

 UNIVERSIDAD AUTONOMA DE MADRID	Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2				
Grupo	2311	Práctica	3	Fecha	08/05/2022
Alumno/a	Fraile, Iglesias, Javier				
Alumno/a	Fernández, París, Iván				

Práctica 3: Seguridad y disponibilidad

Ejercicio número 1:

Preparar 3 máquinas virtuales desde cero (a partir de la VM en moodle) con acceso SSH entre ellas. Esta tarea es necesaria para la correcta gestión del cluster que definiremos en el próximo apartado. Las VMs las denominaremos:

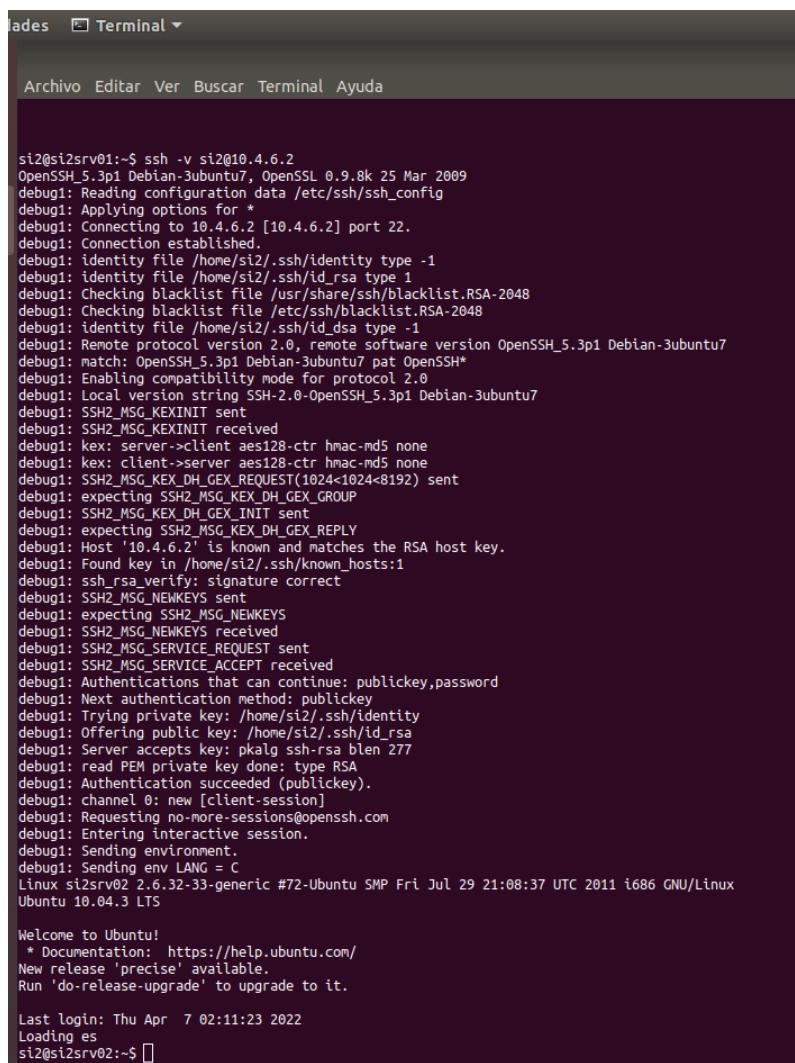
- si2srv01: Dirección IP 10.X.Y.1, 768MB RAM
- si2srv02: Dirección IP 10.X.Y.2, 512MB RAM
- si2srv03: Dirección IP 10.X.Y.3, 512MB RAM

RECUERDE RANDOMIZAR LAS DIRECCIONES MAC DE CADA COPIA ANTES DE INTENTAR USAR EL NODO.

En la primera máquina (10.X.Y.1), generamos el par de claves con RSA. A continuación importamos la clave pública en cada uno de los otros dos nodos (10.X.Y.2 y 10.X.Y.3). Probaremos a acceder por SSH desde .1 a .2 y .3, comprobando que no requiere la introducción de la clave. Obtener una evidencia del inicio remoto de sesión mediante la salida detallada (ssh -v si2@10.X.Y.2 y ssh -v si2@10.X.Y.3). Anote dicha salida en la memoria de prácticas.

Para entrar en las máquinas 2 y 3 desde la máquina 1, se han seguido los pasos, copiando la clave pública en ambos hosts, y se ha iniciado una conexión ssh desde el terminal a la máquina 1.

A continuación, se ha intentado establecer conexión a la máquina 2:



```
lades Terminal ▾
Archivo Editar Ver Buscar Terminal Ayuda

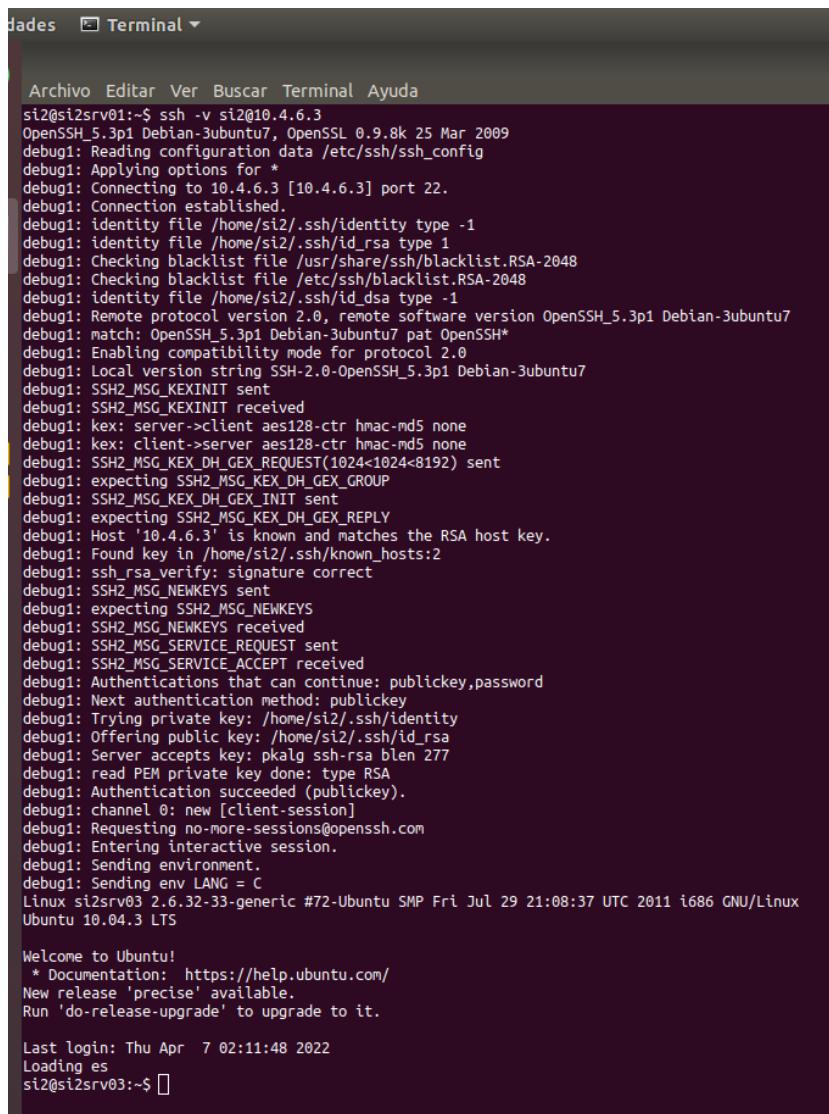
si2@si2srv01:~$ ssh -v si2@10.4.6.2
OpenSSH_5.3p1 Debian-3ubuntu7, OpenSSL 0.9.8k 25 Mar 2009
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Applying options for *
debug1: Connecting to 10.4.6.2 [10.4.6.2] port 22.
debug1: Connection established.
debug1: identity file /home/si2/.ssh/identity type -1
debug1: identity file /home/si2/.ssh/id_rsa type 1
debug1: Checking blacklist file /usr/share/ssh/blacklist.RSA-2048
debug1: Checking blacklist file /etc/ssh/blacklist.RSA-2048
debug1: identity file /home/si2/.ssh/id_dsa type -1
debug1: Remote protocol version 2.0, remote software version OpenSSH_5.3p1 Debian-3ubuntu7
debug1: match: OpenSSH_5.3p1 Debian-3ubuntu7 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-ctr hmac-md5 none
debug1: kex: client->server aes128-ctr hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(1024<1024<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Host '10.4.6.2' is known and matches the RSA host key.
debug1: Found key in /home/si2/.ssh/known_hosts:1
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Trying private key: /home/si2/.ssh/identity
debug1: Offering public key: /home/si2/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-rsa blen 277
debug1: read PEM private key done: type RSA
debug1: Authentication succeeded (publickey).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending env LANG = C
Linux si2srv02 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:37 UTC 2011 i686 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Apr  7 02:11:23 2022
Loading es
si2@si2srv02:~$
```

Las líneas más resaltantes para entender el proceso de autenticación automática con clave son la que dice "Host '10.4.6.2', es conocido, ya se encuentra en known_hosts", y además nos dice que la autenticación con RSA coincide.

