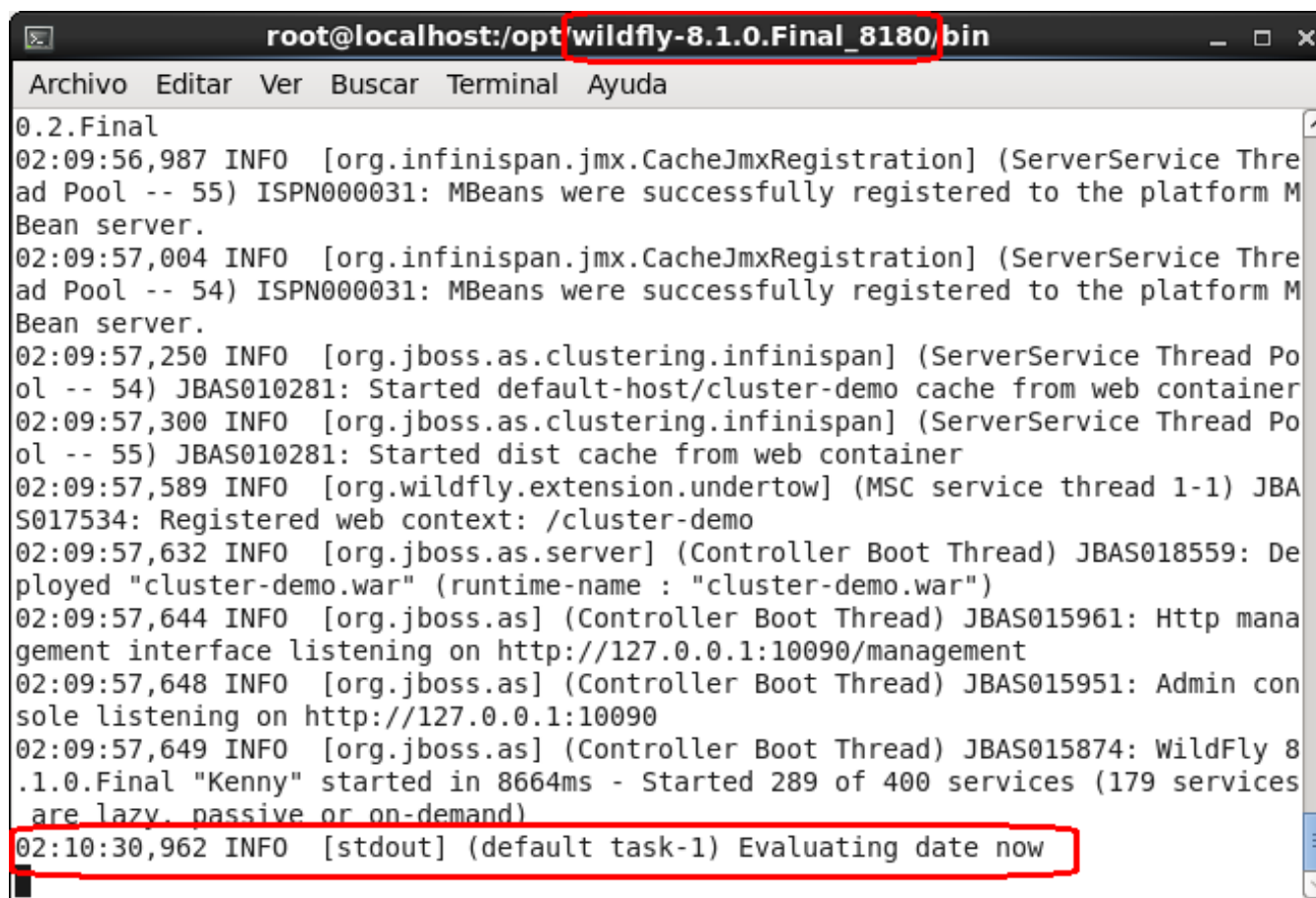
Comprobación del cluster (II)

- Comprobar que la aplicación es accesible a través del servidor web Apache.



Comprobación del cluster (III)

