		Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2			
Grupo	2312	Práctica	3	Fecha	07/05/2021
Alumno/a		San Felipe Martin, Adrián			
Alumno/a		Durán Díaz, Luis Miguel			

Práctica 3: Seguridad y disponibilidad

Ejercicio 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.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), generaremos el par de claves con DSA. A continuación importaremos 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.

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.

Generamos las claves DSA con el comando indicado en el enunciado:

```
si2@si2srv01:~$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/si2/.ssh/id_dsa):
Created directory '/home/si2/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/si2/.ssh/id_dsa.
Your public key has been saved in /home/si2/.ssh/id_dsa.pub.
The key fingerprint is:
90:5a:f6:41:62:ce:95:ce:ac:95:71:93:36:95:f6:23 si2@si2srv01
The key's randomart image is:
+--[ DSA 1024 ]-----+
|  o o . o . . |
| + =o * o |
| B+. = + . |
| + o*. E o |
| . oS . . |
| . |
| |
| |
| |
+-----+
si2@si2srv01:~$
```

Copiamos la clave pública al servidor 2 y 3.

```
si2@si2srv01:~$ scp .ssh/id_dsa.pub si2@10.5.9.2:
The authenticity of host '10.5.9.2 (10.5.9.2)' can't be established.
RSA key fingerprint is d8:ba:48:3c:ad:db:fb:7a:86:e2:a5:f0:5e:3e:66:5e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.5.9.2' (RSA) to the list of known hosts.
si2@10.5.9.2's password:
id_dsa.pub                                100% 602      0.6KB/s   00:00
si2@si2srv01:~$ scp .ssh/id_dsa.pub si2@10.5.9.3:
The authenticity of host '10.5.9.3 (10.5.9.3)' can't be established.
RSA key fingerprint is d8:ba:48:3c:ad:db:fb:7a:86:e2:a5:f0:5e:3e:66:5e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.5.9.3' (RSA) to the list of known hosts.
si2@10.5.9.3's password:
id_dsa.pub                                100% 602      0.6KB/s   00:00
si2@si2srv01:~$
```

Creamos un nuevo directorio en el que copiaremos la clave.

```
si2@si2srv02:~$ mkdir -m 700 ~/.ssh
si2@si2srv02:~$ cat ~/id_dsa.pub >> ~/.ssh
.ssh/
.sudo_as_admin_successful
si2@si2srv02:~$ cat ~/id_dsa.pub >> ~/.ssh/authorized_keys2
si2@si2srv03:~$ mkdir -m 700 ~/.ssh
si2@si2srv03:~$ cat ~/id_dsa.pub >> ~/.ssh/authorized_keys2
si2@si2srv03:~$ _
```

Cuando hacemos el comando “ssh -v [si2@10.X.Y.2](#) y ssh -v [si2@10.X.Y.3](#)” aparece lo siguiente por pantalla:

```
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.

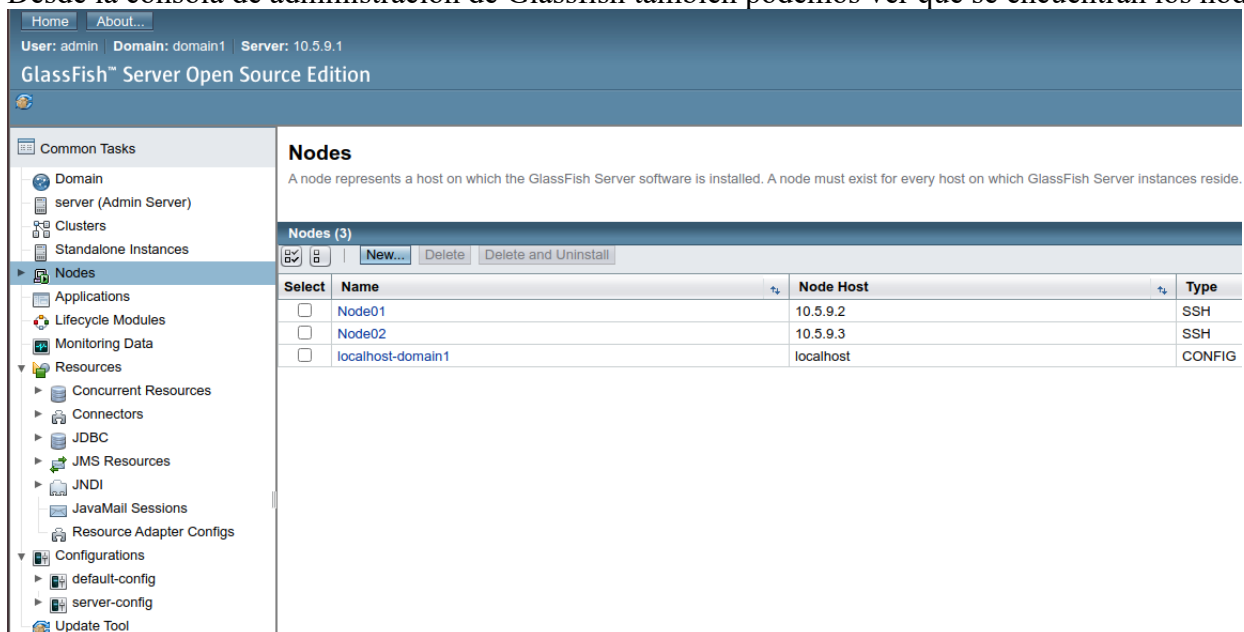
Last login: Wed May  5 02:45:24 2021 from 10.5.9.1
Loading es
si2@si2srv02:~$ _
```

