		Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2			
Grupo	2321	Práctica	3	Fecha	03/05/2021
Alumno/a	Arribas Gonzalo, Francisco Javier				
Alumno/a	Saénz Ferrero, Santos				

Práctica 3: Seguridad y disponibilidad

Ejercicio número 1:

Preparar 3 máquinas virtuales 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 10.7.8.1, 768MB RAM
- si2srv02: Dirección IP 10.7.8.2, 512MB RAM
- si2srv03: Dirección IP 10.7.8.3, 512MB RAM

Recuerde randomizar las direcciones MAC de cada copia antes de intentar usar el nodo.

En la primera máquina (10.7.8.1), generaremos el par de claves con DSA. A continuación importaremos la clave pública en cada uno de los otros dos nodos (10.7.8.2 y 10.7.8.3). Porbaremos a acceder por SSH desde 10.7.8.1 a 10.7.8.2 y 10.7.8.3, comprobando que no requiere la introducción de la clave. Obtener una evidencia del inicio remoto de sesión mediante la salida detallada (ssh -v [si2@10.7.8.2](#) y ssh -v [si2@10.7.8.2](#)). Anote dicha salida en la memoria de prácticas.

Se han creado las claves DSA públicas y privadas, añadiendo las claves públicas a si2srv02 y si2srv03, y asegurándose de que la clave pública esté en la lista de claves autorizadas por la máquina.

Después, mediante SSH, se establece una conexión con el puerto 22 del servidor. Se comprueba la clave que usa el cliente y se mira a ver si esta en la lista negra e claves del servidor. Por último, antes de aceptar la conexión el servidor verifica que la clave publica encaja con la clave privada.

La salida de la conexión con la segunda terminal (**10.7.8.2**) se recoge en la siguiente imagen:

```
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 May 3 07:49:03 2021 from 10.7.8.1
Loading es
si2@si2srv02:~$
```

La salida de la conexión con la tercera terminal se recoge en la siguiente imagen:

```
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 May 3 07:24:15 2021
Loading es
si2@si2srv03:~$
```

Una vez realizado este punto, detendremos las tres máquinas virtuales y obtendremos una copia de las mismas a algún medio externo (USB) para los consiguientes apartados de esta práctica. También es recomendable que preserve los directorios .ssh de cada uno de los nodos.

Ejercicio número 2:

Realizar los pasos del apartado 4 con el fin de obtener una configuración válida del *cluster* SI2Cluster, con la topología 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) correspondientes están efectivamente corriendo en cada una de las dos máquinas virtuales. Adjunte evidencias a la memoria de la práctica.

Como se ha indicado en el apartado 4, hemos iniciado el dominio domain1, hemos creado el **nodo Node01** en la máquina virtual 10.7.8.2 y el **nodo Node02** en la máquina virtual 10.7.8.3. Después de poder realizar el comando ping-node, verificando la correcta creación de ambos nodos, hemos **creado el cluster SI2Cluster**. Se ha comprobado que las máquinas virtuales se “conocen” entre ellas comprobando el fichero etc/hosts, y entonces hemos creado las instancias **Instance01** para Node01 en 10.7.8.2 e **Instance02** para Node02 en 10.7.8.3.

Las evidencias acerca de la correcta realización de este ejercicio se reflejan en las siguientes capturas.

Lista de instancias en el *cluster* SI2Cluster:

```
si2@si2srv01:~$ asadmin list-instances -l
Name      Host      Port  Pid  Cluster  State
Instance01 10.7.8.2  24848 2162 SI2Cluster  running
Instance02 10.7.8.3  24848 2120 SI2Cluster  running
Command list-instances executed successfully.
```