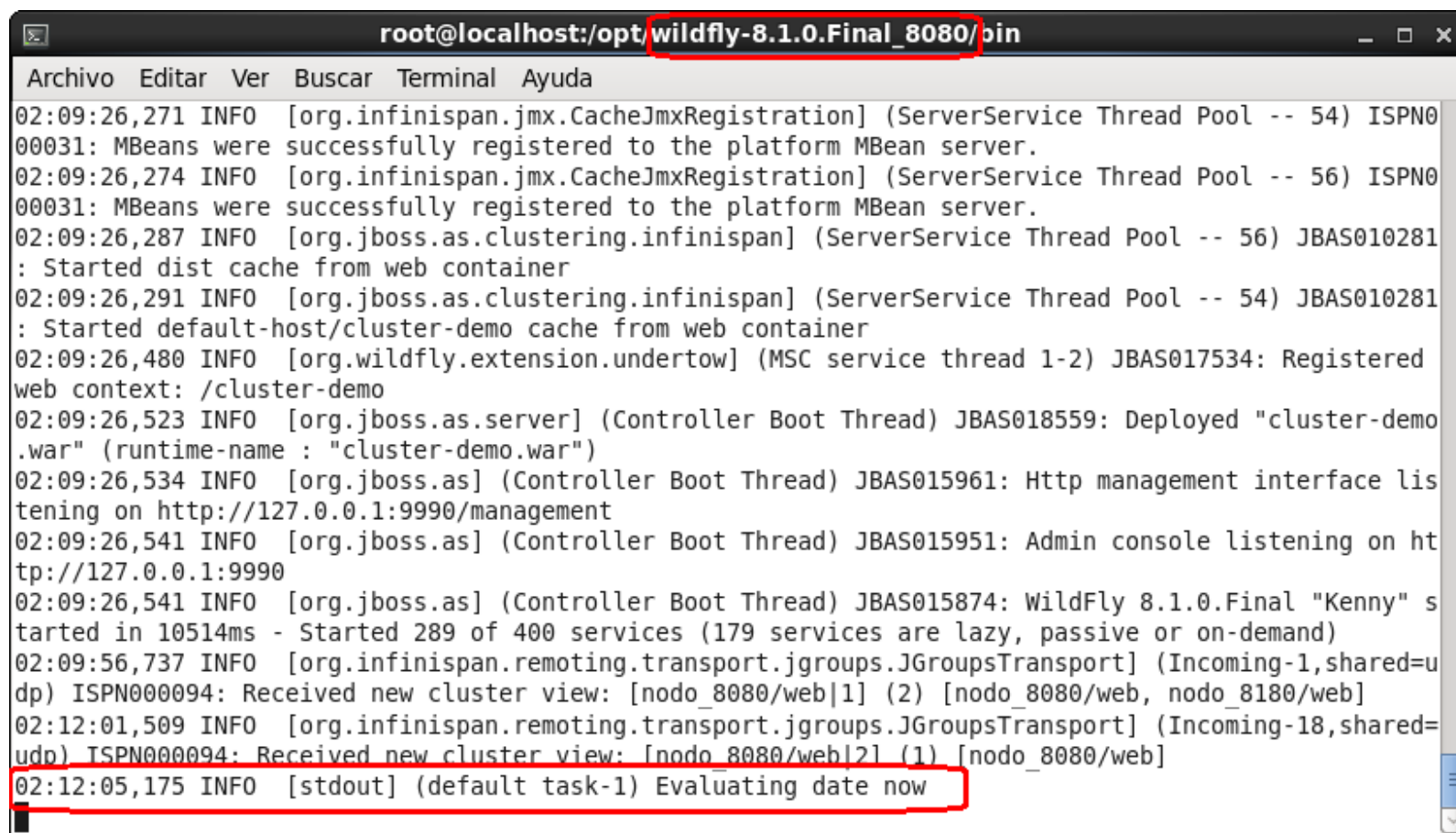
- Se puede comprobar qué nodo de WildFly está atendiendo las peticiones viendo la consola de ejecución.



```
root@localhost:/opt/wildfly-8.1.0.Final_8180/bin
Archivo Editar Ver Buscar Terminal Ayuda
0.2.Final
02:09:56,987 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService Thread Pool -- 55) ISPN000031: MBeans were successfully registered to the platform MBean server.
02:09:57,004 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService Thread Pool -- 54) ISPN000031: MBeans were successfully registered to the platform MBean server.
02:09:57,250 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 54) JBAS010281: Started default-host/cluster-demo cache from web container
02:09:57,300 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 55) JBAS010281: Started dist cache from web container
02:09:57,589 INFO [org.wildfly.extension.undertow] (MSC service thread 1-1) JBAS017534: Registered web context: /cluster-demo
02:09:57,632 INFO [org.jboss.as.server] (Controller Boot Thread) JBAS018559: Deployed "cluster-demo.war" (runtime-name : "cluster-demo.war")
02:09:57,644 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Http management interface listening on http://127.0.0.1:10090/management
02:09:57,648 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin console listening on http://127.0.0.1:10090
02:09:57,649 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.1.0.Final "Kenny" started in 8664ms - Started 289 of 400 services (179 services are lazy, passive or on-demand)
02:10:30,962 INFO [stdout] (default task-1) Evaluating date now
```


Comprobación del cluster (IV)

- Detener el WildFly que responde a las peticiones y comprobar que el `mod_cluster` automáticamente redirige las peticiones al otro nodo.



```
root@localhost:/opt/wildfly-8.1.0.Final_8080/bin
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
02:09:26,271 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService Thread Pool -- 54) ISPN00031: MBeans were successfully registered to the platform MBean server.
02:09:26,274 INFO [org.infinispan.jmx.CacheJmxRegistration] (ServerService Thread Pool -- 56) ISPN00031: MBeans were successfully registered to the platform MBean server.
02:09:26,287 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 56) JBAS010281: Started dist cache from web container
02:09:26,291 INFO [org.jboss.as.clustering.infinispan] (ServerService Thread Pool -- 54) JBAS010281: Started default-host/cluster-demo cache from web container
02:09:26,480 INFO [org.wildfly.extension.undertow] (MSC service thread 1-2) JBAS017534: Registered web context: /cluster-demo
02:09:26,523 INFO [org.jboss.as.server] (Controller Boot Thread) JBAS018559: Deployed "cluster-demo.war" (runtime-name : "cluster-demo.war")
02:09:26,534 INFO [org.jboss.as] (Controller Boot Thread) JBAS015961: Http management interface listening on http://127.0.0.1:9990/management
02:09:26,541 INFO [org.jboss.as] (Controller Boot Thread) JBAS015951: Admin console listening on http://127.0.0.1:9990
02:09:26,541 INFO [org.jboss.as] (Controller Boot Thread) JBAS015874: WildFly 8.1.0.Final "Kenny" started in 10514ms - Started 289 of 400 services (179 services are lazy, passive or on-demand)
02:09:56,737 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (Incoming-1,shared=udp) ISPN000094: Received new cluster view: [nodo_8080/web|1] (2) [nodo_8080/web, nodo_8180/web]
02:12:01,509 INFO [org.infinispan.remoting.transport.jgroups.JGroupsTransport] (Incoming-18,shared=udp) ISPN000094: Received new cluster view: [nodo_8080/web|2] (1) [nodo_8080/web]
02:12:05,175 INFO [stdout] (default task-1) Evaluating date now
```