En cuanto al host 3, el procedimiento es exactamente el mismo:



```
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ ssh -v si2@10.4.6.3
OpenSSH_5.3p1 Debian-3ubuntu7, OpenSSL 0.9.8k 25 Mar 2009
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: Applying options for *
debug1: Connecting to 10.4.6.3 [10.4.6.3] port 22.
debug1: Connection established.
debug1: identity file /home/si2/.ssh/identity type -1
debug1: identity file /home/si2/.ssh/id_rsa type 1
debug1: Checking blacklist file /usr/share/ssh/blacklist.RSA-2048
debug1: Checking blacklist file /etc/ssh/blacklist.RSA-2048
debug1: identity file /home/si2/.ssh/id_dsa type -1
debug1: Remote protocol version 2.0, remote software version OpenSSH_5.3p1 Debian-3ubuntu7
debug1: match: OpenSSH_5.3p1 Debian-3ubuntu7 pat OpenSSH*
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu7
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: server->client aes128-ctr hmac-md5 none
debug1: kex: client->server aes128-ctr hmac-md5 none
debug1: SSH2_MSG_KEX_DH_GEX_REQUEST(1024<1024<8192) sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_GROUP
debug1: SSH2_MSG_KEX_DH_GEX_INIT sent
debug1: expecting SSH2_MSG_KEX_DH_GEX_REPLY
debug1: Host '10.4.6.3' is known and matches the RSA host key.
debug1: Found key in /home/si2/.ssh/known_hosts:2
debug1: ssh_rsa_verify: signature correct
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: SSH2_MSG_SERVICE_REQUEST sent
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Trying private key: /home/si2/.ssh/identity
debug1: Offering public key: /home/si2/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-rsa blen 277
debug1: read PEM private key done: type RSA
debug1: Authentication succeeded (publickey).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: Sending environment.
debug1: Sending env LANG = C
Linux si2srv03 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:37 UTC 2011 i686 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation:  https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Thu Apr  7 02:11:48 2022
Loading es
si2@si2srv03:~$ 
```

Ejercicio número 2:

Realizar los pasos del apartado 4 con el fin de obtener una configuración válida del cluster SI2Cluster, con la tipología indicada de 1 DAS y 2 nodos SSH de instancias. Inicie el cluster. Liste las instancias del cluster y verifique que los pids de los procesos Java (JVM) correspondientes² están efectivamente corriendo en cada una de las dos máquinas virtuales. Adjunte evidencias a la memoria de la práctica.

Se han ejecutado los comandos de crear los nodos desde el host 1, y en nuestro caso la contraseña era la de por defecto:

```

si2@si2srv01:~$ asadmin start-domain domain1
Waiting for domain1 to start ..
Successfully started the domain : domain1
domain Location: /opt/glassfish4/glassfish/domains/domain1
Log File: /opt/glassfish4/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
si2@si2srv01:~$ asadmin create-node-ssh --sshuser si2 --nodehost 10.4.6.2 --nodedir /opt/glassfish4 Node01
Enter admin user name> admin
Enter admin password for user "admin">
Command create-node-ssh executed successfully.
si2@si2srv01:~$ asadmin create-node-ssh --sshuser si2 --nodehost 10.4.6.3 --nodedir /opt/glassfish4 Node02
Enter admin user name> admin
Enter admin password for user "admin">
Command create-node-ssh executed successfully.
si2@si2srv01:~$ asadmin --user admin list-nodes
Enter admin password for user "admin">
localhost-domain1 CONFIG localhost
Node01 SSH 10.4.6.2
Node02 SSH 10.4.6.3
Command list-nodes executed successfully.
si2@si2srv01:~$ 

```

Como podemos observar, cuando listamos los nodos se obtienen los dos nodos cada uno con su host 10.4.6.2 y 10.4.6.3.

Finalmente, hacemos “ping” de ambos nodos:

```

Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin --user admin ping-node-ssh Node01
Enter admin password for user "admin">
Successfully made SSH connection to node Node01 (10.4.6.2)
Command ping-node-ssh executed successfully.
si2@si2srv01:~$ asadmin --user admin ping-node-ssh Node02
Enter admin password for user "admin">
Successfully made SSH connection to node Node02 (10.4.6.3)
Command ping-node-ssh executed successfully.
si2@si2srv01:~$ 

```

Accedemos al servidor de Glassfish 4 y observamos los nodos, uno por defecto de configuración como contenedor de información que explica el tutorial, y los dos nodos SSH que permiten la gestión descentralizada y/o remota del Servidor:

Select	Name	Node Host	Type	Instances	Action
<input type="checkbox"/>	Node01	10.4.6.2	SSH		Ping
<input type="checkbox"/>	Node02	10.4.6.3	SSH		Ping
<input type="checkbox"/>	localhost-domain1	localhost	CONFIG		

Para la creación del cluster, seguimos el tutorial, podemos ver que el cluster no se encuentra corriendo de momento:

```

Archivos Terminal ▾
Archivo Editar Ver Buscar Terminal Ayuda
Jue 12:03
ssh si2@10.4.6.1
si2@si2srv01:~$ asadmin create-cluster SI2Cluster
Enter admin user name> admin
Enter admin password for user "admin">
Command create-cluster executed successfully.
si2@si2srv01:~$ asadmin list-clusters
Enter admin user name> admin
Enter admin password for user "admin">
SI2Cluster not running
Command list-clusters executed successfully.
si2@si2srv01:~$ 

```

A continuación, comprobamos que exista el fichero “/etc/hosts” y además contengan los nombres de los hosts que compondrán el cluster, y efectivamente tiene constancia de ellos (host 1, 2 y 3 respectivamente):

The image contains three separate terminal windows, each showing the output of the command `cat /etc/hosts`. The windows are titled "jades" (top), "si2srv - VMware Workstation 16 Player" (left), and "si2srv - VMware Workstation 16 Player" (right).

```

jades Terminal
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ cat /etc/hosts
10.4.6.1 si2srv01
10.4.6.2 si2srv02
10.4.6.3 si2srv03
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
si2@si2srv01:~$ 

si2srv - VMware Workstation 16 Player (Non-commercial use only)
File Virtual Machine Help
Direccion nat: 172.16.188.131
Direccion bridge: 10.4.6.2
Inicie sesion con usuario si2
Ubuntu 10.04.3 LTS si2srv02 tty1
si2srv02 login: si2
Password:
Last login: Wed Apr  6 06:21:08 PDT 2022 from 10.4.6.1 on pts/0
Linux si2srv02 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:37 UTC 2011 i68
6 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Loading es
si2@si2srv02:~$ cat /etc/hosts
10.4.6.1 si2srv01
10.4.6.2 si2srv02
10.4.6.3 si2srv03
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
si2@si2srv02:~$ 

To grab input, press Ctrl+G

si2srv - VMware Workstation 16 Player (Non-commercial use only)
File Virtual Machine Help
Direccion nat: 172.16.188.130
Direccion bridge: 10.4.6.3
Inicie sesion con usuario si2
Ubuntu 10.04.3 LTS si2srv03 tty1
si2srv03 login: si2
Password:
Last login: Thu Apr  7 02:10:05 PDT 2022 on tty1
Linux si2srv03 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:37 UTC 2011 i68
6 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Loading es
si2@si2srv03:~$ cat /etc/hosts
10.4.6.1 si2srv01
10.4.6.2 si2srv02
10.4.6.3 si2srv03
127.0.0.1 localhost

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
si2@si2srv03:~$ 

To grab input, press Ctrl+G

```

Siguiendo el esquema, se deben crear las instancias que utilizaran los dos nodos en el host 10.4.6.2 (Nodo 1) y 10.4.6.3 (Nodo 2), y que albergan un servidor de aplicaciones virtual, para ello ejecutamos lo siguiente:

The image shows a single terminal window titled "jades Terminal". The user is executing the `asadmin` command to create instances on a cluster.

```

jades Terminal
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin --user admin create-instance --cluster SI2Cluster --node Node01 Instance01
Enter admin password for user "admin">
Command _create-instance-fs executed successfully.
Port Assignments for server instance Instance01:
OSGI_SHELL_TELNET_PORT=26666
JAVA_DEBUGGER_PORT=29009
JMS_PROVIDER_PORT=27676
HTTP_LISTENER_PORT=28080
IIOP_SSL_LISTENER_PORT=23820
ASADMIN_LISTENER_PORT=24848
IIOP_SSL_MUTUALAUTH_PORT=23920
JMX_SYSTEM_CONNECTOR_PORT=28686
HTTP_SSL_LISTENER_PORT=28181
IIOP_LISTENER_PORT=23700
The instance, Instance01, was created on host 10.4.6.2
Command create-instance executed successfully.
si2@si2srv01:~$ asadmin --user admin create-instance --cluster SI2Cluster --node Node02 Instance02
Enter admin password for user "admin">
Command _create-instance-fs executed successfully.
Port Assignments for server instance Instance02:
OSGI_SHELL_TELNET_PORT=26666
JAVA_DEBUGGER_PORT=29009
JMS_PROVIDER_PORT=27676
HTTP_LISTENER_PORT=28080
IIOP_SSL_LISTENER_PORT=23820
ASADMIN_LISTENER_PORT=24848
IIOP_SSL_MUTUALAUTH_PORT=23920
JMX_SYSTEM_CONNECTOR_PORT=28686
HTTP_SSL_LISTENER_PORT=28181
IIOP_LISTENER_PORT=23700
The instance, Instance02, was created on host 10.4.6.3
Command create-instance executed successfully.
si2@si2srv01:~$ 

```

Finalmente, comprobamos que las instancias se han creado correctamente listándolas:

```
Jades Terminal ▾
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin --user admin list-instances -l
Enter admin password for user "admin">
Name      Host     Port   Pid Cluster    State
Instance01 10.4.6.2 24848 -- SI2Cluster  not running
Instance02 10.4.6.3 24848 -- SI2Cluster  not running
Command list-instances executed successfully.
si2@si2srv01:~$
```

Como se puede observar, se han creado correctamente, y su estado es en parado, ya que no las hemos lanzado. Finalmente, vamos a iniciar el cluster, y tras unos segundos de ejecución:

```
Jades Terminal ▾
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin --user admin start-cluster SI2Cluster
Enter admin password for user "admin">
Command start-cluster executed successfully.
si2@si2srv01:~$
```

Ahora accedemos al servidor de aplicaciones para ver el estado del cluster. Se puede observar que se encuentra operativo:

The screenshot shows the GlassFish Server Open Source Edition web interface. On the left, there's a sidebar with navigation links like Home, About..., User: admin, Domain: domain1, and Server: 10.4.6.1. The main content area is titled 'Clusters'. It has a sub-header: 'Create and manage GlassFish Server clusters. A cluster is a named collection of GlassFish Server instances that provides high availability through scalability, load balancing, and failure protection.' Below this is a table titled 'Clusters (1)'. The table has columns: Select, Name, Configuration, and Instances. There is one entry: Name is 'SI2Cluster', Configuration is 'SI2Cluster-config', and Instances are 'Instance01' and 'Instance02', both marked as 'Running'.

Además, ejecutamos “ps aefl | grep java” para corroborar que efectivamente existen procesos que estén corriendo dichas instancias, para ello nos conectaremos mediante ssh a ambas instancias (máquinas virtuales) para ejecutar el comando:

```
Jades Terminal ▾
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv02:~$ ps aefl | grep java
0 1000 2385 2371 20 0 2220 752 - R+ pts/0 0:00 \_ ps aefl TERM=xterm-256color SHELL=/bin/bash XDG_SESSION_COOKIE=edc12e8ab2029e6cc
d027fb94f15fe2f-1649329043.669978-1769487984 SSH_CLIENT=10.4.6.1 49987 22 SSH_TTY=/dev/pts/0 ANT_HOME=/opt/glassfish4/ant USER=si2 MAIL=/var/mail/si2 PATH
=/usr/local/sbin:/usr/local/bin:/usr/bin:/sbin:/bin:/usr/games:/opt/glassfish4/glassfish/bin:/opt/glassfish4/glassfish:/opt/glassfish4/ant
/bin:/opt/SI2/jakarta-jmeter JVM_ARGS=-Duser.language=en PWD=/home/si2 JAVA_HOME=/usr/lib/jvm/java-7-oracle/ J2EE_HOME=/opt/glassfish4/glassfish LANG=C SH
LVL=1 HOME=/home/si2 LOGNAME=si2 SSH_CONNECTION=10.4.6.1 49987 10.4.6.2 22 _/bin/ps
0 1000 2386 2371 20 0 1860 572 pipe_w S+ pts/0 0:00 \_ grep java TERM=xterm-256color SHELL=/bin/bash XDG_SESSION_COOKIE=edc12e8ab2029e6
cc0d27fb94f15fe2f-1649329043.669978-1769487984 SSH_CLIENT=10.4.6.1 49987 22 SSH_TTY=/dev/pts/0 ANT_HOME=/opt/glassfish4/ant USER=si2 MAIL=/var/mail/si2 PATH
=/usr/local/sbin:/usr/local/bin:/usr/bin:/sbin:/bin:/usr/games:/opt/glassfish4/glassfish/bin:/opt/glassfish4/glassfish:/opt/glassfish4/ant
/bin:/opt/SI2/jakarta-jmeter JVM_ARGS=-Duser.language=en PWD=/home/si2 JAVA_HOME=/usr/lib/jvm/java-7-oracle/ J2EE_HOME=/opt/glassfish4/glassfish LANG=C SH
LVL=1 HOME=/home/si2 LOGNAME=si2 SSH_CONNECTION=10.4.6.1 49987 10.4.6.2 22 _/bin/grep
si2@si2srv02:~$
```

```
Jades Terminal ▾
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv03:~$ ps aefl | grep java
0 1000 2030 2044 20 0 2220 756 - R+ pts/0 0:00 \_ ps aefl TERM=xterm-256color SHELL=/bin/bash XDG_SESSION_COOKIE=edc12e8ab2029e6cc
d027fb94f15fe2f-1649329102.478545-937889409 SSH_CLIENT=10.4.6.1 38403 22 SSH_TTY=/dev/pts/0 ANT_HOME=/opt/glassfish4/ant USER=si2 MAIL=/var/mail/si2 PATH
=/usr/local/sbin:/usr/local/bin:/usr/bin:/sbin:/bin:/usr/games:/opt/glassfish4/glassfish/bin:/opt/glassfish4/glassfish:/opt/glassfish4/ant
/bin:/opt/SI2/jakarta-jmeter JVM_ARGS=-Duser.language=en PWD=/home/si2 JAVA_HOME=/usr/lib/jvm/java-7-oracle/ J2EE_HOME=/opt/glassfish4/glassfish LANG=C SH
LVL=1 HOME=/home/si2 LOGNAME=si2 SSH_CONNECTION=10.4.6.1 38403 10.4.6.3 22 _/bin/ps
0 1000 2031 2044 20 0 1860 568 pipe_w S+ pts/0 0:00 \_ grep java TERM=xterm-256color SHELL=/bin/bash XDG_SESSION_COOKIE=edc12e8ab2029e6
cc0d27fb94f15fe2f-1649329102.478545-937889409 SSH_CLIENT=10.4.6.1 38403 22 SSH_TTY=/dev/pts/0 ANT_HOME=/opt/glassfish4/ant USER=si2 MAIL=/var/mail/si2 PATH
=/usr/local/sbin:/usr/local/bin:/usr/bin:/sbin:/bin:/usr/games:/opt/glassfish4/glassfish/bin:/opt/glassfish4/glassfish:/opt/glassfish4/ant
/bin:/opt/SI2/jakarta-jmeter JVM_ARGS=-Duser.language=en PWD=/home/si2 JAVA_HOME=/usr/lib/jvm/java-7-oracle/ J2EE_HOME=/opt/glassfish4/glassfish LANG=C SH
LVL=1 HOME=/home/si2 LOGNAME=si2 SSH_CONNECTION=10.4.6.1 38403 10.4.6.3 22 _/bin/grep
si2@si2srv03:~$
```

Por último, vamos a modificar la configuración de las instancias del clúster como indica el guión:

User: admin | Domain: domain1 | Server: 10.4.6.1

GlassFish™ Server Open Source Edition

Total # of available updates : 1

Nodes Applications Lifecycle Modules Monitoring Data Resources Concurrent Resources Connectors JDBC JMS Resources JNDI JavaMail Sessions Resource Adapter Configs Configurations SI2Cluster-config

General Path Settings JVM Options Profiler

JVM Options

Manage JVM options for the server. Values containing one or more spaces must be enclosed in double quotes ("value string")

Configuration Name: SI2Cluster-config

Select	Value
<input type="checkbox"/>	-Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache.felix.fileinstall,org.apache.felix.scr,org.apache.felix.eventadmin,org.apache.felix.dependencymanager,org.apache.felix.metatype,org.apache.felix.bundlerefresh,org.apache.felix.cm,org.apache.felix.scr osgi.bundles.startLevel=1
<input type="checkbox"/>	-Djavax.net.ssl.keyStore=\${com.sun.aas.instanceRoot}/config/keystore.jks
<input checked="" type="checkbox"/>	-server
<input type="checkbox"/>	-DANTLR_USE_DIRECT_CLASS_LOADING=true
<input type="checkbox"/>	-Dfelix.fileinstall.bundles.startTransient=true
<input type="checkbox"/>	-Djavax.net.ssl.trustStore=\${com.sun.aas.instanceRoot}/config/cacerts.jks

User: admin | Domain: domain1 | Server: 10.4.6.1

GlassFish™ Server Open Source Edition

Total # of available updates : 1

Nodes Applications Lifecycle Modules Monitoring Data Resources Concurrent Resources Connectors JDBC JMS Resources JNDI JavaMail Sessions Resource Adapter Configs Configurations SI2Cluster-config Admin Service Availability Service Connector Service

General Path Settings JVM Options Profiler

JVM Options

Manage JVM options for the server. Values containing one or more spaces must be enclosed in double quotes ("value string")

Configuration Name: SI2Cluster-config

Select	Value
<input type="checkbox"/>	-XX:+UnlockDiagnosticVMOptions
<input type="checkbox"/>	-Djava.security.auth.login.config=\${com.sun.aas.instanceRoot}/config/login.conf
<input type="checkbox"/>	-Dfelix.fileinstall.disableConfigSave=false
<input type="checkbox"/>	-Djava.awt.headless=true
<input type="checkbox"/>	-Djdbc.drivers=org.apache.derby.jdbc.ClientDriver
<input type="checkbox"/>	-Didk.corba.allowOutputStreamSubclass=true
<input type="checkbox"/>	-Dosgi.shell.telnet.port=\${OSGI_SHELL_TELNET_PORT}
<input type="checkbox"/>	-Dosgi.shell.telnet.maxconn=1
<input type="checkbox"/>	-Xms128m
<input type="checkbox"/>	-Xmx128m
<input type="checkbox"/>	-Djava.ext.dirs=\${com.sun.aas.javaRoot}/lib/ext\${path.separator}\${com.sun.aas.javaRoot}/jre/lib/ext\${path.separator}
<input type="checkbox"/>	-Djava.security.policy=\${com.sun.aas.instanceRoot}/config/server.policy
<input type="checkbox"/>	-Dgosh.args=-noshutdown -c noop=true
<input type="checkbox"/>	-Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterprise.config.serverbeans.AppServerEnvFactory
<input checked="" type="checkbox"/>	-XX:MaxPermSize=96m
<input type="checkbox"/>	-Djava.endorsed.dirs=\${com.sun.aas.installRoot}/modules/endorsed\${path.separator}\${com.sun.aas.installRoot}/jre/lib/endorsed\${path.separator}
<input type="checkbox"/>	-Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as
<input type="checkbox"/>	-Dfelix.fileinstall.poll=5000
<input type="checkbox"/>	-Dfelix.fileinstall.bundles.new.start=true

Finalmente guardamos los cambios, y el cluster queda configurado correctamente.

Ejercicio número 3:

Pruebe a realizar un pago individualmente en cada instancia. Para ello, identifique los puertos en los que están siendo ejecutados cada una de las dos instancias (IPs 10.X.Y.2 y 10.X.Y.3 respectivamente). Puede realizar esa comprobación directamente desde la consola de administración, opción Applications, acción Launch, observando los Web Application Links generados.

Realice un único pago en cada nodo. Verifique que el pago se haya anotado correctamente el nombre de la instancia y la dirección IP. Anote sus observaciones (puertos de cada instancia) y evidencias (captura de pantalla de la tabla de pagos).

Primero se estructura la práctica de la manera que nos comentan, y una vez realizado esto, procederemos a los cambios en el código:

1. Se crean los campo de IP e Instancia dentro de la clase PagoBean, ya que se comenta que estas propiedades son incorporadas al pago:

```

/* Las dos nuevas variables privadas de atributos de clase */

private String instancia;

private String ip;

/* Los metodos setter y getter de la clase */

public String getIP() {

    return ip;

}

public void setIP(String ip) {

    this.ip = ip;

}

public String getInstance() {

    return instancia;

}

public void setInstance(String instance) {

    this.instancia = instance;

}

```

2. Lo siguiente es modificar los servlets de ComienzaPago y ProcesaPago, ya que recordemos que había dos manera de comenzar un pago, con lo cual, en concreto su método de creaPago, se añaden el set de IP y de Instance de la siguiente manera:

```

private PagoBean creaPago(HttpServletRequest request) {

    PagoBean pago = new PagoBean();

    pago.setIdTransaccion(request.getParameter(PARAM_ID_TRANSACCION));

    pago.setIdComercio(request.getParameter(PARAM_ID_COMERCIO));

    pago.setInstance(System.getProperty("com.sun.aas.instanceName"));

    double impd = -1.0;

    try {

        pago.setIP(java.net.InetAddress.getLocalHost().getHostAddress());
    }
}

```

```

        impd = Double.parseDouble(request.getParameter(PARAM_IMPORTE));

    } catch (NumberFormatException e) {

        impd = -1.0;

    } catch (NullPointerException e) {

        impd = -1.0;

    } catch (UnknownHostException e) {

        System.out.println(e);

    }

    pago.setImporte(impd);

    pago.setRutaRetorno(request.getParameter(PARAM_RUTA_RETORNO));

    return pago;

}

```

Se ha capturado la excepción si no se obtiene el host

3. Finalmente, se modifica la clase de VisaDAO para que ahora se tengan en cuenta de nuevo los dos campos:

a. La consulta para insertar

```

private static final String INSERT_PAGOS_QRY =
        "insert into pago(" +
        "idTransaccion,importe,idComercio,numeroTarjeta,ip,instancia)" +
        " values (?,?,?,?,?,?)";

```

b. El método de consulta de añadir un pago:

```

String getQryInsertPago(PagoBean pago) {

    String qry = "insert into pago("

        + "idTransaccion,"

        + "importe,idComercio,ip,instancia)"

        + " values ("

        + "''' + pago.getIdTransaccion() + ''',"

```

```

        + pago.getImporte() + ", "
        + " " + pago.getIdComercio() + " , "
        + " " + pago.getTarjeta().getNumero() + " "
        + " " + pago.getIP() + " "
        + " " + pago.getInstance() + " "
        + ") ";

    return qry;
}

}

```

- c. Los nuevos campos en la preparedstatement de realizar un pago en “realizaPago”:

```

stmt = con.prepareStatement(insert);

stmt.setString(1, pago.getIdTransaccion());
stmt.setDouble(2, pago.getImporte());
stmt.setString(3, pago.getIdComercio());
stmt.setString(4, pago.getTarjeta().getNumero());
stmt.setString(5, pago.getIP());
stmt.setString(6, pago.getInstance());

```

4. Modificación del “build.properties”:

```

nombre=P3

...
as.host=10.4.6.1
...
as.target=SI2Cluster

```

5. Modificación del “postgresql.properties”:

```

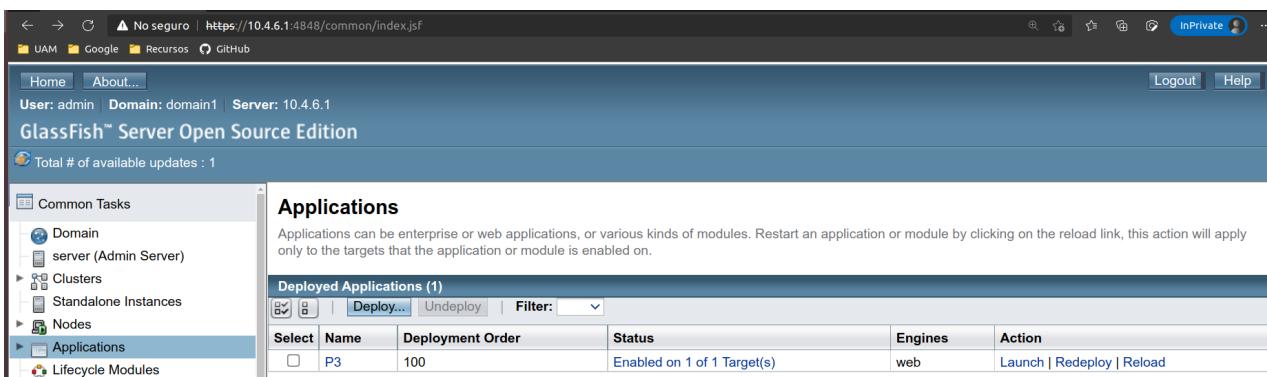
...
db.host=10.4.6.1
...
db.client.host=10.4.6.1

```

6. Modificación del fichero create.sql ya que debemos añadir los dos atributos más, ip e instancia:

```
ip          char(20) not null,  
  
instancia  char(20) not null,
```

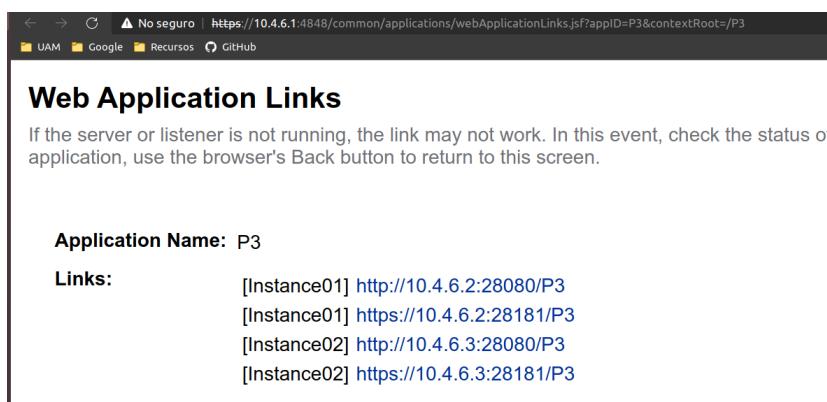
Una vez hemos realizado todos los cambios y ejecutamos “ant todo”, podemos observar en el servidor de aplicaciones que tenemos una nueva aplicación (P3), con un target que es el SI2Cluster:



Select	Name	Deployment Order	Status	Engines	Action
<input type="checkbox"/>	P3	100	Enabled on 1 of 1 Target(s)	web	Launch Redeploy Reload

Ahora vamos a realizar pagos en cada una de las instancias por separado para comprobar su funcionamiento:

1. Primero vamos a observar los puertos en los que están ejecutando cada una de las instancias:



Links:
[Instance01] http://10.4.6.2:28080/P3
[Instance01] https://10.4.6.2:28181/P3
[Instance02] http://10.4.6.3:28080/P3
[Instance02] https://10.4.6.3:28181/P3

Aquí se puede ver las opciones de cada instancia así como su opción “segura” (https), de tal manera que el host 10.4.6.2 (Instancia 1) se encuentra operando en el puerto 28080; y el host 10.4.6.3 (Instancia 2), se encuentra funcionando en el mismo puerto, pero el host obviamente es distinto.

2. Procedemos a realizar un pago como ya sabemos en el host 2:

Pago con tarjeta

Numero de visa:

Titular:

Fecha Emisión:

Fecha Caducidad:

CVV2:

Id Transacción: 1
Id Comercio: 1
Importe: 20.0

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

```
idTransaccion: 1
idComercio: 1
importe: 20.0
codRespuesta: 000
idAutorizacion: 1
```

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Como podemos observar, nos devuelve que el pago se ha realizado con éxito, accedemos adbeaver para comprobarlo:

idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	ip	instancia
1	1	000	20	1	1111 2222 3333 4444	01:40:17.030	10.4.6.2	Instance01

Y ahí observamos que los dos nuevos campos se han guardado correctamente.

3. Ahora repetimos el proceso con el host 10.4.6.3 y se completa con éxito:

Pago con tarjeta

Numero de visa:

Titular:

Fecha Emisión:

Fecha Caducidad:

CVV2:

Id Transacción: 2
Id Comercio: 2
Importe: 25.0

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

```
idTransaccion: 2
idComercio: 2
importe: 25.0
codRespuesta: 000
idAutorizacion: 2
```

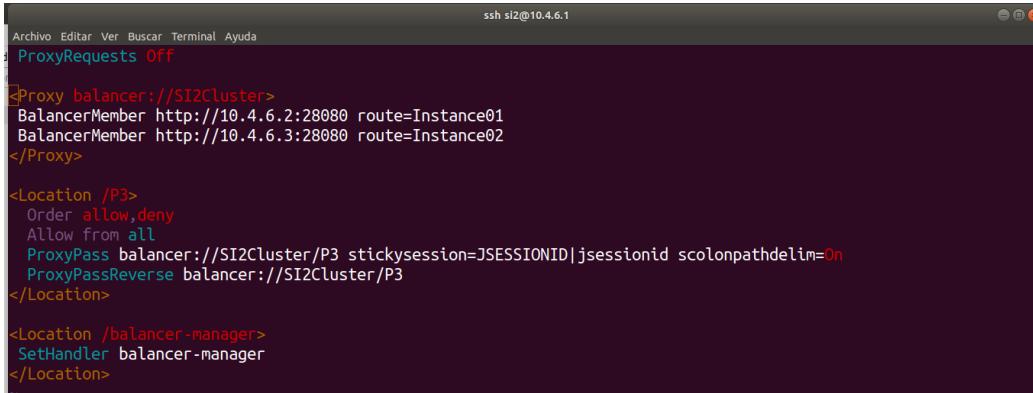
[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	ip	instancia
1	1	000	20	1	1111 2222 3333 4444	01:40:17.030	10.4.6.2	Instance01
2	2	000	25	2	4579 1165 7919 438	01:51:09.917	10.4.6.3	Instance02

Ejercicio número 4:

El fichero de configuración del balanceador de apache se ha editado con vim y ha quedado de la siguiente manera:



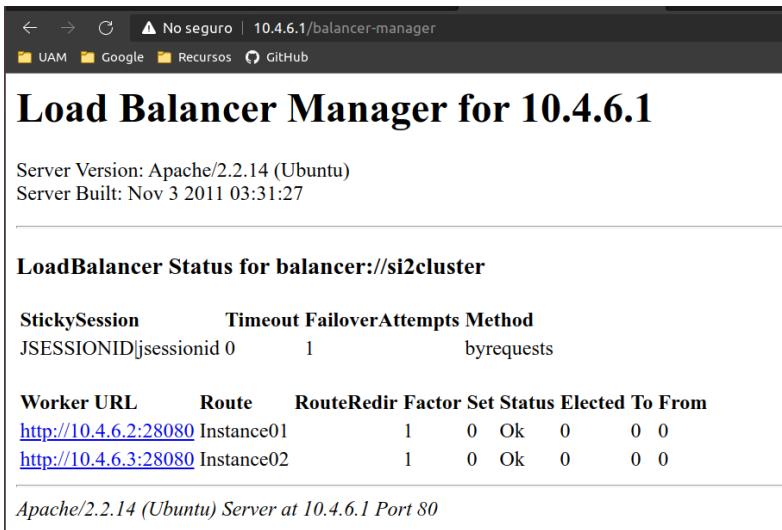
```
ssh si2@10.4.6.1
Archivo Editar Ver Buscar Terminal Ayuda
ProxyRequests Off

<Proxy balancer://SI2Cluster>
    BalancerMember http://10.4.6.2:28080 route=Instance01
    BalancerMember http://10.4.6.3:28080 route=Instance02
</Proxy>

<Location /P3>
    Order allow,deny
    Allow from all
    ProxyPass balancer://SI2Cluster/P3 sticky-session=JSESSIONID|jsessionid scolonpathdelim=On
    ProxyPassReverse balancer://SI2Cluster/P3
</Location>

<Location /balancer-manager>
    SetHandler balancer-manager
</Location>
```

Realizando los comandos de activar el balanceador en apache cargamos la url del balanceador y nos salen las dos instancias:



No seguro | 10.4.6.1/balancer-manager

UAM Google Recursos GitHub

Load Balancer Manager for 10.4.6.1

Server Version: Apache/2.2.14 (Ubuntu)
Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONID jsessionid 0	1		byrequests

Worker	URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
	http://10.4.6.2:28080	Instance01		1	0	Ok	0	0 0
	http://10.4.6.3:28080	Instance02		1	0	Ok	0	0 0

Apache/2.2.14 (Ubuntu) Server at 10.4.6.1 Port 80

Probar la influencia de jvmRoute en la afinidad de sesión.

1- Eliminar todas las cookies del navegador

**2- Sin la propiedad jvmRoute, acceder a la aplicación P3 a través de la URL del balanceador:
<http://10.X.Y.1/P3>**

3- Completar el pago con datos de tarjeta correctos.

4- Repetir los pagos hasta que uno falle debido a la falta de afinidad de sesión.

En nuestro caso, el fallo a llegado en el primer intento, introduciendo los campos de manera correcta:

No es seguro | 10.4.6.1/P3/procesapago

Gmail GitHub GitLab UAM YouTube Moodle UAM W3 IA Ma

Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II

5- Mostrar la cookie “JSESSIONID” correspondiente a la URL del balanceador donde se vea:

10.4.6.1/P3/procesapago

YouTube Moodle UAM W3 IA Maps Twitter

Incógnito Actualizar

jeta

Prácticas II

Aplicación

- Archivo de manifiesto
- Service Workers
- Almacenamiento

Almacenamiento

- Almacenamiento local
- Almacenamiento de sesión
- IndexedDB
- Web SQL
- Cookies
- Tokens de confianza

Filtrar

Nombre	Valor	Domain	Path	Expires...	Tamaño	HttpOnly	Secure	SameSite	SameP...	Priority
JSESSIONID	7fc4b58aacb5207f3105a072a7ee	10.4.6.1	/P3	Sesión	38	✓				Medium

6- Añadir la propiedad “jvmRoute” al cluster y rearrancar el cluster.

Accedemos la servidor de glassfish y modificamos la propiedad en las propiedades del cluster:

No seguro | https://10.4.6.1:4848/common/index.jsf

Logout Help

User: admin Role: domain1 Server: 10.4.6.1

GlassFish™ Server Open Source Edition

Total # of available updates : 1

Common Tasks

- Domain
- server (Admin Server)
- Clusters
- Standalone Instances
- Nodes
- Applications
- Lifecycle Modules
- Monitoring Data
- Resources

 - Concurrent Resources
 - Connectors
 - JDBC
 - JMS Resources
 - JNDI
 - JavaMail Sessions
 - Resource Adapter Configs

- Configurations

 - SI2Cluster-config
 - Admin Service
 - Availability Service
 - Connector Service
 - EJB Container

Configuration Name: SI2Cluster-config

Dynamic Reconfiguration:

Additional Properties (11)

Select	Instance Variable Name	Default Value	Instance Values
<input type="checkbox"/>	ASADMIN_LISTENER_PORT	24848	Instance Values
<input type="checkbox"/>	HTTP_LISTENER_PORT	28080	Instance Values
<input type="checkbox"/>	HTTP_SSL_LISTENER_PORT	28181	Instance Values
<input type="checkbox"/>	IOP_LISTENER_PORT	23700	Instance Values
<input type="checkbox"/>	IOP_SSL_LISTENER_PORT	23820	Instance Values
<input type="checkbox"/>	IOP_SSL_MUTUALAUTH_PORT	23920	Instance Values
<input type="checkbox"/>	JAVA_DEBUGGER_PORT	29009	Instance Values
<input type="checkbox"/>	JMS_PROVIDER_PORT	27676	Instance Values
<input type="checkbox"/>	JMX_SYSTEM_CONNECTOR_PORT	28686	Instance Values
<input type="checkbox"/>	OSGI_SHELL_TELNET_PORT	26666	Instance Values
<input type="checkbox"/>	jvmRoute	\$(com.sun.aas.instanceName)	Instance Values

Clusters and/or Instances using this configuration: Instance01 Instance02

Save Cancel

7- Eliminar todas las cookies del navegador.

8- Acceso a la aplicación P3 a través de la URL del balanceador:

<http://10.X.Y.1/P3>

9- Completar el pago con datos de tarjeta correctos. Se pueden repetir los pagos y no fallará.

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

```

idTransaccion: 3
idComercio: 3
importe: 12.0
codRespuesta: 000
idAutorizacion: 5

```

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Ahora el pago se realiza de manera correcta

10- Mostrar la cookie “JSESSIONID” correspondiente a la URL del balanceador donde se vea:

En cuanto a la cookie queda de la siguiente manera:

Nombre	Valor	Domain	Path	Expires /...	Tamaño	HttpOnly	Secure	SameSite	SameParty	Priority
JSESSIONID	806db81452d020894e1248fe49bd.instance01	10.4.6.1	/P3	Sesión	49	✓				Medium

Mostrar las pantallas y comentar: las diferencias en el contenido de las cookies respecto a jvmRoute, y cómo esta diferencia afecta a la afinidad y por qué.

La diferencia por la cual funciona, es que al añadir la propiedad \${com.sun.aas.instanceName} , hemos añadido al valor de la cookie (concatenado con un punto), la instancia en la que se ejecuta del cluster, por lo que ya tenemos control sobre la afinidad de sesión.

Esto es, que cuando intentemos hacer otro pago, si disponemos de la cookie de sesión, el cluster nos llevará de nuevo al mismo nodo del cluster.

De hecho a continuación, se prueba sin borrar la cookie para comprobar:

Nombre	Valor	Domain	Path	Expires /...	Tamaño	HttpOnly	Secure	SameSite	SameParty	Priority
JSESSIONID	819e2d214e951343645cec8b44c5.Instance01	10.4.6.1	/P3	Sesión	49	✓				Medium

Y como podemos ver, se mantiene la instancia, cumpliendo la afinidad de sesión.

¿Se podría, en general, usar el valor \${com.sun.aas.hostName} para la propiedad jvmRoute, en lugar de \${com.sun.aas.instanceName}?

Antes de comprobarlo, esto no va a ser posible simplemente porque nuestro balanceador no sabe de

nombres de hosts. Esto es que en la directiva ‘route’ para establecer el balanceador de carga, nombramos a las rutas de nuestra aplicación para cada instancia con su nombre de instancia, con lo cual, por mucho que ahora concatenamos con jvmRoute al nombre de host, luego no va a saber a donde “redireccionarlo”.

1. Añadimos la propiedad al cluster:

Select	Instance Variable Name	Default Value	Instance Values
<input type="checkbox"/>	ASADMIN_LISTENER_PORT	24848	Instance Values
<input type="checkbox"/>	HTTP_LISTENER_PORT	28080	Instance Values
<input type="checkbox"/>	HTTP_SSL_LISTENER_PORT	28181	Instance Values
<input type="checkbox"/>	IIOP_LISTENER_PORT	23700	Instance Values
<input type="checkbox"/>	IIOP_SSL_LISTENER_PORT	23820	Instance Values
<input type="checkbox"/>	IIOP_SSL_MUTUALAUTH_PORT	23920	Instance Values
<input type="checkbox"/>	JAVA_DEBUGGER_PORT	29009	Instance Values
<input type="checkbox"/>	JMS_PROVIDER_PORT	27676	Instance Values
<input type="checkbox"/>	JMX_SYSTEM_CONNECTOR_PORT	28686	Instance Values
<input type="checkbox"/>	OSGI_SHELL_TELNET_PORT	26666	Instance Values
<input type="checkbox"/>	jvmRoute	\$(com.sun.aas.hostName)	Instance Values

2. Una vez relanzamos el cluster por el cambio procedemos al pago:

Como podemos ver, esta vez ha cogido el host 3 para realizar el pago, pero el balanceador de carga al no conocer de nombres de host, no sabe a qué instancia del glassfish mandar la petición.

Ejercicio número 5:

Probar el balanceo de carga y la afinidad de sesión, realizando un pago directamente contra la dirección del cluster <http://10.X.Y.1/P3> desde distintos ordenadores. Comprobar que las peticiones se reparten entre ambos nodos del cluster, y que se mantiene la sesión iniciada por cada usuario sobre el mismo nodo.

Comentad la información mostrada en la página del Load Balancer Manager.

Como no se disponía de varios equipos para hacer este proceso, vamos a realizarlo sobre distintas sesiones (pestañas con cookies borradas) del navegador.

Si iniciamos dos pantallas del navegador distintas, podemos ver como la primera petición se va a la instancia 1:

No es seguro | 10.4.6.1/P3/comienzapago

The screenshot shows the Network tab of a browser developer tools interface. A cookie named JSESSIONID is listed, with the value db602de1673720adea3988f916a9.Instance01. The cookie is set for domain 10.4.6.1, path /P3, session, and has a size of 49 bytes. The cookie is marked as HttpOnly and Secure.

Nombre	Valor	Domain	Path	Expires ...	Tamaño	HttpOnly	Secure	SameSite	Sa...
JSESSIONID	db602de1673720adea3988f916a9.Instance01	10.4.6.1	/P3	Sesión	49	✓			

Pago con tarjeta

Numero de visa: 3703 9193 7960 479€

Titular: Kate Torres San Martín

Fecha Emisión: 10/10

Fecha Caducidad: 01/23

CVV2: 100

Id Transacción: 4
Id Comercio: 12
Importe: 44.0

Prácticas de Sistemas Informáticos II

Y la segunda va a la instancia 2:

No es seguro | 10.4.6.1/P3/comienzapago

The screenshot shows the Network tab of a browser developer tools interface. A cookie named JSESSIONID is listed, with the value db850ad16f57ec5b20a75a8796bd.Instance02. The cookie is set for domain 10.4.6.1, path /P3, session, and has a size of 49 bytes. The cookie is marked as HttpOnly and Secure.

Nombre	Valor	Domain	Path	Expires ...	Tamaño	HttpOnly	Secure	SameSite	Sa...
JSESSIONID	db850ad16f57ec5b20a75a8796bd.Instance02	10.4.6.1	/P3	Sesión	49	✓			

Pago con tarjeta

Numero de visa: 2039 6424 7433 0841

Titular: Luis Moreno Lopez

Fecha Emisión: 08/09

Fecha Caducidad: 05/23

CVV2: 411

Id Transacción: 5
Id Comercio: 12
Importe: 33.0

Prácticas de Sistemas Informáticos II

Y como comentamos en el ejercicio anterior, al disponer de esta cookie de sesión, las peticiones se mantendrán en la misma instancia.

A continuación se muestra que al finalizar el pago anterior, se ha vuelto a comenzar otro pago y las instancias se mantienen:

No es seguro | 10.4.6.1/P3/comienzapago

The screenshot shows the Network tab of a browser developer tools interface. A cookie named JSESSIONID is listed, with the value dc1cbdc1db836f7d83ee62f0177.Instance01. The cookie is set for domain 10.4.6.1, path /P3, session, and has a size of 49 bytes. The cookie is marked as HttpOnly and Secure.

Nombre	Valor	Domain	Path	Expires ...	Tamaño	HttpOnly	Secure	SameSite	Sa...
JSESSIONID	dc1cbdc1db836f7d83ee62f0177.Instance01	10.4.6.1	/P3	Sesión	49	✓			

Pago con tarjeta

Numero de visa: [redacted]

Titular: [redacted]

Fecha Emisión: [redacted]

Fecha Caducidad: [redacted]

CVV2: [redacted]

Id Transacción: 22
Id Comercio: 5
Importe: 44.0

Prácticas de Sistemas Informáticos II

Finalmente vamos a observar el balanceador del cluster, las peticiones son repartidas de manera equitativa entre ambos nodos del cluster, donde el nodo con route 'Instance 1' ha sido elegido 9 veces, y el nodo con route 'Instance 2', ha sido elegido 7 veces:

Ejercicio número 6:

Comprobación del proceso de fail-over. Parar la instancia del cluster que haya tenido menos elecciones hasta el momento. Para ello, identificamos el pid (identificador del proceso java) de la instancia usando las herramientas descritas en esta práctica o el mandato 'ps -afe | grep java'. Realizaremos un kill -9 pid en el nodo correspondiente. Vuelva a realizar peticiones y compruebe (accediendo a la página /balancer-manager y revisando el contenido de la base de datos) que el anterior nodo ha sido marcado como "erróneo" y que todas las peticiones se dirijan al nuevo servidor. Adjunte la secuencia de comandos y evidencias obtenidas en la memoria de la práctica.

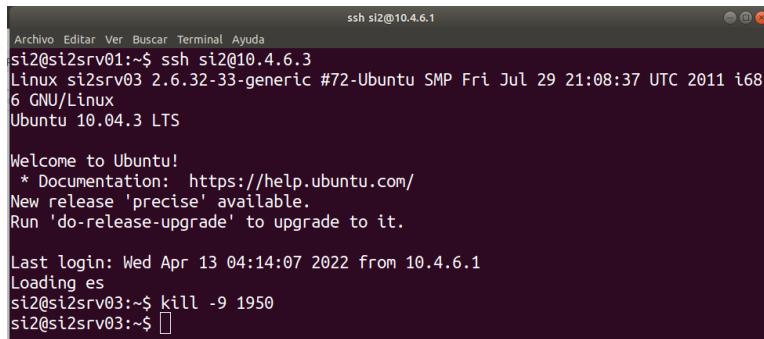
Para este apartado como en el anterior apartado se podía observar, la Instancia 2 tiene menos peticiones, así que es la que vamos a interrumpir.

Para ello, en primer lugar vamos a ver en qué PID se está ejecutando dicha instancia:

```
ssh si2@10.4.6.1
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin list-instances -l
Enter admin user name> admin
Enter admin password for user "admin">
Name      Host     Port   Pid   Cluster   State
Instance01 10.4.6.2 24848 1974  SI2Cluster running
Instance02 10.4.6.3 24848 1950  SI2Cluster running
Command list-instances executed successfully.
si2@si2srv01:~$
```

Como podemos observar, la instancia 2 se está ejecutando en el proceso 1950. Con lo cual vamos a ejecutar el comando **kill -9 1950** en su propio host:

1. Para ello iniciamos sesión a través de ssh al host 10.4.6.3, y una vez dentro lo ejecutamos

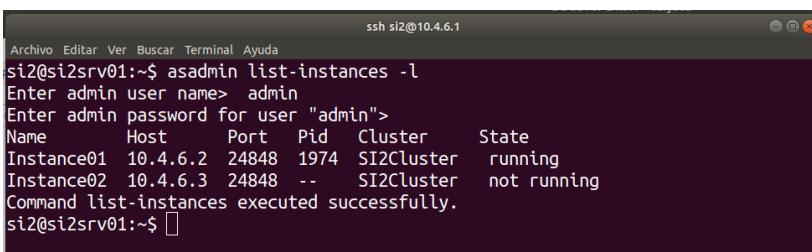


```
ssh si2@10.4.6.1
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ ssh si2@10.4.6.3
Linux si2srv03 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:37 UTC 2011 i68
6 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Apr 13 04:14:07 2022 from 10.4.6.1
Loading es
si2@si2srv03:~$ kill -9 1950
si2@si2srv03:~$ 
```

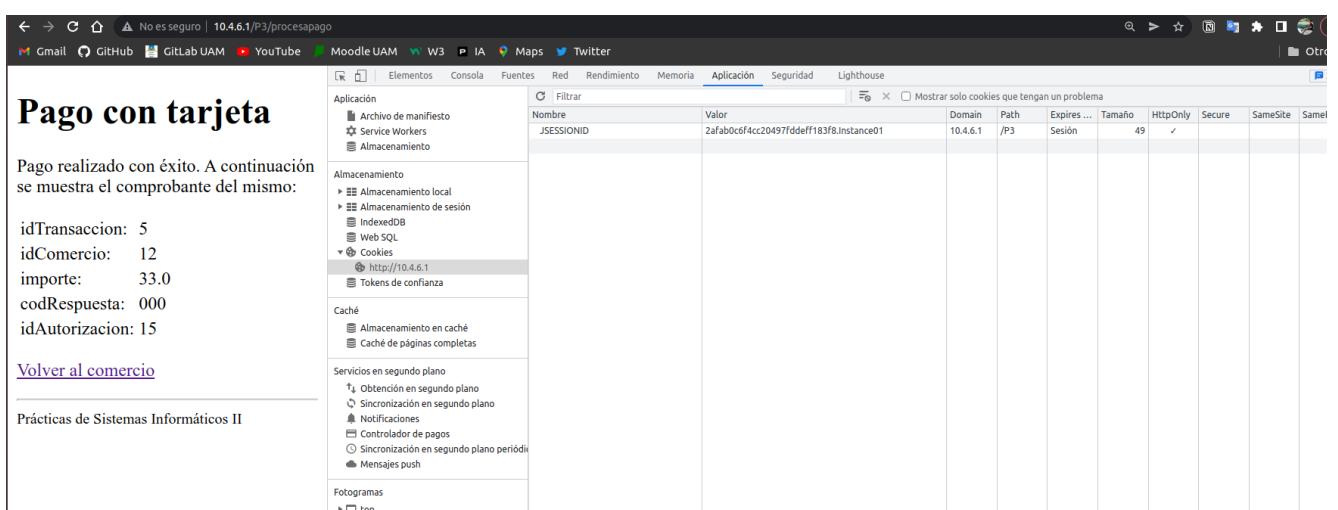
A continuación, vamos a comprobar de nuevo con el comando de listar las instancias si se encuentra parado el proceso:



```
ssh si2@10.4.6.1
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin list-instances -l
Enter admin user name> admin
Enter admin password for user "admin">
Name          Host      Port    Pid   Cluster     State
Instance01    10.4.6.2  24848  1974  SI2Cluster  running
Instance02    10.4.6.3  24848   --    SI2Cluster  not running
Command list-instances executed successfully.
si2@si2srv01:~$ 
```

Y como podemos observar, el proceso se encuentra parado. Para hacer una comprobación del mismo vamos a crear varias sesiones de pagos:

Nos ha pasado que teníamos un navegador que en ese momento, tenía una cookie con la Instancia 2 de un pago anterior, y al darle a “volver a realizar un pago” se ha cambiado de inmediato a la Instancia 1, con lo cual podemos estar seguros que todos los pagos se realizarán con la Instancia 1.



Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

idTransaccion: 5
idComercio: 12
importe: 33.0
codRespuesta: 000
idAutorizacion: 15

Volver al comercio

Prácticas de Sistemas Informáticos II

The screenshot shows a browser developer tools Network tab. It lists several cookies under the 'Aplicación' section. One cookie is highlighted: 'JSESSIONID' with the value '2afab0c6f4cc20497fddef1183f8.Instance01'. Other visible cookies include '_ga', '_gat', '_gid', 'PHPSESSID', 'session_id', and 'ci_sessions'. The 'Almacenamiento' section shows local storage and indexedDB entries. The 'Caché' section shows a few items. The 'Servicios en segundo plano' section lists background services like 'Obtención en segundo plano', 'Sincronización en segundo plano', 'Notificaciones', 'Controlador de pagos', 'Sincronización en segundo plano periódico', and 'Mensajes push'. The 'Fotogramas' section shows a 'top' frame.

A continuación, eliminamos la cookie de sesión y volvemos a lanzar otro pago:

Y nos lanza de nuevo la instancia 1.

Ejercicio número 7:

Comprobación del proceso de fail-back. Inicie manualmente la instancia detenida en el comando anterior. Verificar la activación de la instancia en el gestor del balanceador. Incluir todas las evidencias en la memoria de prácticas y comentar qué sucede con los nuevos pagos. Consulte los apéndices para información detallada de comandos de gestión individual de las instancias.

Comentar qué sucede con los nuevos pagos.

Ahora mismo nuestro balanceador de carga se encuentra en el siguiente estado:

Lo que haremos a continuación, será relanzar la instancia que se encuentra parada (Instancia 2), mediante el comando **start-instance**:

```
ssh si2@10.4.6.1
Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin start-instance Instance02
Enter admin user name> admin
Enter admin password for user "admin">
Waiting for Instance02 to start .....
Successfully started the instance: Instance02
instance Location: /opt/glassfish4/Node02/Instance02
Log File: /opt/glassfish4/Node02/Instance02/Logs/server.log
Admin Port: 24848
Command start-local-instance executed successfully.
The instance, Instance02, was started on host 10.4.6.3
Command start-instance executed successfully.
si2@si2srv01:~$
```

Ahora haremos comprobaciones, primero vamos a listar las instancias para observar que se ha asignado un nuevo proceso a la misma:

```
ssh s12@10.4.6.1
Archivo Editor Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin list-instances -l
Enter admin user name> admin
Enter admin password for user "admin">
Name      Host     Port   Pid Cluster State
Instance01 10.4.6.2 24848 1974 SI2Cluster running
Instance02 10.4.6.3 24848 2237 SI2Cluster running
Command list-instances executed successfully.
si2@si2srv01:~$
```

Se le asigna el proceso 2237 y finalmente vamos a comprobarlo en el servidor glassfish:

Select	Name	Configuration	Instances				
<input type="checkbox"/>	SI2Cluster	SI2Cluster-config	<table border="1"> <tr> <td>Instance01</td> <td>Running</td> </tr> <tr> <td>Instance02</td> <td>Running</td> </tr> </table>	Instance01	Running	Instance02	Running
Instance01	Running						
Instance02	Running						

A continuación, hemos realizado 4 pagos en 4 sesiones distintas con el siguiente orden:

Nombre	Valor	Domain	Path	Expires...	Tamaño	HttpOnly	Secure	SameSite
JSESSIONID	2cc21b911c50515fd136c2c32Ba.Instance01	10.4.6.1	/P3	Sesión	49	✓		

Nombre	Valor	Domain	Path	Expires...	Tamaño	HttpOnly	Secure	SameSite
JSESSIONID	2cce4e09687af0df241ffe07409.Instance02	10.4.6.1	/P3	Sesión	49	✓		

Sistema de Pago con tarjeta | Sistema de Pago con tarjeta | Sistema de Pago con tarjeta | +

No es seguro | 10.4.6.1/P3/procesapago

Gmail GitHub GitLab UAM YouTube Moodle UAM W3 IA Maps Twitter

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

```

idTransaccion: 67
idComercio: 434
importe: 33.0
codRespuesta: 000
idAutorizacion: 18

```

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Aplicación

Nombre	Valor	Domain	Path	Expires...	Tamaño	HttpOnly	Secure	SameSite
JSESSIONID	2cd91a219e6b0c4aa186ea174541.instance01	10.4.6.1	/P3	Sesión	49	✓		

Almacenamiento

- Almacenamiento local
- Almacenamiento de sesión
 - IndexedDB
 - Web SQL
 - Cookies
 - http://10.4.6.1
 - Tokens de confianza

Caché

- Almacenamiento en caché
- Caché de páginas completas

Servicios en segundo plano

- Obtención en segundo plano
- Sincronización en segundo plano
- Notificaciones
- Controlador de pagos
- Sincronización en segundo plano periódica
- Mensajes push

Fotogramas

- top

Sistema de Pago con tarjeta | +

No es seguro | 10.4.6.1/P3/procesapago

Gmail GitHub GitLab UAM YouTube Moodle UAM W3 IA Maps Twitter

Pago con tarjeta

Pago realizado con éxito. A continuación se muestra el comprobante del mismo:

```

idTransaccion: 66
idComercio: 77
importe: 345.0
codRespuesta: 000
idAutorizacion: 19

```

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Aplicación

Nombre	Valor	Domain	Path	Expires...	Tamaño	HttpOnly	Secure	SameSite
JSESSIONID	2ce47c7861099011b44b91c9ec0f.instance02	10.4.6.1	/P3	Sesión	49	✓		

Almacenamiento

- Almacenamiento local
- Almacenamiento de sesión
 - IndexedDB
 - Web SQL
 - Cookies
 - http://10.4.6.1
 - Tokens de confianza

Caché

- Almacenamiento en caché
- Caché de páginas completas

Servicios en segundo plano

- Obtención en segundo plano
- Sincronización en segundo plano
- Notificaciones
- Controlador de pagos
- Sincronización en segundo plano periódica
- Mensajes push

Fotogramas

- top

Como se puede observar, los pagos se van intercalando entre ambas instancias, por lo que podemos intuir que está utilizando algún tipo de algoritmo de estilo como round-robin.

Finalmente, mostramos el estado final del balanceador de carga a método de comprobacion:

Load Balancer Manager for 10.4.6.1

Server Version: Apache/2.2.14 (Ubuntu)

Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONID	[jsessionid]	0	1
			byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.4.6.2:28080	Instance01		1	0	Ok	18	14K 18K
http://10.4.6.3:28080	Instance02		1	0	Ok	14	9.2K 12K

Apache/2.2.14 (Ubuntu) Server at 10.4.6.1 Port 80

Ejercicio número 8:

Fallo en el transcurso de una sesión.

- Desde un navegador, comenzar una petición de pago introduciendo los valores del mismo en la pantalla inicial y realizando la llamada al servlet ComienzaPago.

Vamos a la pantalla de iniciar el pago:

Nombre	Valor	Domain	Path	Expires...	Tamaño	HttpOnly	Secure
JSESSIONID	2dcdebb5f8f9102e8fb2077ab4.Instance01	10.4.6.1	/P3	Sesión	49		

En este momento tenemos un pago que está en proceso en la instancia 1.

- Al presentarse la pantalla de "Pago con tarjeta", leer la instancia del servidor que ha procesado la petición y detenerla. Se puede encontrar la instancia que ha procesado la petición revisando la cookie de sesión (tiene la instancia como sufijo), el balancer-manager o el server.log de cada instancia.

Paramos la instancia de la misma manera que en el ejercicio 7, primero listamos las instancias para ver que la 1 se está ejecutando en el proceso 1974, posteriormente iniciamos el ssh con el host 2 que es el encargado de correr dicha instancia, y hacemos el kill -9.

```

Archivo Editar Ver Buscar Terminal Ayuda
si2@si2srv01:~$ asadmin list-instances -l
Enter admin user name> admin
Enter admin password for user "admin">
Name          Host      Port    Pid   Cluster     State
Instance01    10.4.6.2  24848  1974  SI2Cluster  running
Instance02    10.4.6.3  24848  2237  SI2Cluster  running
Command list-instances executed successfully.
si2@si2srv01:~$ ssh si2@10.4.6.2
Linux si2srv02 2.6.32-33-generic #72-Ubuntu SMP Fri Jul 29 21:08:37 UTC 2011 i68
6 GNU/Linux
Ubuntu 10.04.3 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
New release 'precise' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Wed Apr 13 03:38:03 2022
Loading es
si2@si2srv02:~$ kill -9 1974
si2@si2srv02:~$ 

```

Finalmente comprobamos que la instancia efectivamente ha parado:

Select	Name	Configuration	Instances				
<input type="checkbox"/>	SI2Cluster	SI2Cluster-config	<table border="1"> <tr> <td>Instance01</td> <td>Stopped</td> </tr> <tr> <td>Instance02</td> <td>Running</td> </tr> </table>	Instance01	Stopped	Instance02	Running
Instance01	Stopped						
Instance02	Running						

- Completar los datos de la tarjeta de modo que el pago fuera válido, y enviar la petición.

Los datos de pago quedan de la siguiente manera, donde de momento al no haber actualizado la página, la instancia se mantiene:

Nombre	Valor	Domain	Path	Expires ...	Tamaño	HttpOnly	Secure
JSESSIONID	2dc6eb5f8f9102e8fb2077ab84.Instance01	10.4.6.1	/P3	Sesión	49	✓	

- Observar la instancia del cluster que procesa el pago, y razonar las causas por las que se rechaza

la petición.

Se finaliza el pago y lo que pasa es que finaliza el pago como Incorrecto, y justo después se crea una nueva cookie de sesión, pero de la instancia 2 que es la que está funcionando:

Nombre	Valor	Domain	Path	Expires ...	Tamano	HttpOnly	Secure
JSESSIONID	2e59ab214570f72f292ef900ec38.instance02	10.4.6.1	/P3	Sesión	49	✓	

Este proceso tiene sentido, el pago no se puede finalizar de manera correcta porque la información de la sesión ha sido perdida en medio del proceso de compra al pararse el host donde se estaba ejecutando.

Ejercicio número 9:

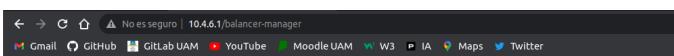
Modificar el script de pruebas JMeter desarrollado durante la P2. (P2.jmx) Habilitar un ciclo de 1000 pruebas en un solo hilo contra la IP del cluster y nueva URL de la aplicación: <http://10.X.Y.1/P3>

Eliminar posibles pagos previos al ciclo de pruebas. Verificar el porcentaje de pagos realizados por cada instancia, así como (posibles) pagos correctos e incorrectos. ¿Qué algoritmo de reparto parece haber seguido el balanceador? Comente todas sus conclusiones en la memoria de prácticas

Se han limpiado todos los pagos de la base de datos, y arrancado el cluster para activar la instancia que fue parada en anterior apartado.

Además, se va a hacer un restart de apache2, para que el balancer-manager vuelva a su estado inicial con 0 transacciones realizadas, con el siguiente comando:

```
sudo service apache2 restart
```



Load Balancer Manager for 10.4.6.1

Server Version: Apache/2.2.14 (Ubuntu)
Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

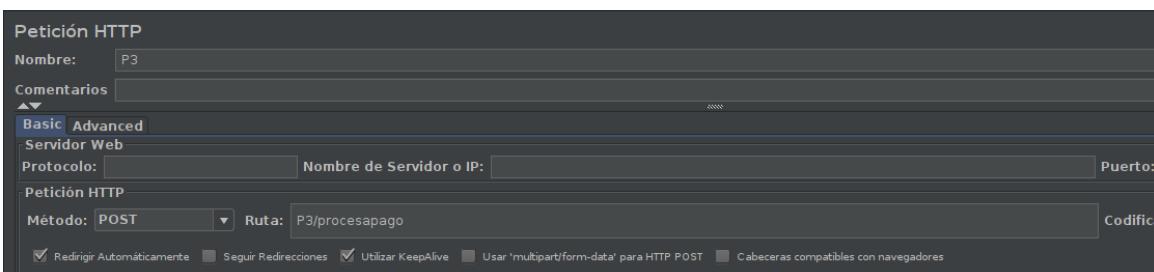
StickySession	Timeout	FailoverAttempts	Method					
JSESSIONID[jsessionid 0]	1		byrequests					
Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	To	From
http://10.4.6.2:28080	Instance01		1	0	Ok	0	0	0
http://10.4.6.3:28080	Instance02		1	0	Ok	0	0	0

Apache/2.2.14 (Ubuntu) Server at 10.4.6.1 Port 80

A continuación, se ha reutilizado el plan de pruebas de la práctica 2 anterior, quedándonos tan solo con el de P1-base, y se ha modificado el host y el puerto poniendo al host 1 y el puerto por defecto 80:



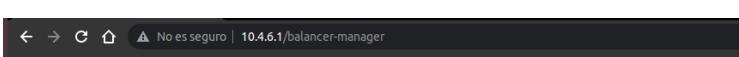
Finalmente, se ha cambiado la pestaña de la petición http para configurar en la nueva ruta del proyecto P3:



Finalmente, se ejecuta el plan de pruebas y observamos los resultado, con un 100% de acierto:

Etiqueta	# Muestras	Media	Mediana	90% Line	95% Line	99% Line	Mín	Máx	% Error	Rendimiento	Kb/sec	Sent KB/sec
P3	1000	11	6	11	13	24	4	1936	0,00%	85,3/sec	184,14	0,00
Total	1000	11	6	11	13	24	4	1936	0,00%	85,3/sec	184,14	0,00

Ahora iremos a la página del balanceador para comprobar cómo se han distribuido los pagos en el cluster:



Load Balancer Manager for 10.4.6.1

Server Version: Apache/2.2.14 (Ubuntu)
Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method					
JSESSIONID[jsessionid 0]	1		byrequests					
Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	To	From
http://10.4.6.2:28080	Instance01		1	0	Ok	500	287K	907K
http://10.4.6.3:28080	Instance02		1	0	Ok	500	287K	907K

Apache/2.2.14 (Ubuntu) Server at 10.4.6.1 Port 80

Como se puede observar, ha habido un reparto equitativo de los pagos, lo que corrobora la teoría que teníamos en el ejercicio 7 de qué se sigue un algoritmo del estilo round-robin.