		Escuela Politécnica Superior Ingeniería Informática Prácticas de Sistemas Informáticos 2			
Grupo	2363	Práctica	3	Fecha	23/4/2019
Alumno/a	Román García Fernández				

Práctica 3: Seguridad y disponibilidad

Ejercicio 1

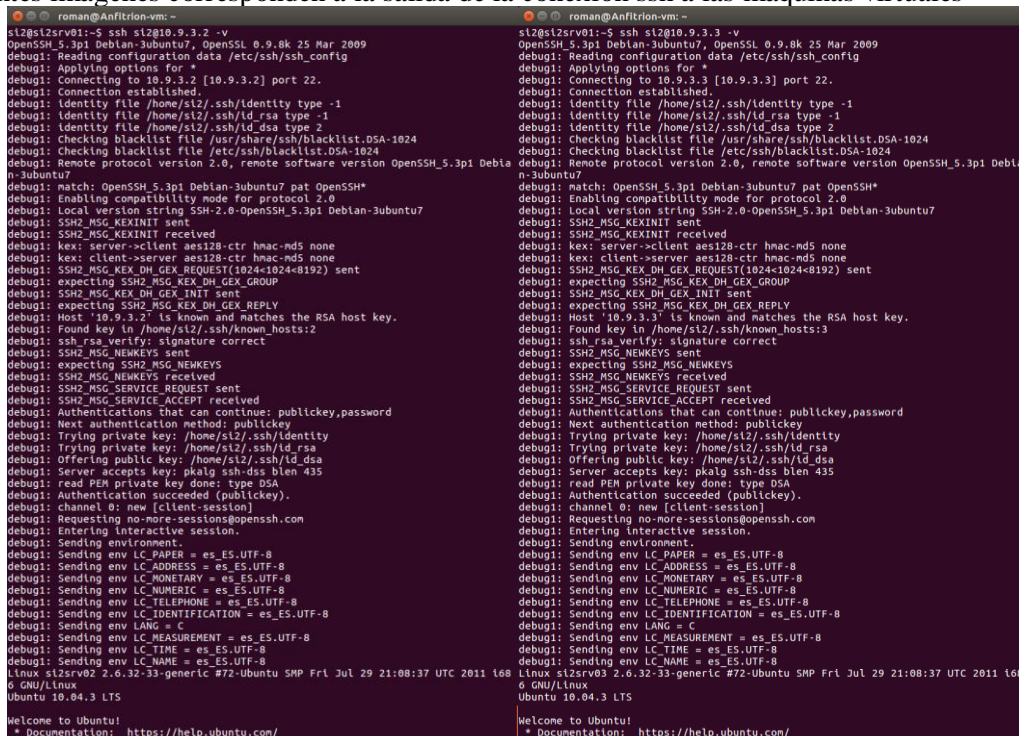
- Generamos claves DSA en todas las máquinas: `ssh-keygen -t dsa`
- Copiamos la clave pública en las máquinas virtuales: `scp .ssh/id_dsa.pub si2@10.9.3.x: (x = 1,2 y 3)`
 - Creamos el directorio `.ssh`: `mkdir -m 700 ~/.ssh`
 - Y guardamos la clave pública de cada máquina en el archivo `authorized_keys`: `cat (ruta a id_dsa.pub) >> ~/.ssh/authorized_keys`
 - Se puede usar `authorized_keys2` como nombre del archivo, pero su uso está obsoleto
 - Añadir la línea **PubkeyAcceptedKeyTypes +ssh-dss** al archivo `/etc/ssh/ssh_config` del anfitrión (solo si la distribución de Linux usada está actualizada), para que acepte expresamente cifrado con DSA: `echo 'PubkeyAcceptedKeyTypes +ssh-dss' >> /etc/ssh/ssh_config`
 - Esta configuración no hace falta en las máquinas de si2 facilitadas
 - Asegurarse de que las máquinas aceptan conexiones de protocolo 2. Para eso verificar que la línea correspondiente a protocolos del anterior fichero (`/etc/ssh/ssh_config`) está sin comentar

Aclaracion: En cada máquina tiene que haber un par de claves y un archivo `authorized_keys` que contenga las **claves públicas** de las otras máquinas a las que quieres que se conecte esta primera

Nota: Podemos evitar introducir el usuario y la contraseña de glassfish estableciendo las siguientes variables de entorno:

- `export AS_ADMIN_USER=admin`
- `export AS_ADMIN_PASSWORDFILE=~/.passwordfile`

Las siguientes imágenes corresponden a la salida de la conexión ssh a las máquinas virtuales



```

roman@Anfitrión-vm:~$ ssh si2@10.9.3.2 -v
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 10.9.3.2 [10.9.3.2] 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: Identity file /home/si2/.ssh/id_dsa type 2
debug1: Checking blacklist file /usr/share/ssh/blacklist.DSA-1024
debug1: Checking blacklist file /etc/ssh/blacklist.DSA-1024
debug1: Remote protocol version 2.0, remote software version OpenSSH_5.3p1 Debia
n-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(1024x1024x8192) 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 '10.9.3.2' 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: Trying private key: /home/si2/.ssh/id_rsa
debug1: Offering public key: /home/si2/.ssh/id_rsa
debug1: Server accepts key: pkalg ssh-dss blen 435
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 LC_ADDRESS = es_ES.UTF-8
debug1: Sending env LC_MONETARY = es_ES.UTF-8
debug1: Sending env LC_NUMERIC = es_ES.UTF-8
debug1: Sending env LC_TELEPHONE = es_ES.UTF-8
debug1: Sending env LC_IDENTIFICATION = es_ES.UTF-8
debug1: Sending env LC_LANG = C
debug1: Sending env LC_MEASUREMENT = es_ES.UTF-8
debug1: Sending env LC_TIME = es_ES.UTF-8
debug1: Sending env LC_NAME = es_ES.UTF-8
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/

```

Ejercicio 2

- Iniciamos glassfish en el primer nodo (si2srv01)
- Creamos los nodos ssh
 - (Node01): `asadmin --user admin --passwordfile ~/passwordfile create-node-ssh --sshuser si2 --nodehost 10.9.3.2 --nodedir /opt/glassfish4 Node01`
 - (Node02): `asadmin --user admin --passwordfile ~/passwordfile create-node-ssh --sshuser si2 --nodehost 10.9.3.3 --nodedir /opt/glassfish4 Node02`

En el administrador web (10.9.3.1:4848) podemos ver que se han creado los nodos, en la sección nodos

Nota: A partir de aquí he exportado las variables anteriormente comentadas, por lo que no hago uso de los parámetros de usuario y contraseña de asadmin

- Creamos el cluster: `asadmin create-cluster SI2Cluster`
- Listamos el cluster: `asadmin list-clusters`
- Verificamos que las ips de todas las máquinas tienen el nombre correspondiente en el fichero `/etc/hosts`:
 - 10.9.3.1 si2srv01
 - 10.9.3.2 si2srv02
 - 10.9.3.3 si2srv03
- Creamos dos instancias asociadas a los nodos:
 - `asadmin create-instance --cluster SI2Cluster --node Node01 Instance01`
 - `asadmin create-instance --cluster SI2Cluster --node Node02 Instance02`
- Listamos las instancias creadas: `asadmin list-instances -l`
- Iniciamos el cluster: `asadmin start-cluster SI2Cluster`
- Configuramos el cluster. En la consola de administración (10.9.3.1:4848), en la sección Configurations seleccionamos la configuración creada (SI2Cluster-config) y realizamos los siguientes cambios en JVM Settings -> JVM Options (un campo de texto):
 - Añadimos la opción `-server` y quitamos la opción `-client`. De esta forma permitimos multihilo
 - Añadimos `-Xms128m` que define la memoria mínima de los servidores a 128MB
 - Modificamos `-Xmx512m` a `-Xmx128m` para cambiar la memoria máxima de los servidores a 128MB
 - Modificamos `-XX:MaxPermSize=192m` a `-XX:MaxPermSize=96m` para bajar el tamaño del pool de objetos permanentes a 96MB
- Reiniciamos todas las instancias para que se aplique la configuración:
 - `asadmin stop-cluster SI2Cluster`
 - `asadmin start-cluster SI2Cluster`

Las siguientes imágenes corresponde a la ejecución del comando sugerido (`ps -aefl | grep java`) y la muestra del listado de instancias

The image shows two terminal windows from a user named 'roman' on a system named 'Anfiltrion-vm'.

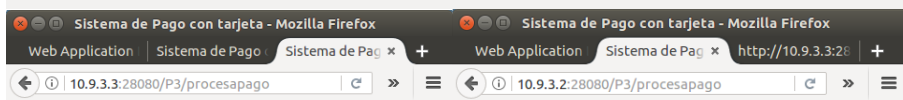
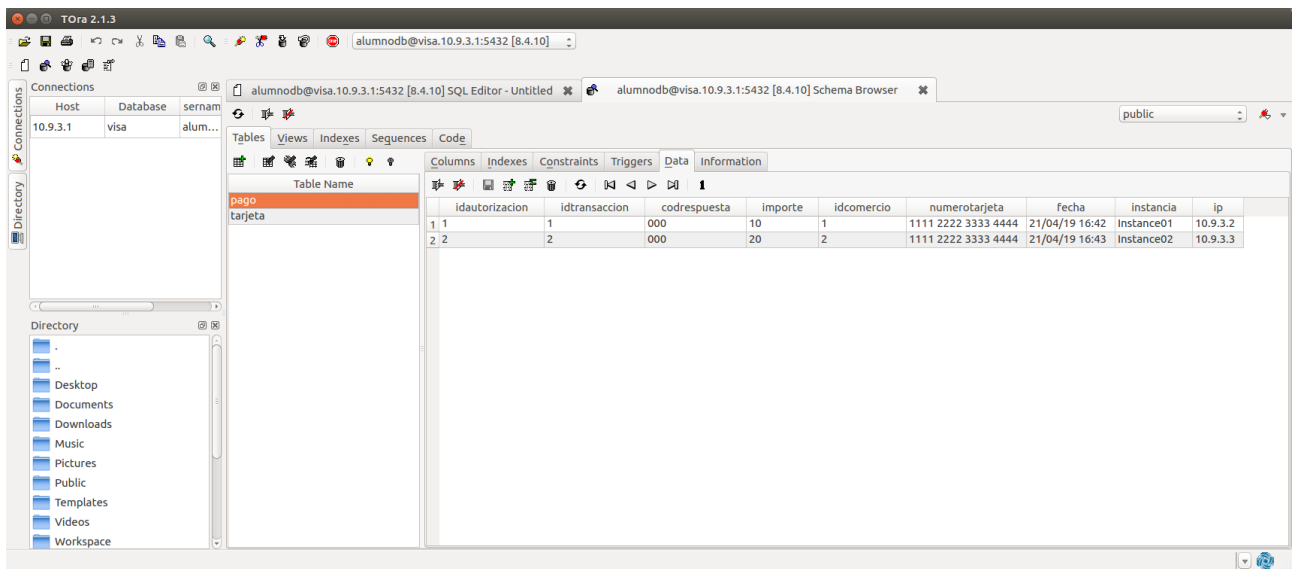
The top terminal window shows the command `asadmin list-instances -l` being executed. The output is a table with columns: Name, Host, Port, Pid, Cluster, and State. It lists two instances: Instance01 on host 10.9.3.2 at port 24848, and Instance02 on host 10.9.3.3 at port 24848, both belonging to the SI2Cluster and currently in a 'not running' state. Below the table, it says 'Command list-instances executed successfully.'