Como se aprecia, los *pid* de los procesos java efectivamente están corriendo en cada una de las dos máquinas virtuales:

```

sl2@sl2srv02:~$ ps -aefl | grep java
0 S sl2 2162 1 4 80 0 - 235717 futex_10:06 ? 00:00:19 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/glassfish4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRat
lo=2 -XX:MaxPermSize=192m -Xmx512m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.net.ssl.trustStore=/opt/glassfish4/glassfish/nodes/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.go
go.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.endorsed.dirs=/op
t/glassfish4/glassfish/modules/endorsed:/opt/glassfish4/glassfish/lib/endorsed -Djava.security.policy=/opt/glassfish4/glassfish/nodes/Node01/Instance01/config/server.policy -Dosgi.shell.telnet.maxconn=1 -
Dfelix.fileinstall.bundles.startTransient=true -Dcom.sun.enterprise.config.config_environment_factory.class=com.sun.enterprise.config.serverbeans.AppserverConfigEnvironmentFactory -Dfelix.fileinstall.log
level=3 -Djavax.net.ssl.keyStore=/opt/glassfish4/glassfish/nodes/Node01/Instance01/config/keystore.jks -Djava.security.auth.login.config=/opt/glassfish4/glassfish/nodes/Node01/Instance01/config/login.conf
-Dfelix.fileinstall.disableConfigSave=false -Dfelix.fileinstall.bundles.new.start=true -Dcom.sun.aas.instanceRoot=/opt/glassfish4/glassfish/nodes/Node01/Instance01 -Dosgi.shell.telnet.port=26666 -Dgosh.a
rgs=-noshutdown -c noop=true -Dcom.sun.enterprise.security.httpsOutboundKeyAlias=sias -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/glassfish/nodes/Node01/Instance01/lib/ext -Djdbc.drivers=org.apache.derby.jdbc.ClientDriver -Djava.library.path=/opt/gl
assfish4/glassfish/lib:/usr/java/packages/lib/i386:/lib:/usr/lib com.sun.enterprise.glassfish.bootstrap.ASMain -upgrade false -read-stdin true -asadmin-args -host,,sl2srv01,,--port,,4848,,--secure=fa
lse,,--terse=false,,--echo=false,,--interactive=false,,--start-local-instance,,--verbose=false,,--watchdog=false,,--debug=false,,--node,,Node01,,Instance01 -instancename Instance01 -type INSTANCE
-verbose false -instancedir /opt/glassfish4/glassfish/nodes/Node01/Instance01 -asadmin-classpath /opt/glassfish4/glassfish/modules/admin-cl.jar -debug false -asadmin-classname com.sun.enterprise.admin.cl
i.AdminMain
0 S sl2 2349 2342 0 80 0 - 465 pipe_w 10:14 pts/0 00:00:00 grep java

```

```

sl2@sl2srv03:~$ ps -aefl | grep java
0 S sl2 2120 1 3 80 0 - 234804 futex_10:06 ? 00:00:20 /usr/lib/jvm/java-8-oracle/bin/java -cp /opt/glassfish4/glassfish/modules/glassfish.jar -XX:+UnlockDiagnosticVMOptions -XX:NewRat
lo=2 -XX:MaxPermSize=192m -Xmx512m -server -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.net.ssl.trustStore=/opt/glassfish4/glassfish/nodes/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.go
go.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.endorsed.dirs=/op
t/glassfish4/glassfish/modules/endorsed:/opt/glassfish4/glassfish/lib/endorsed -Djava.security.policy=/opt/glassfish4/glassfish/nodes/Node02/Instance02/config/server.policy -Dosgi.shell.telnet.maxconn=1 -
Dfelix.fileinstall.bundles.startTransient=true -Dcom.sun.enterprise.config.config_environment_factory.class=com.sun.enterprise.config.serverbeans.AppserverConfigEnvironmentFactory -Dfelix.fileinstall.log
level=3 -Djavax.net.ssl.keyStore=/opt/glassfish4/glassfish/nodes/Node02/Instance02/config/keystore.jks -Djava.security.auth.login.config=/opt/glassfish4/glassfish/nodes/Node02/Instance02/config/login.conf
-Dfelix.fileinstall.disableConfigSave=false -Dfelix.fileinstall.bundles.new.start=true -Dcom.sun.aas.instanceRoot=/opt/glassfish4/glassfish/nodes/Node02/Instance02 -Dosgi.shell.telnet.port=26666 -Dgosh.a
rgs=-noshutdown -c noop=true -Dcom.sun.enterprise.security.httpsOutboundKeyAlias=sias -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/glassfish/nodes/Node02/Instance02/lib/ext -Djdbc.drivers=org.apache.derby.jdbc.ClientDriver -Djava.library.path=/opt/gl
assfish4/glassfish/lib:/usr/java/packages/lib/i386:/lib:/usr/lib com.sun.enterprise.glassfish.bootstrap.ASMain -upgrade false -read-stdin true -asadmin-args -host,,sl2srv01,,--port,,4848,,--secure=fa
lse,,--terse=false,,--echo=false,,--interactive=false,,--start-local-instance,,--verbose=false,,--watchdog=false,,--debug=false,,--node,,Node02,,Instance02 -instancename Instance02 -type INSTANCE
-verbose false -instancedir /opt/glassfish4/glassfish/nodes/Node02/Instance02 -asadmin-classpath /opt/glassfish4/glassfish/modules/admin-cl.jar -debug false -asadmin-classname com.sun.enterprise.admin.cl
i.AdminMain
0 S sl2 2339 2332 0 80 0 - 465 pipe_w 10:17 pts/0 00:00:00 grep java

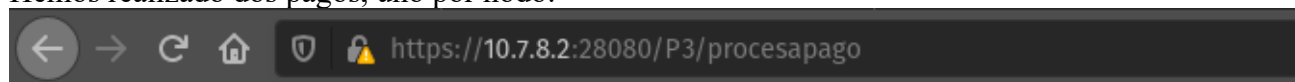
```