Ejercicio 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) correspondientes² están efectivamente corriendo en cada una de las dos máquinas virtuales. Adjunte evidencias a la memoria de la práctica.

Comandos de creación de los nodos, listado de ellos y ping:

```
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 --user admin --passwordfile /opt/SI2/passwordfile create-node-ssh --sshuser si2 --nodehost 10.5.9.2 --nodedir /opt/glassfish4 Node01
Command create-node-ssh executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create-node-ssh --sshuser si2 --nodehost 10.5.9.3 --nodedir /opt/glassfish4 Node02
Command create-node-ssh executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile list-nodes
localhost-domain1 CONFIG localhost
Node01 SSH 10.5.9.2
Node02 SSH 10.5.9.3
Command list-nodes executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile ping-node-ssh Node01
Successfully made SSH connection to node Node01 (10.5.9.2)
Command ping-node-ssh executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile ping-node-ssh Node02
Successfully made SSH connection to node Node02 (10.5.9.3)
Command ping-node-ssh executed successfully.
```

Desde la consola de administración de Glassfish también podemos ver que se encuentran los nodos:



The screenshot shows the GlassFish Server Open Source Edition administration console. The 'Nodes' page is active, displaying a table of nodes. The table has columns for 'Select', 'Name', 'Node Host', and 'Type'. There are three nodes listed: Node01 (SSH, 10.5.9.2), Node02 (SSH, 10.5.9.3), and localhost-domain1 (CONFIG, localhost). The left sidebar shows the navigation menu with 'Nodes' selected.

Select	Name	Node Host	Type
<input type="checkbox"/>	Node01	10.5.9.2	SSH
<input type="checkbox"/>	Node02	10.5.9.3	SSH
<input type="checkbox"/>	localhost-domain1	localhost	CONFIG

Creamos el clúster y lo listamos:

```
si2@si2srv01:~$ export AS_ADMIN_USER=admin
si2@si2srv01:~$ export AS_ADMIN_PASSWORDFILE=/opt/SI2/passwordfile
si2@si2srv01:~$ asadmin create-cluster SI2Cluster
Command create-cluster executed successfully.
si2@si2srv01:~$ asadmin list-clusters
SI2Cluster not running
Command list-clusters executed successfully.
```

Los archivos hosts de las 3 máquinas virtuales se encuentran así:

```
si2@si2srv01:~$ cat /etc/hosts
10.5.9.1 si2srv01
10.5.9.2 si2srv02
10.5.9.3 si2srv03
10.5.9.4 si2srv04
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
```