The bottom terminal window shows the command `ps -aefl | grep java` being executed. The output is a long list of Java processes running on the system, including the Java Virtual Machine (JVM) processes for the GlassFish instances.

Ejercicio 3

- Copiamos P1-base de la práctica 1 en la carpeta raíz de la práctica 3 y le cambiamos el nombre a P3
- Sustituimos el archivo listado.csv de la ruta P3/datagen por el facilitado para esta práctica
- Copiar el archivo insert.sql facilitado, en la carpeta P3/sql
- En create.sql agregamos dos campos a la tabla pago
 instancia varchar(50),
 ip varchar(50),
- Añadimos estos campos a la clase de java PagoBean.java (P3/src/ssii2/visa)
 private String ip;
 private String instancia;
- Añadimos los setters y getters de estos atributos
- Añadimos la obtención de estos parámetros en la creación de un pago en ComienzaPago.java (P3/src/ssii2/controlador/ComienzaPago.java)
 pago.setIp(java.net.InetAddress.getLocalHost().getHostAddress());
 pago.setInstancia(System.getProperty("com.sun.aas.instanceName"));
 Nota: Hay que importar la librería UnknownHostException y hacer que la función de creación pueda lanzar una excepción de ese tipo:
 import java.net.UnknownHostException;
Lo hacemos también con el controlador Procesapago.java (P3/src/ssii2/controlador/ProcesaPago.java)
- En VisaDAO.java () usamos los campos "instancia" e "ip":
Dentro del atributo insertpagos_qry:
 - private static final String INSERT_PAGOS_QRY = "insert into pago(" + "idTransaccion,importe,idComercio,numeroTarjeta, instancia, ip)" + " values (?, ?, ?, ?, ?)";En el método getQryInsertPago
 - String qry = "insert into pago(" + "idTransaccion," + "importe,idComercio," + "numeroTarjeta)" + " values (" + "\"" + pago.getIdTransaccion() + "\", " + pago.getImporte() + ", " + "\"" + pago.getIdComercio() + "\", " + "\"" + pago.getTarjeta().getNumero() + "\"" + "\"" + pago.getInstancia() + "\"" + "\"" + pago.getIp() + "\"" + ")"; return qry;En realizaPago, en la composición de la consulta preparada
 - pstmt.setString(5, pago.getInstancia());
 - pstmt.setString(6, pago.getIp());En getPagos hacemos lo mismo que en ComienzaPago
 - p.setInstancia(rs.getString("instancia"));
 - p.setIp(rs.getString("ip"));
- Reemplazar el fichero postgresql.xml por el suministrado
- Modificar build.properties
 - nombre=P3
 - as.host=10.9.3.1
 - as.target=SI2Cluster
- Modificar postgresql.properties
 - db.host=10.9.3.1
 - db.client.host=10.9.3.1

A continuación, se muestran las imágenes correspondientes a los enlaces de la aplicación, los pagos con en ambas instancias a partir de su link y la comprobación en tora de su correcta ejecución en cada uno:



Pago con tarjeta

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

idTransaccion: 2
idComercio: 2
importe: 20.0
codRespuesta: 000
idAutorizacion: 2

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II

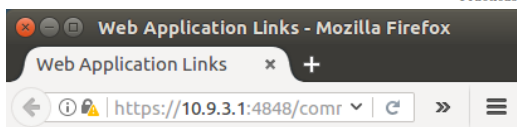
Pago con tarjeta

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

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

[Volver al comercio](#)

Prácticas de Sistemas Informáticos II



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://10.9.3.2:28080/P3>
[Instance01] <https://10.9.3.2:28181/P3>
[Instance02] <http://10.9.3.3:28080/P3>
[Instance02] <https://10.9.3.3:28181/P3>