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 10.7.8.2 y 10.7.8.3 respectivamente). Puede realizar esa comprobación directamente desde la consola de administraci3n, opci3n Applications, acci3n Launch, observando los Web Application Links generados.

Realice un 3nico pago en cada nodo. Verifique que el pago se ha anotado correctamente el nombre de la instancia y la direcci3n IP. Anote sus observaciones (puertos de cada instancia) y evidencias (captura de pantalla de la tabla de pagos).

Hemos realizado dos pagos, uno por nodo:



Pago con tarjeta

Pago realizado con 3xito. A continuaci3n se muestra el comprobante del mismo:

```

idTransaccion: 1
idComercio: 1
importe: 10.0
codRespuesta: 000
idAutorizacion: 1

```

[Volver al comercio](#)

Pago con tarjeta

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

idTransaccion: 2
idComercio: 2
importe: 2.0
codRespuesta: 000
idAutorizacion: 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

Verificación:

	idautorizacion [PK] serial	idtransaccion character(16)	codrespuesta character(3)	importe double precision	idcomercio character(16)	numerotarjeta character(19)	fecha timestamp without time zone	instancia character(50)	ip character(50)
1	1	1	000	1	1	4579 1165 7919 4384	2021-05-05 19:31:17.224706	Instance01	10.7.8.2
2	2	2	000	2	2	4579 1165 7919 4384	2021-05-05 19:34:40.768227	Instance02	10.7.8.3
*									

Podemos observar que se usa el puerto 28080.

Ejercicio número 4:

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://10.7.8.1/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.**
- 5. Mostrar la cookie “JSESSIONID” correspondiente a la URL del balanceador donde se vea:**

Name: JSESSIONID
Content: YYYYYYYYYYYYYYYYYYYY
Domain: 10.7.8.1
Path: /P3

- 6. Añadir la propiedad “jvmRoute” al cluster y rearrancar el cluster.**
- 7. Eliminar todas las cookies del navegador.**
- 8. Acceso a la aplicación P3 a través de la URL del balanceador:**
http://10.7.8.1/P3
- 9. Completar el pago con datos de tarjeta correctos. Se pueden repetir los pagos y no fallarán.**
- 10. Mostrar la cookie “JSESSIONID” correspondiente a la URL del balanceador donde se vea:**

Name: JSESSIONID
Content: ZZZZZZZZZZZZZZZZZZZZZZZ
Domain: 10.7.8.1
Path: /P3

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

Las diferencias entre cookies están en que cuando no se establece la variable jvm route el contenido de la cookie solo contiene el identificador de sesión mientras que en el otro caso contiene además del identificador de sesión concatenado a este la instancia que se encarga de procesar sus peticiones. En el primer caso se pueden producir errores debido a que la información relativa a la sesión de un usuario puede no corresponderse a la información de la sesión que contiene la instancia a la cual se le ha realizado una petición a menos que se haga uso de algún algoritmo costoso computacionalmente, GMS en el caso de Glassfish. Sin embargo en el segundo caso al desviarse siempre las peticiones a la misma instancia esta va a tener los datos relativos a la sesión actualizados pues no atendería ninguna petición para ese cliente ninguna otra instancia.