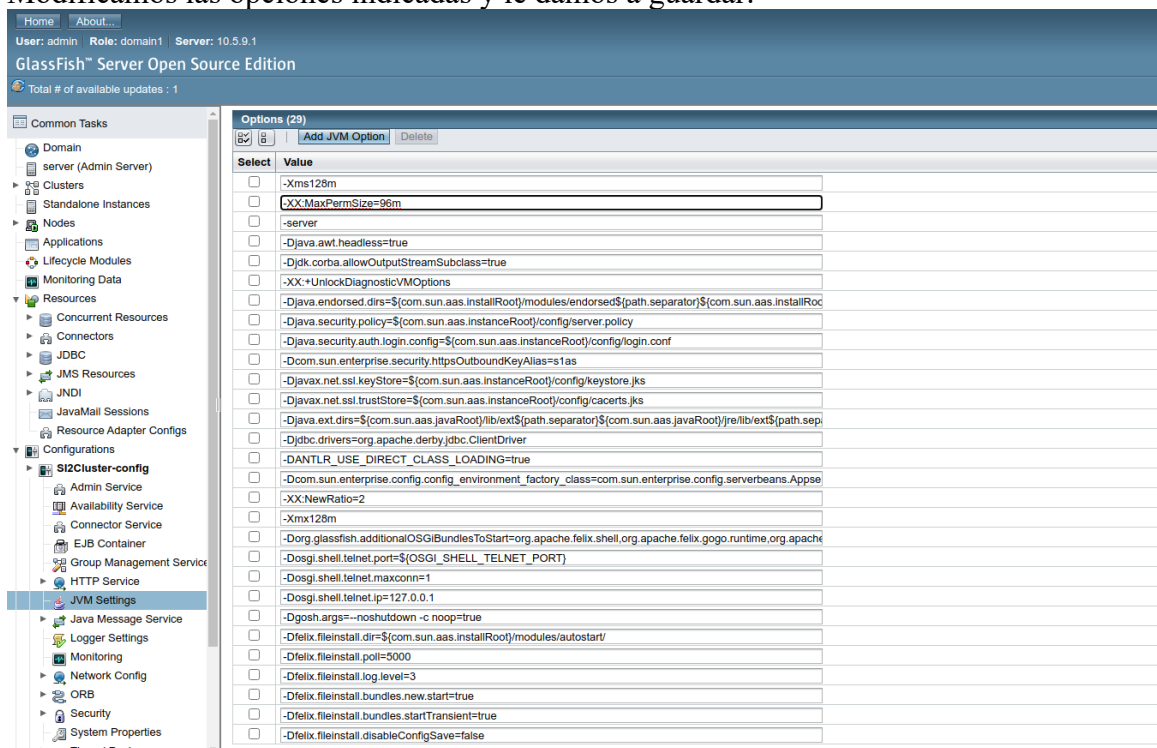
Creamos las 2 instancias que se nos piden:

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create-instance --cluster SI2Cluster --node Node01 Instance01
Command create-instance-filesystem 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 10.5.9.2
Command create-instance executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile create-instance --cluster SI2Cluster --node Node02 Instance02
Command create-instance-filesystem 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 10.5.9.3
Command create-instance executed successfully.
si2@si2srv01:~$
```

Listamos las instancias del cluster y lo ejecutamos:

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile list-instances -l
Name          Host      Port  Pid Cluster      State
Instance01    10.5.9.2  24848  --  SI2Cluster    not running
Instance02    10.5.9.3  24848  --  SI2Cluster    not running
Command list-instances executed successfully.
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile start-cluster SI2Cluster
Command start-cluster executed successfully.
```

Modificamos las opciones indicadas y le damos a guardar:



The screenshot shows the GlassFish Server Open Source Edition configuration interface. The left sidebar displays a tree view of the configuration hierarchy, with 'SI2Cluster-config' selected. The main panel shows the 'JVM Settings' tab, which contains a list of JVM options. The 'XX:MaxPermSize=96m' option is highlighted. Other options visible include '-Xms128m', '-server', '-Djava.awt.headless=true', '-Djdk.corba.allowOutputStreamSubclass=true', '-XX:+UnlockDiagnosticVMOptions', and various system and security properties.