[Close](#)

Ejercicio 4

- Creamos el archivo de configuración /etc/apache2/mods-available/proxy_balancer.conf **como root** y lo rellenamos con la siguiente información:

```
ProxyRequests Off
<Proxy balancer://SI2Cluster>
BalancerMember http://10.9.3.2:28080 route=Instance01
BalancerMember http://10.9.3.3:28080 route=Instance02
</Proxy>
<Location /P3>
Order allow,deny
Allow from all
ProxyPass balancer://SI2Cluster/P3 stickysession=JSESSIONID|jsessionid scolonpathdelim=On
ProxyPassReverse balancer://SI2Cluster/P3
</Location>
<Location /balancer-manager>
SetHandler balancer-manager
</Location>
```
- Entramos en la carpeta de mods en apache (/etc/apache2/mods-enabled) y creamos enlaces a los archivos del balanceador para que apache los pueda usar

```
sudo ln -sf ../mods-available/proxy.load
sudo ln -sf ../mods-available/proxy_http.load
sudo ln -sf ../mods-available/proxy_balancer.load
sudo ln -sf ../mods-available/proxy_balancer.conf
```
- Después reiniciamos el servicio apache: `sudo service apache2 restart`
- Agregamos una nueva propiedad en glassfish desde la consola de administración (Configurations -> SI2Cluster-config -> System Properties -> Add Property)
 Instance Variable Name: `jvmRoute`
 Default Value: `${com.sun.aas.instanceName}`
Pulsamos en save para guardar la modificación
- Reiniciamos el cluster

```
asadmin stop-cluster SI2Cluster
asadmin start-cluster SI2Cluster
```
- Diferencia entre usar la cookie o no usarla:

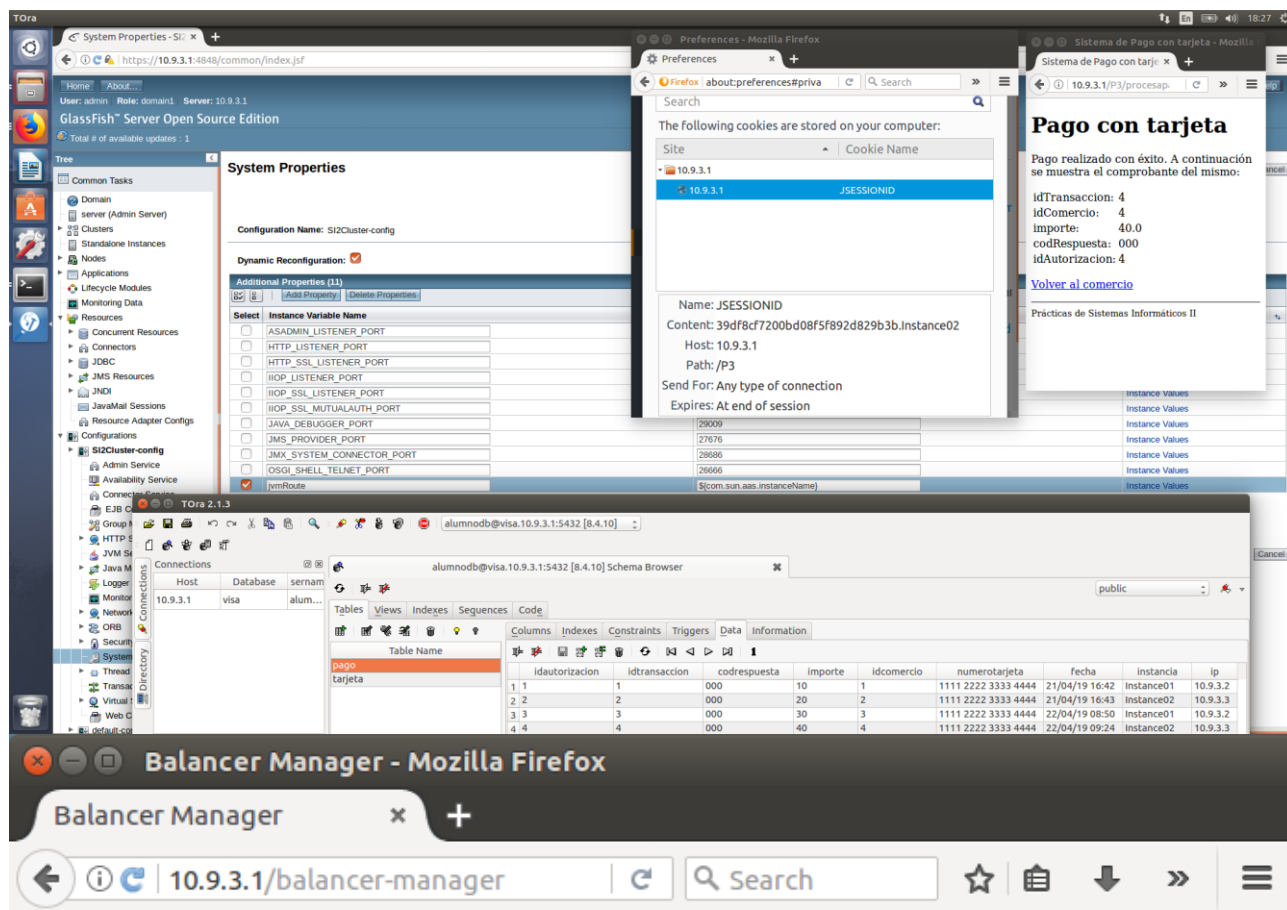
Sin usar la cookie

The screenshot shows the GlassFish administration console with the 'System Properties' configuration page for the 'SI2Cluster-config'. A message indicates 'New values successfully saved.' The 'Additional Properties (10)' table lists various instance variables and their default values.

Overlaid on the console is a Firefox 'Cookies' window showing cookies stored on the computer. The 'JSESSIONID' cookie is highlighted, showing its content as '5c9ba52a41a2803fd7bd06c816ab' and its path as '/'. Below the cookie window, a table displays transaction data:

	idautorizacion	idtransaccion	codrespuesta	importe	idcomercio	numerotarjeta	fecha	instancia	ip
1	1	000	10	1	1111 2222 3333 4444	21/04/19 16:42	Instance01	10.9.3.2	
2	2	000	20	2	1111 2222 3333 4444	21/04/19 16:43	Instance02	10.9.3.3	
3	3	000	30	3	1111 2222 3333 4444	22/04/19 08:50	Instance01	10.9.3.2	

Usando la cookie



Load Balancer Manager for 10.9.3.1

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

Balanceador

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://10.9.3.2:28080	Instance01		1	0	Ok	0	0	0
http://10.9.3.3:28080	Instance02		1	0	Ok	0	0	0

Apache/2.2.14 (Ubuntu) Server at 10.9.3.1 Port 80

El balanceador está configurado para usar esta cookie de forma que sabe a qué instancia enviar cada petición. Cuando este valor no es almacenado, el balanceador no sabe a qué instancia enviarla y devuelve un error, por lo que no se efectúa el pago.

En cuanto a la cuestión referente al valor de esta, no sería apropiado usar el valor hostName porque perderíamos la transparencia de ubicación. No obstante, con este valor seguiría funcionando.

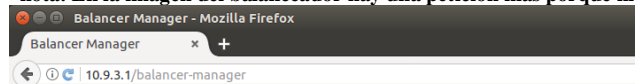
Ejercicio 5

Por razones horarias y de recursos, para este ejercicio y los siguientes, he usado una arquitectura basada en un solo PC físico. He mantenido la arquitectura de red y recursos de forma que la única diferencia sea la máquina física que ejecuta las virtuales.

En este ejercicio se pedía verificar que la cookie servía para mantener la sesión y comprobar que, dado un valor para esta variable, la instancia que atendía la petición era por defecto la misma que en las anteriores peticiones para el mismo cliente. Dado que en el momento de la realización de la práctica no tengo acceso a un segundo PC físico, adjunto un pantallazo de la maquina anfitriona en la que podemos observar cómo, para un mismo cliente, la instancia que lo atiende es la misma.

Para contrastar esta afirmación adjunto también otro pantallazo del número de peticiones atendidas por cada instancia después de 20 peticiones con Jmeter: En este caso, dado que las peticiones se realizan a una velocidad mayor, observamos que el balanceador alterna entre ambas instancias aun siendo el mismo cliente.

***nota: En la imagen del balanceador hay una petición más porque hice una prueba de conexión por el método web**



Load Balancer Manager for 10.9.3.1

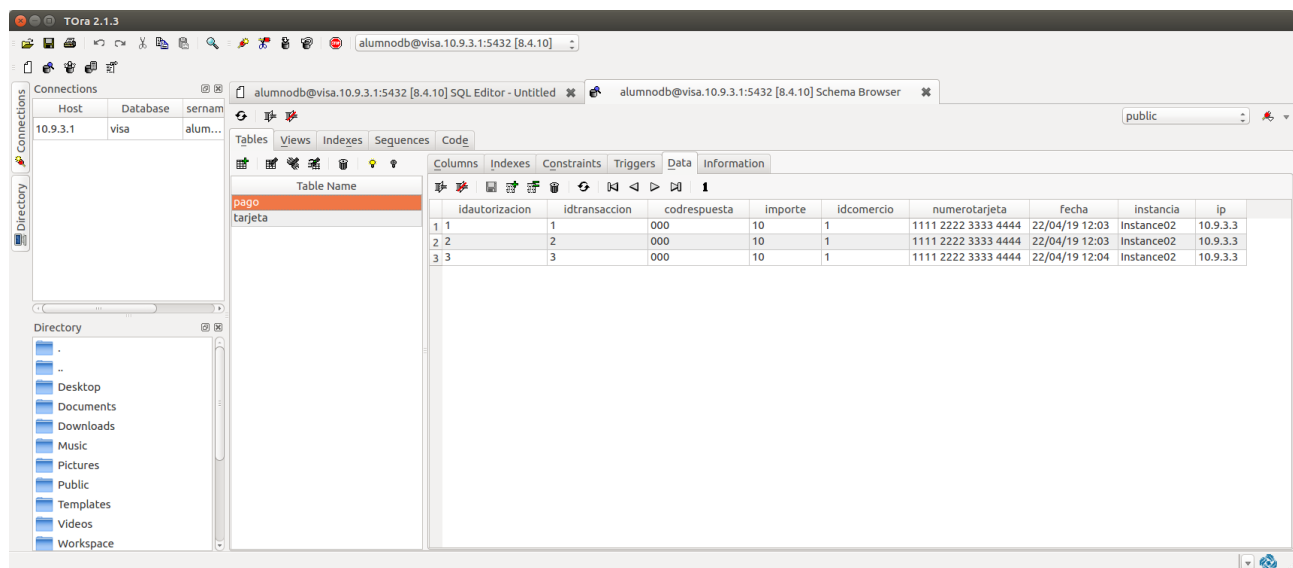
Server Version: Apache/2.2.14 (Ubuntu)
Server Built: Nov 3 2011 03:31:27

LoadBalancer Status for balancer://si2cluster

StickySession	Timeout	FailoverAttempts	Method
JSESSIONID jsessionid	0	1	byrequests

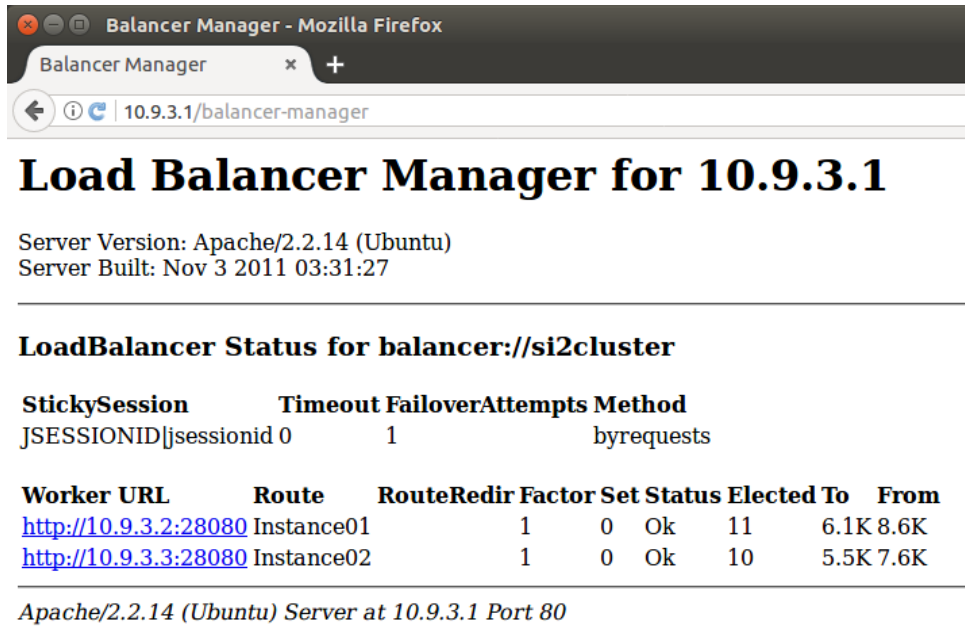
Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.9.3.2:28080	Instance01	1	0	Ok	11	6.1K	8.6K
http://10.9.3.3:28080	Instance02	1	0	Ok	10	5.5K	7.6K

Apache/2.2.14 (Ubuntu) Server at 10.9.3.1 Port 80



Ejercicio 6

Balanceador con las
dos instancias

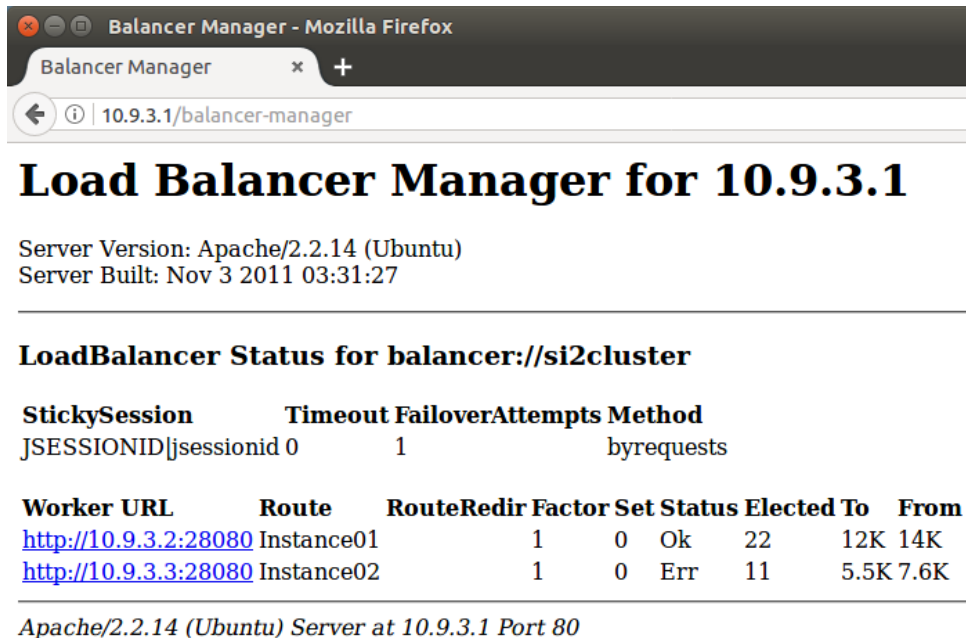


The screenshot shows the Balancer Manager web interface in a Mozilla Firefox browser. The title bar says "Balancer Manager - Mozilla Firefox". The address bar shows "10.9.3.1/balancer-manager". The main heading is "Load Balancer Manager for 10.9.3.1". Below it, the server information is displayed: "Server Version: Apache/2.2.14 (Ubuntu)" and "Server Built: Nov 3 2011 03:31:27". The section "LoadBalancer Status for balancer://si2cluster" contains a table with configuration details: "StickySession" is "JSESSIONID|jsessionid", "Timeout" is "0", "FailoverAttempts" is "1", and "Method" is "byrequests". Below this is a table of worker instances:

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.9.3.2:28080	Instance01		1	0	Ok	11	6.1K 8.6K
http://10.9.3.3:28080	Instance02		1	0	Ok	10	5.5K 7.6K

At the bottom, it says "Apache/2.2.14 (Ubuntu) Server at 10.9.3.1 Port 80".

Balanceador con
una instancia caída



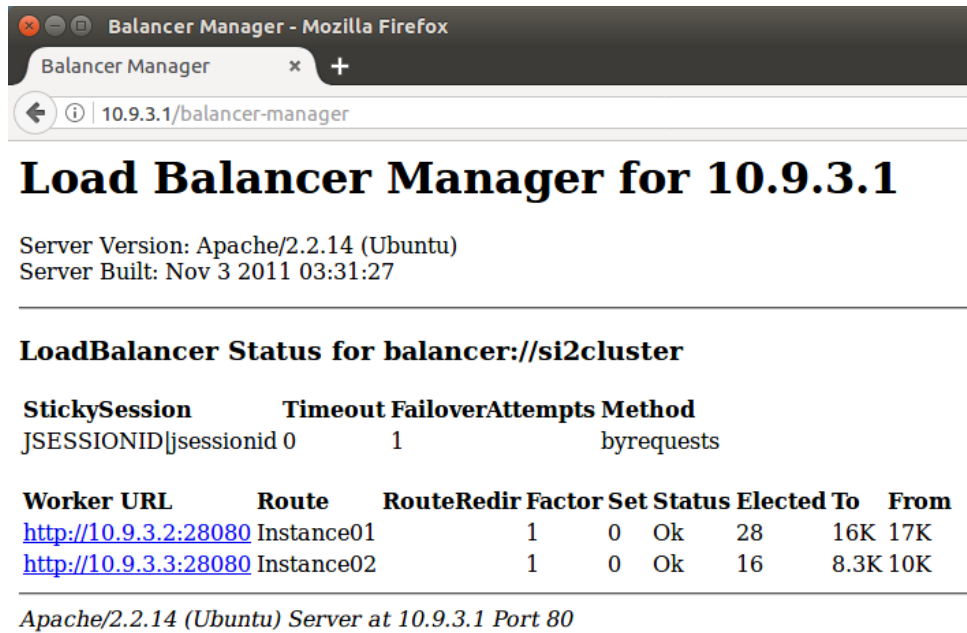
The screenshot shows the Balancer Manager web interface in a Mozilla Firefox browser. The title bar says "Balancer Manager - Mozilla Firefox". The address bar shows "10.9.3.1/balancer-manager". The main heading is "Load Balancer Manager for 10.9.3.1". Below it, the server information is displayed: "Server Version: Apache/2.2.14 (Ubuntu)" and "Server Built: Nov 3 2011 03:31:27". The section "LoadBalancer Status for balancer://si2cluster" contains a table with configuration details: "StickySession" is "JSESSIONID|jsessionid", "Timeout" is "0", "FailoverAttempts" is "1", and "Method" is "byrequests". Below this is a table of worker instances:

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.9.3.2:28080	Instance01		1	0	Ok	22	12K 14K
http://10.9.3.3:28080	Instance02		1	0	Err	11	5.5K 7.6K

At the bottom, it says "Apache/2.2.14 (Ubuntu) Server at 10.9.3.1 Port 80".