El las siguientes imágenes se muestran las cookies del servidor cuando no se hace uso de jvm route y cuando si.

← 10.7.8.1 ha almacenado datos de forma local

Eliminar todo

JSESSIONID

^ | ✕

Nombre	JSESSIONID
Contenido	1fae37e67e3f50e9073b572e9526
Dominio	10.7.8.1
Ruta	/P3
Enviar para	Solo conexiones al mismo sitio web
Accesible para secuencia de comandos	No (HttpOnly)
Creada	Jueves, 6 de mayo de 2021, 16:00:32
Caduca	Al finalizar la sesión de navegación

← 10.7.8.1 ha almacenado datos de forma local

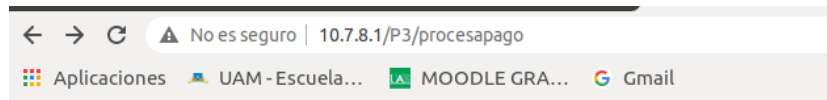
Eliminar todo

JSESSIONID

^ | ✕

Nombre	JSESSIONID
Contenido	2115fdd8f8c040b26df13ce4f344.Instance02
Dominio	10.7.8.1
Ruta	/P3
Enviar para	Solo conexiones al mismo sitio web
Accesible para secuencia de comandos	No (HttpOnly)
Creada	Jueves, 6 de mayo de 2021, 16:25:05
Caduca	Al finalizar la sesión de navegación

Imagen del fallo en el pago por la afinidad de las sesiones

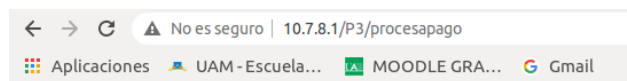


Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II

Pago correcto en el primer intento cuando no hay jvm reoute y cuando se estableció esta propiedad en la consola de administración de Glassfish.



Pago con tarjeta

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

idTransaccion: 123
idComercio: 123
importe: 10.0
codRespuesta: 000
idAutorizacion: 4

[Volver al comercio](#)



Pago con tarjeta

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

idTransaccion: 11
idComercio: 11
importe: 111.0
codRespuesta: 000
idAutorizacion: 9

[Volver al comercio](#)

Ejercicio número 5:

Probar el balanceo de carga y la afinidad de sesión, realizando un pago directamente contra la dirección del *cluster* <http://10.7.8.1/P3> desde distintos ordenadores. Comprobar que las peticiones se reparten entre ambos nodos del *cluster*, y que mantiene la sesión iniciada por cada usuario sobre el mismo nodo.

A partir de las siguientes imágenes se puede verificar como el load *balancer* mantiene los identificadores de sesión, mostrados en la primera tabla, y se aprecia en la segunda tabla que las peticiones se reparten entre ambos nodos, pues la primera imagen se corresponde a cuando se accedió desde un ordenador y la segunda desde otro ordenador más tarde, en *elected* se indica que para el segundo caso la primera instancia ha sido elegida mas veces por que antes de esta interacción había sido por el contrario la segunda instancia elegida mas veces

Load Balancer Manager for 10.7.8.1									
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[jsessionid0]	1		byrequests						
Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	To	From	
http://10.7.8.2:28080	Instance01	1	0	Ok	51	36K	83K		
http://10.7.8.3:28080	Instance02	1	0	Ok	54	41K	78K		

Apache/2.2.14 (Ubuntu) Server at 10.7.8.1 Port 80

Load Balancer Manager for 10.7.8.1									
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[jsessionid0]	1		byrequests						
Worker URL	Route	RouteRedir	Factor	Set	Status	Elected	To	From	
http://10.7.8.2:28080	Instance01	1	0	Ok	59	42K	91K		
http://10.7.8.3:28080	Instance02	1	0	Ok	55	41K	79K		

Apache/2.2.14 (Ubuntu) Server at 10.7.8.1 Port 80

Ejercicio número 6:

Comprobación del proceso de *fail-over*.

Parar la instancia del cluster que haya tenido menos elecciones hasta el momento. Para ello, identificaremos el pid (identificador del proceso java) de la instancia usando las herramientas descritas en esta práctica o el mandato “ps -aef | 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.

A partir del acceso a *Load Balancer* puesto que en la segunda tabla la segunda instancia, referenciada por la columna *route*, es la que menos accesos ha tenido, valor dado por la columna *elected*, deberemos parar esta instancia.

Para ello por un lado accedemos al terminal correspondiente al segundo nodo mediante `ssh -v si2@10.7.8.3`, por otra parte para hallar el identificador del proceso java usamos `ps -aef | grep java`, y obtenemos el id del proceso correspondiente a la segunda instancia, 3840, a partir de este id finalizamos el proceso de la instancia mediante `kill -9 3840`.

Ahora se puede apreciar como el valor de la columna *status* para la segunda instancia es Err y no Ok, lo cual indica que la instancia no se encuentra disponible. Realizamos pagos desde los diferentes ordenadores y todas ellas se van a redirigir hacia la primera instancia pues su columna *elected* aumenta mientras que para la segunda instancia no cambia ya que no se dirige a ella ninguna petición. Esto también se verifica en la base de datos donde se muestra que todas los pagos corresponden a la instancia uno pues la dos se detuvo.

[←](#) [→](#) [🔄](#) [🔒](#) No es seguro | 10.7.8.1/balancer-manager

Load Balancer Manager for 10.7.8.1

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://10.7.8.2:28080	Instance01		1	0	Ok	66	47K	96K
http://10.7.8.3:28080	Instance02		1	0	Err	56	41K	79K

Apache/2.2.14 (Ubuntu) Server at 10.7.8.1 Port 80

[←](#) [→](#) [🔄](#) [🔒](#) No es seguro | 10.7.8.1/balancer-manager

Load Balancer Manager for 10.7.8.1

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://10.7.8.2:28080	Instance01		1	0	Ok	74	52K	111K
http://10.7.8.3:28080	Instance02		1	0	Err	56	41K	79K

Apache/2.2.14 (Ubuntu) Server at 10.7.8.1 Port 80

5	9	11	000	111	11	1111 2222 3333 4444	06/05/21 07:41	Instance01	10.7.8.2
6	10	24	000	21	24	1111 2222 3333 4444	06/05/21 07:52	Instance01	10.7.8.2
7	11	26	000	21	26	1111 2222 3333 4444	06/05/21 07:53	Instance01	10.7.8.2
8	12	17	000	12	17	1111 2222 3333 4444	06/05/21 08:43	Instance01	10.7.8.2
9	13	41	000	41	41	1111 2222 3333 4444	06/05/21 08:45	Instance01	10.7.8.2
10	14	242	000	212	2442	1111 2222 3333 4444	06/05/21 08:46	Instance01	10.7.8.2

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.

Tras volver a iniciar la instancia se verifica el funcionamiento de esta pues ahora la columna *status* para esta instancia pasa a *OK* y las peticiones de los diferentes ordenadores se van a redirigir a esta instancia ya que tiene menos peticiones procesadas que la primera, algo que se refleja en que la columna *elected* aumenta para la instancia 2, mientras que para la instancia 1 se mantiene. Esto también se verifica en la base de datos pues para la tabla de pagos la instancia correspondiente a cada uno de estos nuevos pagos desde diferentes ordenadores es la instancia 2.

← → ↻ ① No es seguro 10.7.8.1/balancer-manager	
Load Balancer Manager for 10.7.8.1	
Server Version: Apache/2.2.14 (Ubuntu) Server Built: Nov 3 2011 03:31:27	
LoadBalancer Status for balancer://si2cluster	
StickySession	Timeout Fallover:Attempts Method
JSESSIONIDjsessionid 0	1 byrequests
Worker URL	Route RouteRedir Factor Set Status Elected To From
http://10.7.8.2:28080	Instance01 1 0 Ok 76 54K 113K
http://10.7.8.3:28080	Instance02 1 0 Ok 60 44K 81K

12	17	32	000	32	32	1111 2222 3333 4444	06/05/21 09:07	Instance02	10.7.8.3
13	18	101	000	101	101	1111 2222 3333 4444	06/05/21 09:13	Instance02	10.7.8.3
14	19	102	000	102	102	1111 2222 3333 4444	06/05/21 09:13	Instance02	10.7.8.3
15	20	103	000	103	103	1111 2222 3333 4444	06/05/21 09:14	Instance02	10.7.8.3
16	21	104	000	104	104	1111 2222 3333 4444	06/05/21 09:14	Instance02	10.7.8.3