Y reiniciamos el cluster:

```
si2@si2srv01:~$ asadmin stop-cluster SI2Cluster
Command stop-cluster executed successfully.
si2@si2srv01:~$ asadmin start-cluster SI2Cluster
Command start-cluster executed successfully.
si2@si2srv01:~$
```

Ejercicio 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 ha 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).

Añadimos instancia e ip a la tabla pago:

```
CREATE TABLE pago
(
-- idAutorizacion se autogenera con cada inserción
idAutorizacion serial not null,
idTransaccion char(16) not null,
codRespuesta char(3) not null default '000',
importe float not null,
idComercio char(16) not null,
numeroTarjeta char(19) not null references tarjeta,
fecha timestamp not null default current_timestamp,
instancia varchar(50) not null,
ip varchar(50) not null,
CONSTRAINT Pago_UC unique(idTransaccion, idComercio),
PRIMARY KEY (idAutorizacion)
);
```

Añadimos a PagoBean.java las siguientes líneas:

```
/* EJERCICIO 3 P3 */
private String instancia;
private String ip;
/**
 * @return la ip
 */
public String getIp() {
    return ip;
}

/**
 * @param ip String
 */
public void setIp(String ip) {
    this.ip = ip;
}

/**
 * @return la instancia
 */
public String getInstancia() {
    return instancia;
}

/**
 * @param instancia String
 */
public void setInstancia(String instancia) {
    this.instancia = instancia;
}
```

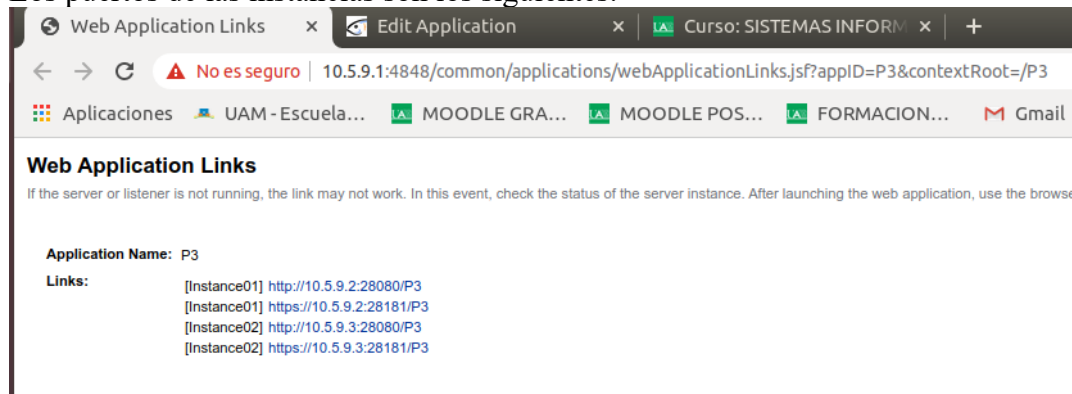
Modificación de ComienzaPago:

```
try {
    pago.setIp(java.net.InetAddress.getLocalHost().getHostAddress());
} catch (Exception e){
    pago.setIp("ERROR");
}
pago.setInstancia(System.getProperty("com.sun.aas.instanceName"));
```

Modificación de ProcesaPago:

```
try {
    pago.setIp(java.net.InetAddress.getLocalHost().getHostAddress());
} catch (Exception e){
    pago.setIp("ERROR");
}
pago.setInstancia(System.getProperty("com.sun.aas.instanceName"));
```

Los puertos de las instancias son los siguientes:



Pago con instancia01:



Pago con instancia02:



Podemos observar como la instancia y la ip aparecen en la tabla de pago:

Table Name	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	instancia	ip
pago	1	1	000	100	1	1111 2222 3333 4444	05/05/21 06:10	Instance01	10.5.9.2
tarjeta	2	2	000	200	1	4579 1165 7919 4384	05/05/21 06:11	Instance02	10.5.9.3

Ejercicio 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.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.

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

Name: JSESSIONID
Content: YYYYYYYYYYYYYYYYYYYY
Domain: 10.X.Y.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.X.Y.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: ZZZZZZZZZZZZZZZZZZZZZZ
Domain: 10.X.Y.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.