Ejercicio 7

Balanceador
con la
instancia
recuperada



Balancer Manager for 10.9.3.1

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

LoadBalancer Status for balancer://si2cluster

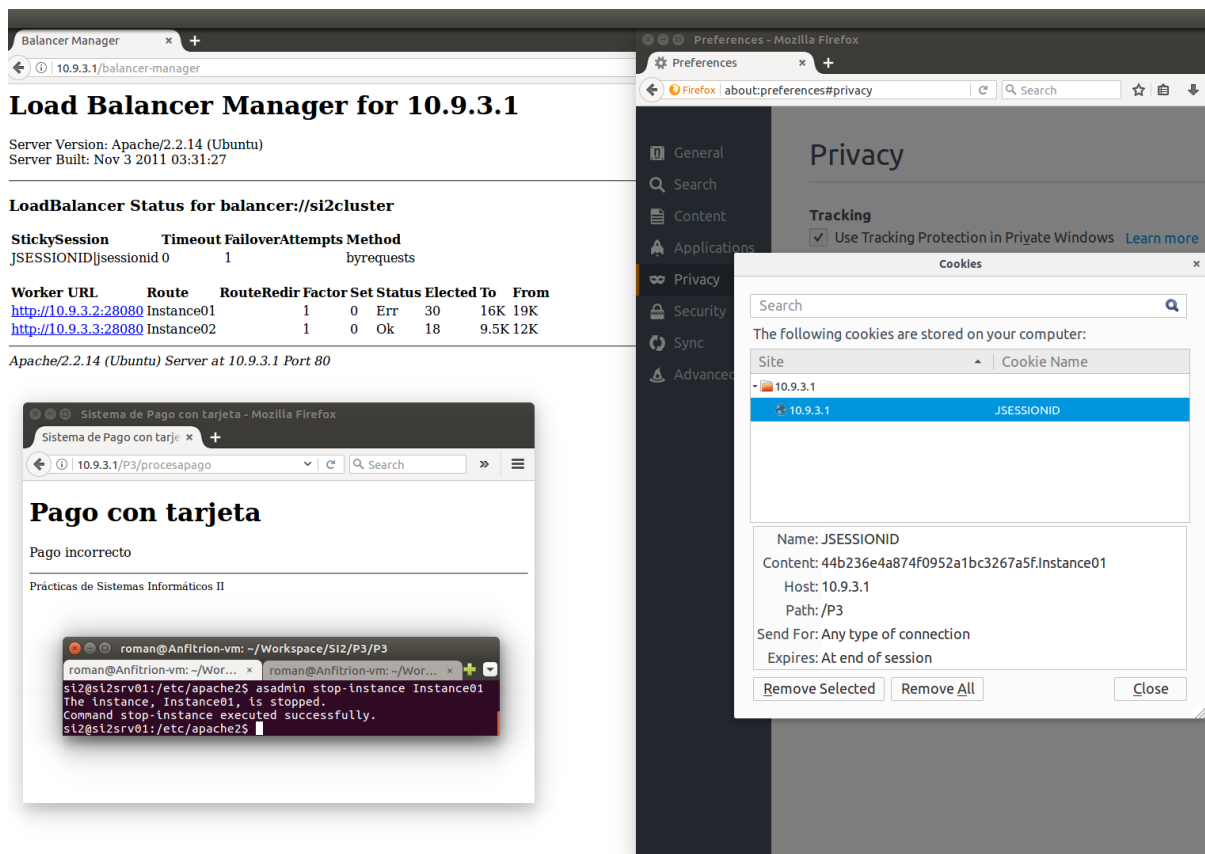
StickySession	Timeout	FailoverAttempts	Method
JSESSIONIDjsessionid 0	1		byrequests

Worker URL	Route	RouteRedir	Factor	Set	Status	Elected To	From
http://10.9.3.2:28080	Instance01		1	0	Ok	28	16K 17K
http://10.9.3.3:28080	Instance02		1	0	Ok	16	8.3K 10K

Apache/2.2.14 (Ubuntu) Server at 10.9.3.1 Port 80

Ejercicio 8

En este ejercicio he empezado un pago y, a mitad de pago, he parado la instancia. La siguiente imagen fue tomada justo después de parar la instancia. Como consecuencia vemos que la cookie se ha generado con el nombre de la instancia que ha atendido la petición y cómo, después de pararla, el pago no se ha efectuado correctamente. Además, se puede ver que el balanceador ha detectado la caída de la instancia y ha establecido su estado como Err



The composite image illustrates the state of the system during a payment process. It includes three main components:

- Balancer Manager for 10.9.3.1:** Shows the status of the load balancer. The 'Worker URL' table indicates that Instance01 has a status of 'Err' (Error) and Instance02 is 'Ok'. The 'StickySession' section shows 'JSESSIONIDjsessionid 0' with a timeout of 1 and failover attempts of 1.
- Pago con tarjeta:** A screenshot of a payment page showing a 'Pago incorrecto' (Incorrect payment) message. The page title is 'Sistema de Pago con tarjeta' and the URL is '10.9.3.1/P3/procesapago'.
- Terminal Screenshot:** A terminal window showing the command 'stop-instance Instance01' being executed successfully, indicating that the instance was stopped.

Ejercicio 9

Como podemos ver en la siguiente imagen, el balanceador reparte la carga equitativamente entre las instancias.

Balancer Manager

10.9.3.1/balancer-manager

Load Balancer Manager for 10.9.3.1

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://10.9.3.2:28080	Instance01		1	0	Ok	500	279K	514K
http://10.9.3.3:28080	Instance02		1	0	Ok	500	279K	514K

Apache/2.2.14 (Ubuntu) Server at 10.9.3.1 Port 80

Esto lleva a pensar que el algoritmo tras el balanceador es Round Robin, ya que va alternando cada vez una instancia diferente.