

Memoria Practica 3 SI2

Ejercicio 1

En primer lugar creamos y configuramos correctamente las 3 máquinas virtuales necesarias para la práctica, y creamos en la primera las claves pública y privada.

Tras esto, enviamos la clave pública a las otras dos máquinas virtuales usando el comando scp, y las insertamos en el fichero `~/.ssh/authorized_keys2` como se indica en el enunciado.

Entonces, procedemos a comprobar que podemos conectarnos a ambas máquinas virtuales sin necesidad de introducir la contraseña.

Probamos en primer lugar que se conecta correctamente a la segunda máquina virtual ejecutando el comando `ssh -v si2@10.1.2.2`, de este modo obtenemos la siguiente salida:

```
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: Trying private key: /home/si2/.ssh/id_rsa
debug1: Offering public key: /home/si2/.ssh/id_dsa
debug1: Server accepts key: pkalg ssh-dss blen 433
debug1: read PEM private key done: type DSA
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: Mon Apr 20 11:03:44 2020 from 10.1.2.1
Loading es
si2@si2srv02:~$
```

Del mismo modo, comprobamos que se conecta correctamente a la máquina virtual 3 con el comando `ssh -v si2@10.1.2.3` y obtenemos:

```

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: Trying private key: /home/si2/.ssh/id_rsa
debug1: Offering public key: /home/si2/.ssh/id_dsa
debug1: Server accepts key: pkalg ssh-dss blen 433
debug1: read PEM private key done: type DSA
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: Mon Apr 20 10:52:18 2020
Loading es
si2@si2srv03:~$ █

```

Debido a las limitaciones en el tamaño de la pantalla de la máquina virtual, nos es imposible mostrar en las capturas que el comando ejecutado es el mencionado anteriormente. Sin embargo, las capturas nos sirven para mostrar que, en ambos casos, la conexión ha sido aceptada con una clave pública, y que las máquinas virtuales se han conectado correctamente por ssh.

Por tanto, tras esto, realizamos copias de las tres máquinas virtuales y las guardamos para los siguientes ejercicios de la práctica.

Ejercicio 2

Para comenzar este ejercicio iniciamos las 3 máquinas virtuales guardadas, nos conectamos mediante SSH a la primera y arrancamos Glassfish.

Tras esto procedemos a la creación del cluster configurando, en primer lugar, los nodos SSH de este. Para ello, siguiendo los pasos del enunciado iniciamos sesión SSH en la primer máquina y creamos el `Node01` indicando que este se va a ejecutar en la máquina `10.1.2.2` con el usuario `si2`. Realizamos el mismo proceso para el `Node02` que se ejecutará en la máquina `10.1.2.3`. Una vez ambos nodos están configurados, comprobamos que el proceso se ha realizado de forma correcta listándolos y realizando un ping a cada uno como nos indica el enunciado.

Pasamos ahora a crear el cluster con el comando:

```
1 | asadmin create-cluster SI2Cluster
```

Listamos el cluster para comprobar que se ha creado correctamente y comprobamos en el fichero `etc/hosts` de cada uno de los nodos que se conocen, es decir, que dichos ficheros tienen el nombre y la dirección IP de los nodos del cluster. Finalmente, como nos indica el enunciado, creamos 2 instancias, una a cada uno de los 2 nodos que habíamos creado, tras esto iniciamos el cluster.

Entramos en la consola de administración de glassfish (de la IP 10.1.2.1) para modificar la configuración del cluster estableciendo en JVM Sttings -> JVM Options las opciones de configuración que nos indica el enunciado, tras lo cual reiniciamos el cluster.

El cluster estará ahora creado y configurado, y, por tanto, procedemos a realizar las comprobaciones que se nos piden.

En primer lugar listamos las instancias del cluster para comprobar que se han iniciado correctamente:

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile list-instances -l
Name      Host      Port  Pid   Cluster  State
Instance01 10.1.2.2  24848 1809  SI2Cluster  running
Instance02 10.1.2.3  24848 1801  SI2Cluster  running
Command list-instances executed successfully.
si2@si2srv01:~$
```

Podemos comprobar que ambas instancias del servidor están iniciadas.

Observamos ahora los PIDs relacionados con java de ambas máquinas virtuales, y vemos que, como esperábamos, solo corre el Node almacenado en cada una de ellas (los otros dos números de proceso son los relacionados con ps y con grep java).

```
si2@si2srv03:~$ ps -aefl | grep java
0 S si2      1801      1 6 80 0 - 139701 futex 11:45 ?      00:00:24 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/glassfish4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRatio=2 -XX:MaxPermSize=96m -Xmx128m -Xms128m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.net.ssl.trustStore=/opt/glassfish4/Node02/Instance02/config/cacerts.jks -Djdk.corba.allowOutputStreamSubclass=true -Dfelix.fileinstall.dir=/opt/glassfish4/glassfish/modules/autostart/ -Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache.felix.gogo.shell,org.apache.felix.gogo.command,org.apache.felix.fileinstall -Dcom.sun.aas.installRoot=/opt/glassfish4/glassfish -Dfelix.fileinstall.poll=5000 -Djava.security.policy=/opt/glassfish4/Node02/Instance02/config/server.policy -Djava.endorsed.dirs=/opt/glassfish4/glassfish/modules/endorsed:/opt/glassfish4/glassfish/lib/endorsed -Dfelix.fileinstall.bundles.startTransient=true -Dosgi.shell.telnet.maxconn=1 -Dfelix.fileinstall.log.level=3 -Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterprise.config.serverbeans.AppserverConfigEnvironmentFactory -Djavax.net.ssl.keyStore=/opt/glassfish4/Node02/Instance02/config/keystore.jks -Djava.security.auth.login.config=/opt/glassfish4/Node02/Instance02/config/login.conf -Dfelix.fileinstall.disableConfigSave=false -Dfelix.fileinstall.bundles.new.start=true -Dcom.sun.aas.instanceRoot=/opt/glassfish4/Node02/Instance02 -Dosgi.shell.telnet.port=26666 -Dgosh.args=-noshutdown -c noop=true -Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as -Dosgi.shell.telnet.ip=127.0.0.1 -DANTLR.USE_DIRECT_CLASS_LOADING=true -Djava.awt.headless=true -Djava.ext.dirs=/usr/lib/jvm/java-8-oracle/lib/ext:/usr/lib/jvm/java-8-oracle/jre/lib/ext:/opt/glassfish4/Node02/Instance02/lib/ext -Djdbc.driver=org.apache.derby.jdbc.ClientDriver -Djava.library.path=/opt/glassfish4/glassfish/lib:/usr/java/packages/lib/i386/lib:/usr/lib com.sun.enterprise.glassfish.bootstrap.ASMMain -upgrade false -read-stdin true -asadmin-args --host,,si2srv01,,--port,,4848,,--secure=false,,--terse=false,,--echo=false,,--interactive=false,,start-local-instance,,--verbose=false,,--watchdog=false,,--debug=false,,--nodedir,,/opt/glassfish4,,--node,,Node02,,Instance02 -instancename Instance02 -type INSTANCE -verbose false -instancedir /opt/glassfish4/Node02/Instance02 -asadmin-classpath /opt/glassfish4/glassfish/modules/admin-cli.jar -debug false -asadmin-classname com.sun.enterprise.admin.cli.AdminMain
0 R si2      2008 2002 0 80 0 - 465 -      11:52 pts/0      00:00:00 grep java
si2@si2srv03:~$
```

```
si2@si2srv02:~$ ps -aefl |grep java
0 S si2      1809      1 6 80 0 - 135797 futex 11:45 ?      00:00:23 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/glassfish4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRatio=2 -XX:MaxPermSize=96m -Xmx128m -Xms128m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.net.ssl.trustStore=/opt/glassfish4/Node01/Instance01/config/cacerts.jks -Djdk.corba.allowOutputStreamSubclass=true -Dfelix.fileinstall.dir=/opt/glassfish4/glassfish/modules/autostart/ -Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache.felix.gogo.shell,org.apache.felix.gogo.command,org.apache.felix.fileinstall -Dcom.sun.aas.installRoot=/opt/glassfish4/glassfish -Dfelix.fileinstall.poll=5000 -Djava.security.policy=/opt/glassfish4/Node01/Instance01/config/server.policy -Djava.endorsed.dirs=/opt/glassfish4/glassfish/modules/endorsed:/opt/glassfish4/glassfish/lib/endorsed -Dfelix.fileinstall.bundles.startTransient=true -Dosgi.shell.telnet.maxconn=1 -Dfelix.fileinstall.log.level=3 -Dcom.sun.enterprise.config.config_environment_factory_class=com.sun.enterprise.config.serverbeans.AppserverConfigEnvironmentFactory -Djavax.net.ssl.keyStore=/opt/glassfish4/Node01/Instance01/config/keystore.jks -Djava.security.auth.login.config=/opt/glassfish4/Node01/Instance01/config/login.conf -Dfelix.fileinstall.disableConfigSave=false -Dfelix.fileinstall.bundles.new.start=true -Dcom.sun.aas.instanceRoot=/opt/glassfish4/Node01/Instance01 -Dosgi.shell.telnet.port=26666 -Dgosh.args=-noshutdown -c noop=true -Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as -Dosgi.shell.telnet.ip=127.0.0.1 -DANTLR.USE_DIRECT_CLASS_LOADING=true -Djava.awt.headless=true -Djava.ext.dirs=/usr/lib/jvm/java-8-oracle/lib/ext:/usr/lib/jvm/java-8-oracle/jre/lib/ext:/opt/glassfish4/Node01/Instance01/lib/ext -Djdbc.driver=org.apache.derby.jdbc.ClientDriver -Djava.library.path=/opt/glassfish4/glassfish/lib:/usr/java/packages/lib/i386/lib:/usr/lib com.sun.enterprise.glassfish.bootstrap.ASMMain -upgrade false -read-stdin true -asadmin-args --host,,si2srv01,,--port,,4848,,--secure=false,,--terse=false,,--echo=false,,--interactive=false,,start-local-instance,,--verbose=false,,--watchdog=false,,--debug=false,,--nodedir,,/opt/glassfish4,,--node,,Node01,,Instance01 -instancename Instance01 -type INSTANCE -verbose false -instancedir /opt/glassfish4/Node01/Instance01 -asadmin-classpath /opt/glassfish4/glassfish/modules/admin-cli.jar -debug false -asadmin-classname com.sun.enterprise.admin.cli.AdminMain
0 S si2      2005 1999 0 80 0 - 465 pipe_w 11:51 pts/0      00:00:00 grep java
si2@si2srv02:~$
```

Podemos comprobar con esto que el cluster se ha creado de forma correcta.

Ejercicio 3

En primer lugar, realizamos todos los cambios que nos indica el enunciado, que incluyen desactivar el modo debug, añadir a tabla pago de la base de datos las columnas `instancia` e `ip` y modificar los distintos Beans y el `VisaDAO` para que tengan en cuenta estos dos nuevos campos. Del mismo modo cambiamos los archivos `postgresql.xml`, `build.properties`, y `postgresql.properties` como nos indica el enunciado para el despliegue correcto en el cluster.

Tras esto desplegamos la aplicación con `ant todo`.

Comprobamos desde la consola de administración de GlassFish los puertos de cada una de las instancias, observando que son el mismo entre las distintas distancias, y que solo varían entre el acceso con http o con https.

Web Application Links

If the server or listener is not running, the link may not work. In this event, check the status of the server instance. After launching the web application, use the browser's Back button to return to this screen.

Application Name: P3

Links:
[Instance01] <http://10.1.2.2:28080/P3>
[Instance01] <https://10.1.2.2:28181/P3>
[Instance02] <http://10.1.2.3:28080/P3>
[Instance02] <https://10.1.2.3:28181/P3>

Close

Hacemos un único pago en cada una de ellas, que se realizan correctamente, y comprobamos usando Tora que la información relativa a la dirección IP y nombre de la instancia se han almacenado en los nuevos campos correspondientes.

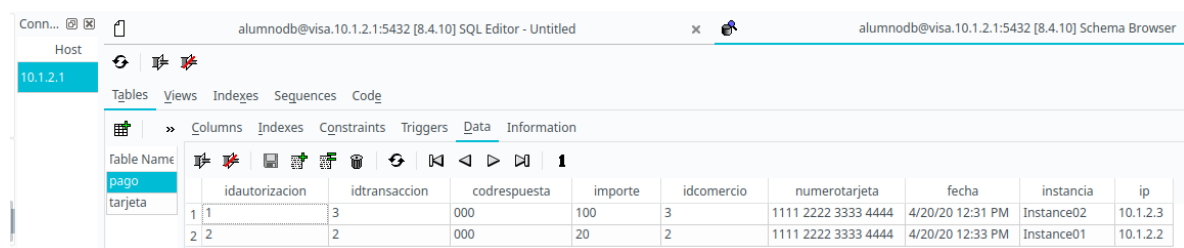


Table Name	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	instancia	ip
1	1	3	000	100	3	1111 2222 3333 4444	4/20/20 12:31 PM	Instance02	10.1.2.3
2	2	2	000	20	2	1111 2222 3333 4444	4/20/20 12:33 PM	Instance01	10.1.2.2

Vemos, de este modo, que el despliegue en el cluster se ha realizado correctamente y la aplicación funciona de la manera esperada.

Ejercicio 4

En primer lugar, tras eliminar todas las cookies del navegador, y sin la propiedad `jvmRoute`, comprobamos que al intentar realizar un pago obtenemos un error, y vemos que la información de la cookie almacenada es la siguiente:

Sistema de Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II

Inspector Console Debugger Network Style Editor Performance Memory Storage

Cache Storage Cookies Indexed DB Local Storage Session Storage

Name	Domain	Path	Expires	LastAccessed
JSESSIO...	10.1.2.1	/P3	Session	Mon, 20 Apr 2020 ...

Filter values

Data

JSESSIONID: "959de8413ed5691ebf8c5a6c0dde"

CreationTime: "Mon, 20 Apr 2020 20:28:28 GMT"

Domain: "10.1.2.1"

Expires: "Session"

HostOnly: true

HttpOnly: true

LastAccessed: "Mon, 20 Apr 2020 20:48:54 GMT"

Path: "/P3"

SameSite: "Unset"

Secure: false

Esto se debe a que el balanceador de carga envía al cliente a una de las instancias cada vez, lo que provoca que al intentar completar el pago, dicha instancia no tenga información de la sesión asociada a este cliente, y por tanto, falle.

Volvemos a borrar cookies, pero esta vez sí configuramos la propiedad *jvmRoute* y vemos que, en esta ocasión, sí podemos realizar pagos, obteniendo la siguiente cookie almacenada:

Sistema de Pago con tarjeta

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

idTransaccion: 1
idComercio: 1
importe: 1.0
codRespuesta: 000
idAutorizacion: 1

[Volver al comercio](#)

Inspector Console Debugger Network Style Editor Performance Memory Storage

Cache Storage Cookies Indexed DB Local Storage Session Storage

Name	Domain	Path	Expires	LastAccessed
JSESSIO...	10.1.2.1	/P3	Session	Mon, 20 Apr 2020 ...

Filter values

Data

JSESSIONID: "95fbff5623ab50b09...16418.Instance01"

CreationTime: "Mon, 20 Apr 2020 20:54:51 GMT"

Domain: "10.1.2.1"

Expires: "Session"

HostOnly: true

HttpOnly: true

LastAccessed: "Mon, 20 Apr 2020 20:55:19 GMT"

Path: "/P3"

SameSite: "Unset"

Secure: false

Parsed Value

JSESSIONID: Array

0: "95fbff5623ab50b09abec4316418"

1: "Instance01"

length: 2

__proto__: Array

Como se puede comprobar la diferencia entre ambas cookies radica (además de en parámetros obvios como la hora) en la variable *JSESSIONID*, como era de esperar. En la primera prueba, esta variable contiene tan solo un simple identificador que no determina la instancia del cluster en la que se está ejecutando la petición, mientras que en el segundo caso, el parámetro *JSESSIONID* está compuesto de *identificador.nombreDeLaInstancia*.

En general, no se puede usar el valor `${com.sun.aas.hostName}` para la propiedad `jvmRoute` en lugar de `${com.sun.aas.instanceName}` pues no tienen el mismo contenido. `${com.sun.aas.hostName}` contiene el nombre de la máquina virtual que es el host del servicio, mientras que `${com.sun.aas.instanceName}` contiene el nombre de la instancia en la que se ejecuta el servicio. Si bien ambos parámetros representan la máquina en que se está ejecutando el servicio, `jvmRoute` necesita el nombre de la instancia y no el de la máquina virtual. Por tanto el único caso en que se podrían usar ambos valores indistintamente es si el nombre de las instancias y de la máquina virtual en que se ejecutan es el mismo, y en nuestro caso, no lo es, pues las máquinas se llaman `si2srvx` y las instancias creadas `Instance0X`. Alternativamente al nombre de la instancia, `jvmRoute` puede emplear también el nombre del "worker" correspondiente a la instancia en caso de que se haya creado un fichero `workers.properties`, pero, como este no es nuestro caso, no lo tomamos en cuenta.

Ejercicio 5

Enviamos varias peticiones de pago contra el cluster desde varios navegadores como se nos indica en el enunciado. Tras esto, consultamos el *Load Balancer Manager* y vemos que cada uno de los navegadores empleados almacena un *JSESSIONID* que hace referencia a una de las 2 instancias del cluster, y que permanece constante dentro de cada navegador. Esto quiere decir que cada navegador realiza sus peticiones a una de las dos instancias, pero siempre la misma dentro de cada navegador.

Load Balancer Manager for 10.1.2.1

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://10.1.2.2:28080	Instance01		1	0	Ok	24	15K	24K
http://10.1.2.3:28080	Instance02		1	0	Ok	21	13K	18K

Apache/2.2.14 (Ubuntu) Server at 10.1.2.1 Port 80

Esto nos indica que la propiedad `jvmRoute` funciona correctamente y la carga se balancea de la forma esperada.

Ejercicio 6

A partir de la captura anterior, vemos que la instancia 2 es la que tiene menos elecciones. Usamos `ps -aef | grep java` para saber cual es el pid del proceso, y lo ejecutamos con `kill -9 pid`.


```
si2@si2srv03:~$ ps -aef | grep java
si2      3303      1  2 13:52 ?        00:00:35 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/glassfish4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRatio=2 -XX:MaxPermSize=96m -Xmx128m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.net.ssl.trustStore=/opt/glassfish4/Node02/Instance02/config/cacerts.jks -Djdk.corba.allowOutputStreamSubclass=true -Dfelix.fileinstall.dir=/opt/glassfish4/glassfish/modules/autostart/ -Dorg.glassfish.additionalOSGiBundlesToStart=org.apache.felix.shell,org.apache.felix.gogo.runtime,org.apache.felix.gogo.shell,org.apache.felix.gogo.command,org.apache.felix.fileinstall -Dcom.sun.aas.installRoot=/opt/glassfish4/glassfish -Dfelix.fileinstall.poll=5000 -Djava.security.policy=/opt/glassfish4/Node02/Instance02/config/serve
r.policy -Djava.endorsed.dirs=/opt/glassfish4/glassfish/modules/endorsed:/opt/glassfish4/glassfish/lib/endorsed -Dfelix.fileinstall.bundles.startTransient=true -Dosgi.shell.telnet.maxconn=1 -Dfelix.fileinstall.log.level=3 -Dcom.sun.enterprise.config.config.environment.factory.class=com.sun.enterprise.config.serverbeans.AppserverConfigEnvironmentFactory -Djavax.net.ssl.keyStore=/opt/glassfish4/Node02/Instance02/config/keystore.jks -Djava.security.auth.login.config=/opt/glassfish4/Node02/Instance02/config/login.conf -Dfelix.fileinstall.disableConfigSave=false -Dfelix.fileinstall.bundle
s.new.start=true -Dcom.sun.aas.instanceRoot=/opt/glassfish4/Node02/Instance02 -Dosgi.shell.telnet.port=26666 -Dgosh.args=-noshutdown -c noop=true -Dcom.sun.enterprise.security.httpsOutboundKeyAlias=s1as -Dosgi.shell.telnet.ip=127.0.0.1 -DANTLR_USE_DIRECT_CLASS_LOADING=true -Djava.awt.headless=true -Djava.ext.dirs=/usr/lib/jvm/java-8-oracle/lib/ext:/usr/lib/jvm/java-8-oracle/jre/lib/ext:/opt/glassfish4/Node02/Instance02/lib/ext -Djdbc.drivers=org.apache.derby.jdbc.ClientDriver -Djava.library.path=/opt/glassfish4/glassfish/lib:/usr/java/packages/lib/i386:/lib:/usr/lib com.sun.enterprise.glassfish.bootstrap.ASMain -upgrade false -read-stdin true -asadmin-args -host,,si2srv01,--port,,4848,--security=false,--terse=false,--echo=false,--interactive=false,--start-local-instance,--verbose=false,--watchdog=false,--debug=false,--nodedir,,/opt/glassfish4,--node,,Node02,--Instance02 -instancename Instance02 -type INSTANCE -verbose false -instancedir /opt/glassfish4/Node02/Instance02 -asadmin-classpath /opt/glassfish4/glassfish/modules/admin-cli.jar -debug false -asadmin-classname com.sun.enterprise.admin.cli.AdminMain
si2      3573    3566  0 14:18 pts/0    00:00:00 grep java
si2@si2srv03:~$ kill -9 3303
si2@si2srv03:~$ kill -9 3573
-bash: kill: (3573) - No such process
si2@si2srv03:~$ kill -9 3566
Connection to 10.1.2.3 closed.
```

Tras esto, plasmamos el estado de la base de datos antes de ejecutar ningún pago nuevo:

Tables Views Indexes Sequences Code									
Columns Indexes Constraints Triggers Data Information									
Table Name									
pago	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	instancia	ip
tarjeta									
1	1	1	000	1	1	1111 2222 3333 4444	4/20/20 1:55 PM	Instance01	10.1.2.2
2	2	2	000	10	1	4579 1165 7919 4384	4/20/20 2:06 PM	Instance02	10.1.2.3
3	3	3	000	30	3	6496 3318 3339 9724	4/20/20 2:08 PM	Instance01	10.1.2.2
4	4	6	000	12	1	5804 7661 8556 1810	4/20/20 2:11 PM	Instance01	10.1.2.2
5	5	5	000	10	5	5804 7661 8556 1810	4/20/20 2:12 PM	Instance02	10.1.2.3

Usamos nuevas ventanas de incógnito dentro de los navegadores para hacer más pagos, y podemos comprobar de esta forma que la instancia 2 no está funcionando pues: todos los nuevos pagos aparecen en la base de datos como ejecutados en la instancia 1, y en el balanceador de carga aparece con error.

Tables Views Indexes Sequences Code									
Columns Indexes Constraints Triggers Data Information									
Table Name									
pago	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	instancia	ip
tarjeta									
1	1	1	000	1	1	1111 2222 3333 4444	4/20/20 1:55 PM	Instance01	10.1.2.2
2	2	2	000	10	1	4579 1165 7919 4384	4/20/20 2:06 PM	Instance02	10.1.2.3
3	3	3	000	30	3	6496 3318 3339 9724	4/20/20 2:08 PM	Instance01	10.1.2.2
4	4	6	000	12	1	5804 7661 8556 1810	4/20/20 2:11 PM	Instance01	10.1.2.2
5	5	5	000	10	5	5804 7661 8556 1810	4/20/20 2:12 PM	Instance02	10.1.2.3
6	7	7	000	10	1	4579 1165 7919 4384	4/20/20 2:20 PM	Instance01	10.1.2.2
7	8	8	000	100	10	1725 7286 0489 3263	4/20/20 2:20 PM	Instance01	10.1.2.2
8	9	9	000	30	9	2641 2366 4022 9561	4/20/20 2:21 PM	Instance01	10.1.2.2
9	11	11	000	10	1	1111 2222 3333 4444	4/20/20 2:27 PM	Instance01	10.1.2.2

Load Balancer Manager for 10.1.2.1

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

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONIDjsessionid	0	1	byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.1.2.2:28080	Instance01		1	0	Ok	35	22K 36K
http://10.1.2.3:28080	Instance02		1	0	Err	22	13K 18K

Apache/2.2.14 (Ubuntu) Server at 10.1.2.1 Port 80

Ejercicio 7

Usamos el comando `asadmin start-instance Instance02` para arrancar la instancia 2, hacemos más pagos, y podemos comprobar, tanto en la base de datos como en la página del balanceador de carga que la instancia 2 está activa otra vez.

Database Navigator										
Tables Views Indexes Sequences Code										
Columns Indexes Constraints Triggers Data Information										
Table Name										
pago										
tarjeta										
	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	instancia	ip	
1	1	1	000	1	1	1111 2222 3333 4444	4/20/20 1:55 PM	Instance01	10.1.2.2	
2	2	2	000	10	1	4579 1165 7919 4384	4/20/20 2:06 PM	Instance02	10.1.2.3	
3	3	3	000	30	3	6496 3318 3339 9724	4/20/20 2:08 PM	Instance01	10.1.2.2	
4	4	6	000	12	1	5804 7661 8556 1810	4/20/20 2:11 PM	Instance01	10.1.2.2	
5	5	5	000	10	5	5804 7661 8556 1810	4/20/20 2:12 PM	Instance02	10.1.2.3	
6	6	7	000	10	1	4579 1165 7919 4384	4/20/20 2:20 PM	Instance01	10.1.2.2	
7	7	8	000	100	10	1725 7286 0489 3263	4/20/20 2:20 PM	Instance01	10.1.2.2	
8	8	9	000	30	9	2641 2366 4022 9561	4/20/20 2:21 PM	Instance01	10.1.2.2	
9	9	11	000	10	1	1111 2222 3333 4444	4/20/20 2:27 PM	Instance01	10.1.2.2	
10	10	12	000	20	2	1111 2222 3333 4444	4/20/20 2:35 PM	Instance02	10.1.2.3	
11	11	13	000	30	4	4579 1165 7919 4384	4/20/20 2:36 PM	Instance02	10.1.2.3	

Load Balancer Manager for 10.1.2.1

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

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONIDjsessionid 0	1		byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.1.2.2:28080	Instance01	1	0	Ok	35	22K	36K
http://10.1.2.3:28080	Instance02	1	0	Ok	28	17K	25K

Apache/2.2.14 (Ubuntu) Server at 10.1.2.1 Port 80

Ejercicio 8

Como se nos indica en el enunciado, comenzamos una petición de pago y tras realizar la llamada al servlet `ComienzaPago`, comprobamos que ésta comienza en la instancia 2. A su vez, el Load Balancer nos muestra que ambas instancias están funcionando de forma correcta y que las peticiones se están balanceando correctamente:

Load Balancer Manager for 10.1.2.1

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

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONIDjsessionid 0	1		byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.1.2.2:28080	Instance01	1	0	Ok	36	23K	36K
http://10.1.2.3:28080	Instance02	1	0	Ok	32	20K	30K

Apache/2.2.14 (Ubuntu) Server at 10.1.2.1 Port 80

Entonces, detenemos la instancia dos del cluster, y comprobamos como tras completar el pago y recibir el mensaje de "Pago incorrecto", el Load Balancer nos muestra error en la instancia 2 ya que la hemos detenido, y el `JSESSIONID` almacenado en la cookie del navegador se refiere ahora a la instancia 1.

Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II

The screenshot shows the Chrome DevTools Storage Inspector. The left sidebar lists storage types: Cache Storage, Cookies, Indexed DB, Local Storage, and Session Storage. The 'Cookies' section is expanded, showing a table with columns: Name, Domain, Path, Expires, and LastAccessed. A single cookie is listed with Name 'JSESSIONID', Domain '10.1.2.1', Path '/P3', Expires 'Session', and LastAccessed 'Mon, 20 Apr 2020 ...'. The right pane shows the 'Data' tab for this cookie, displaying its value as a JSON string: {"JSESSIONID": "98ab9851ea0ed3a9...6a577.Instance01", "CreationTime": "Mon, 20 Apr 2020 21:39:41 GMT", "Domain": "10.1.2.1", "Expires": "Session", "HostOnly": true, "HttpOnly": true, "LastAccessed": "Mon, 20 Apr 2020 21:42:16 GMT", "Path": "/P3", "SameSite": "Unset", "Secure": false}. Below this, the 'Parsed Value' tab shows the cookie's structure as an object with a 'JSESSIONID' property containing an array of two elements: a long hexadecimal string and 'Instance01'.

Load Balancer Manager for 10.1.2.1

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://10.1.2.2:28080	Instance01		1	0	Ok	37	23K	37K
http://10.1.2.3:28080	Instance02		1	0	Err	33	20K	30K

Apache/2.2.14 (Ubuntu) Server at 10.1.2.1 Port 80

El pago realizado en esta operación resulta incorrecto debido a que, para completarse el pago, se necesita tener el pago almacenado en la sesión, y este no está disponible en la instancia 1 que es la que se ha encargado de finalizar el pago, puesto que la operación del pago ha sido iniciada por la instancia 2 que es la que tendrá estos datos almacenados en sesión.

Ejercicio 9

Para realizar estas pruebas correctamente, en primer lugar eliminamos todos los pagos previos al ciclo de pruebas, para ello eliminamos la base de datos empleando Tora.

Modificamos el script p2.jmx, entregado ahora como p3.jmx.

Configuramos ahora la pantalla de pruebas de JMeter para realizar un ciclo de 1000 pruebas en un solo hilo a la IP del cluster y la nueva URL como nos indica el enunciado, para ello establecemos la IP a `10.1.2.1`, borramos el Port Number, dejamos un único Thread Group que se llama P3, establecemos un único hilo y 1000 pruebas (estableciendo la variable *samples* a 1000) e introducimos en el Path la dirección *"P3/procesapago"*.

Tras estos cambios, corremos el test y vemos que se ejecuta correctamente.

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename: Browse...

Log/Display Only: ☐ Errors ☒ Successes ☐ Configure

Label	# Samples	Average	Median	90% Line	95% Line	99% Line	Min	Maximum	Error %	Throughput	Received KB	Sent KB/sec
P3	1000	9	7	9	9	12	5	1947	0.00%	103.4/sec	142.09	0.00
TOTAL	1000	9	7	9	9	12	5	1947	0.00%	103.4/sec	142.09	0.00

Usando Tora, comprobamos que el ciclo de tests se ha ejecutado correctamente comprobando que se han realizado 1000 pagos, 500 con cada una de las 2 instancias.

Conn... alumnodb@visa.10.1.2.1:5432 [8.4.10] SQL Editor - *Untitled

Host: 10.1.2.1

1 `SELECT COUNT(*) FROM PAGO;`

Result Execution plan Visualize Logging

#	count
1	1000

Host: 10.1.2.1

`SELECT COUNT(*) FROM PAGO WHERE IP='10.1.2.2';`

Result Execution plan Visualize Logging

#	count
1	500

Host: 10.1.2.1

`SELECT COUNT(*) FROM PAGO WHERE IP='10.1.2.3';`

Result Execution plan Visualize Logging

#	count
1	500

Observando detalladamente la distribución de los pagos realizados, vemos que el algoritmo manda un pago a cada una de las 2 instancias de forma alterna, distribuyendo estos de forma uniforme entre ambas, de modo que la carga de ambas instancias sea la misma.