1- Eliminar todas las cookies del navegador

Borrar datos de navegación

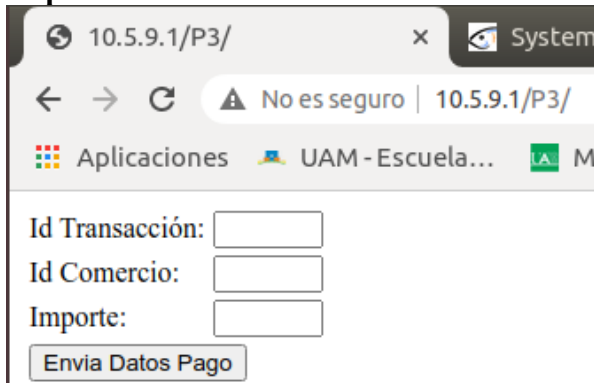
Básico Configuración avanzada

Intervalo de tiempo Desde siempre

- ☒ Historial de navegación
142 elementos
- ☒ Historial de descargas
33 elementos
- ☒ Cookies y otros datos de sitios
De 99 sitios web
- ☒ Archivos e imágenes almacenados en caché
189 MB
- ☒ Contraseñas y otros datos de inicio de sesión
No hay
- ☒ Datos de Autocompletar formulario

Cancelar Borrar datos

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

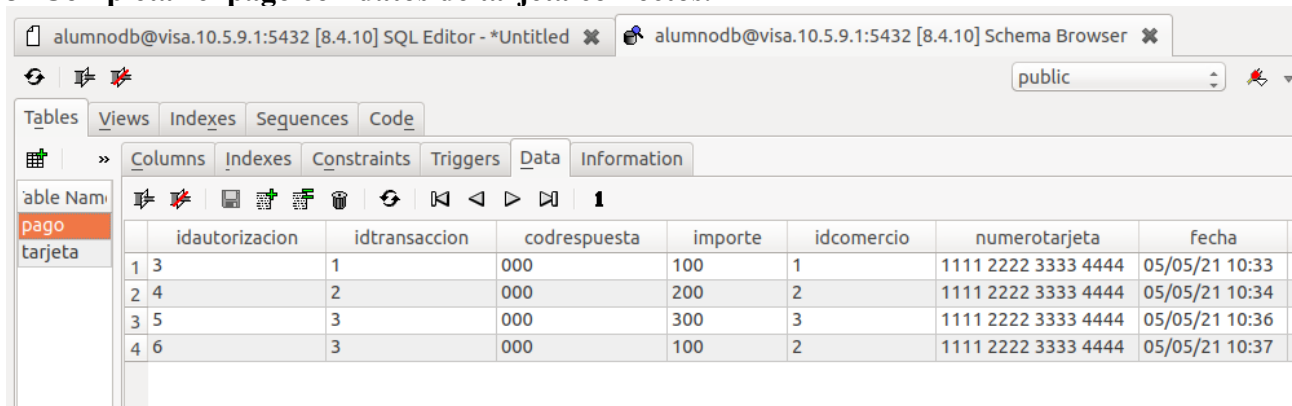


Id Transacción:

Id Comercio:

Importe:

3- Completar el pago con datos de tarjeta correctos.



	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha
1	3	1	000	100	1	1111 2222 3333 4444	05/05/21 10:33
2	4	2	000	200	2	1111 2222 3333 4444	05/05/21 10:34
3	5	3	000	300	3	1111 2222 3333 4444	05/05/21 10:36
4	6	3	000	100	2	1111 2222 3333 4444	05/05/21 10:37

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



Pago con tarjeta

Pago incorrecto

Prácticas de Sistemas Informáticos II

5- Mostrar la cookie “JSESSIONID” correspondiente a la URL del balanceador donde se vea:
Name: JSESSIONID
Content: YYYYYYYYYYYYYYYYYYYY
Domain: 10.X.Y.1
Path: /P3

← 10.5.9.1 ha almacenado datos de forma local Eliminar todo

JSESSIONID ^ X

Nombre
JSESSIONID

Contenido
d9b0681172efd593cdfa0b46daf8

Dominio
10.5.9.1

Ruta
/P3

Enviar para
Solo conexiones al mismo sitio web

Accesible para secuencia de comandos
No (HttpOnly)

Creada
miércoles, 5 de mayo de 2021, 19:37:20

Caduca
Al finalizar la sesión de navegación

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

System Properties Save Cancel

✓ New values successfully saved.

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"/>	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.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

Curso: SISTEMAS INFORM x 10.5.9.1/P3/ x +

← → ↺ No es seguro | 10.5.9.1/P3/

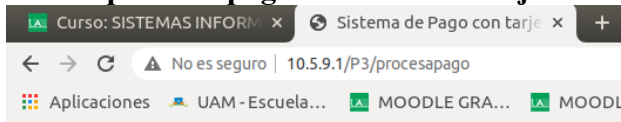
Aplicaciones UAM - Escuela... MOODLE GRA... MOOD

Id Transacción:

Id Comercio:

Importe:

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



Pago con tarjeta

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

```
idTransaccion: 4
idComercio: 4
importe: 444.0
codRespuesta: 000
idAutorizacion: 8
```

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

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

Name: JSESSIONID

Content: ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ

Domain: 10.X.Y.1

Path: /P3

Nombre	JSESSIONID
Contenido	dada410fea42300abdfd32eceece.Instance02
Dominio	10.5.9.1
Ruta	/P3
Enviar para	Solo conexiones al mismo sitio web
Accesible para secuencia de comandos	No (HttpOnly)
Creada	miércoles, 5 de mayo de 2021, 19:57:38
Caduca	Al finalizar la sesión de navegación

La diferencia entre ambas cookies es que cuando no se le indica el `jvmRoute`, no sabe la instancia del nodo que realiza la petición, de modo que en el primer caso, las peticiones se van balanceando en las distintas instancias pero en el segundo caso se le asigna una instancia para que procese todas las peticiones, de modo que balancea las peticiones a nivel de usuario.

Ejercicio 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.

Hemos realizado 12 pagos, de los cuales 7 se han realizado desde la instancia01 y 5 se han realizado desde la instancia02

Table Name	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	instancia	ip
1	22	1	000	111	1	1111 2222 3333 4444	06/05/21 03:00	Instance02	10.5.9.3
2	23	2	000	222	1	4579 1165 7919 4384	06/05/21 03:01	Instance01	10.5.9.2
3	24	3	000	434	1	1111 2222 3333 4444	06/05/21 03:02	Instance01	10.5.9.2
4	25	23	000	1223	2	1111 2222 3333 4444	06/05/21 03:05	Instance01	10.5.9.2
5	26	2312	000	3443	11	1111 2222 3333 4444	06/05/21 03:06	Instance01	10.5.9.2
6	27	653	000	4324	1	1111 2222 3333 4444	06/05/21 03:06	Instance01	10.5.9.2
7	28	4332	000	111	11	1111 2222 3333 4444	06/05/21 03:06	Instance01	10.5.9.2
8	29	23	000	111	1	1111 2222 3333 4444	06/05/21 03:11	Instance02	10.5.9.3
9	30	3211	000	222	1	1111 2222 3333 4444	06/05/21 03:11	Instance02	10.5.9.3
10	31	9998	000	1233	1	1111 2222 3333 4444	06/05/21 03:12	Instance02	10.5.9.3
11	32	654	000	1111	1	1111 2222 3333 4444	06/05/21 03:12	Instance01	10.5.9.2
12	33	6000	000	111	1	1111 2222 3333 4444	06/05/21 03:13	Instance02	10.5.9.3

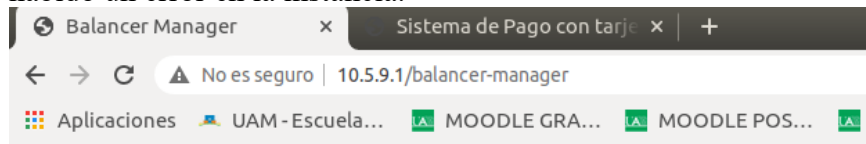
Ejercicio 6: Comprobación del proceso de *fail-over*. Parar la instancia del cluster que haya tenido menos elecciones hasta el momento. Para ello, identificaremos el *pid* (identificador del proceso java) de la instancia usando las herramientas descritas en esta práctica o el mandato 'ps -aef | grep 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.

```
si2@si2srv01:~$ ps -aef | grep java
si2 1845 1 1 02:42 tty1 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=192m -Xmx512m -client -javaagent:/opt/glassfish4/glassfish/lib/monitor/flashlight-agent.jar -Djavax.xml.accessExternalSchema=all -Djavax.net.ssl.trustStore=/opt/glassfish4/glassfish/domains/domain1/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.shell.remote,org.apache.felix.fileinstall -Dcom.sun.aas.installRoot=/opt/glassfish4/glassfish -Dfelix.fileinstall.poll=5000 -Djava.endorsed.dirs=/opt/glassfish4/glassfish/modules/endorsed:/opt/glassfish4/glassfish/lib/endorsed -Djava.security.policy=/opt/glassfish4/glassfish/domains/domain1/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=2 -Djavax.net.ssl.keyStore=/opt/glassfish4/glassfish/domains/domain1/config/keystore.jks -Djava.security.auth.login.config=/opt/glassfish4/glassfish/domains/domain1/config/login.conf -Dfelix.fileinstall.disableConfigSave=false -Dfelix.fileinstall.bundles.new.start=true -Dcom.sun.aas.instanceRoot=/opt/glassfish4/glassfish/domains/domain1 -Dosgi.shell.telnet.port=6666 -Dgosh.args=--nointeractive -Dcom.sun.enterprise.security.httpsOutboundKeyAlias=sias -Dosgi.shell.telnet.ip=127.0.0.1 -DANTLR_USE_DIRECT_CLASS_LOADING=true -Djava.awt.headless=true -Dcom.ctc.wstx.returnNullForDefaultNamespace=true -Djava.ext.dirs=/usr/lib/jvm/java-8-oracle/lib/ext:/usr/lib/jvm/java-8-oracle/jre/lib/ext:/opt/glassfish4/glassfish/domains/domain1/lib/ext -Djdbc.drivers=org.apache.derby.jdbc.ClientDriver -Djava.library.path=/opt/glassfish4/glassfish/lib:/usr/java/packages/lib/i386:/lib:/usr/lib -Dcom.sun.enterprise.glassfish.bootstrap.ASMain -upgrade false -domainindir /opt/glassfish4/glassfish/domains/domain1 -read-stdin true -asadmin-args --host,,localhost,--port,,4848,--secure=false,--terse=false,--echo=false,--interactive=true,--start-domain,--verbose=false,--watchdog=false,--debug=false,--domainindir,,/opt/glassfish4/glassfish/domains,,domain1 -domainname domain1 -instancename server -type DAS -verbose false -asadmin-classpath /opt/glassfish4/glassfish/lib/client/appserver-cli.jar -debug false -asadmin-classname com.sun.enterprise.admin.cli.AdminMain
si2 2350 2230 0 03:16 pts/0 00:00:00 grep java
si2@si2srv01:~$ kill -9 1845
```

Paramos la instancia con el siguiente comando:

```
si2@si2srv01:~$ asadmin stop-instance Instance01
The instance, Instance01, is stopped.
Command stop-instance executed successfully.
```

Después de parar la instancia01, podemos observar como el balanceador de carga nos indica que ha habido un error en la instancia.



Load Balancer Manager for 10.5.9.1

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://10.5.9.2:28080	Instance01		1	0	Err	23	14K	32K
http://10.5.9.3:28080	Instance02		1	0	Ok	60	48K	97K

Apache/2.2.14 (Ubuntu) Server at 10.5.9.1 Port 80

Ejercicio 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.**

Arrancamos la instancia con el siguiente comando:

```
si2@si2srv01:~$ asadmin start-instance Instance01
Waiting for Instance01 to start .....
Successfully started the instance: Instance01
instance Location: /opt/glassfish4/Node01/Instance01
Log File: /opt/glassfish4/Node01/Instance01/logs/server.log
Admin Port: 24848
Command start-local-instance executed successfully.
The instance, Instance01, was started on host 10.5.9.2
Command start-instance executed successfully.
```

Y comprobamos que la instancia vuelve a funcionar de manera normal:

