

Sistemas Informáticos 2

Práctica 3

Cristina Kasner Tourné Guillermo Guridi Mateos

23 de abril de 2015

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, Y ELIMINAR EL FICHERO

/etc/udev/rules.d/70-persistent-net.rules

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

Hemos realizado el ejercicio siguiendo los pasos que nos daban en el enunciado y no hemos tenido ningún problema.

Ejercicio2

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.

Adjuntamos la imagen con el resultado de ejecutar el comando `asadmin --user admin --passwordfile /opt/SI2/passwordfile list-instances {1`
Aquí se pueden observar los pids de cada una de las instancias.

```
si2@si2srv01:~$ asadmin --user admin --passwordfile /opt/SI2/passwordfile list-instances -l
Name      Host      Port  Pid   Cluster  State
Instance01 10.1.3.2  24848  2029  SI2Cluster  running
Instance02 10.1.3.3  24848  2123  SI2Cluster  running
Command list-instances executed successfully.
```

Figura 1: Lista de instancias

Ejercicio3

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).

Este ejercicio nos ha costado un poc más que los anteriores ya que hemos tenido un par de fallos trabajando con la base de datos.

Finalmente hemos conseguido solucionarlos y realizar los pagos.

Para encontrar el puerto de cada instancia hemos mirado en la consola de administración, tal y como nos decía el enenuciado. Los puertos son:

- Instancia01 → 28080
- Instancia02 → 28080

Adjuntamos las pantallas que confirma la realización de pago con cada una de las instancias y la base de datos resultante.

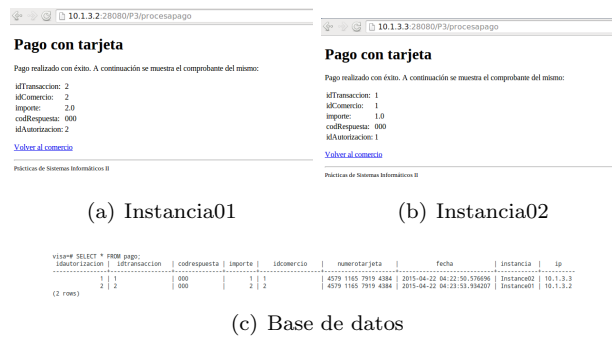


Figura 2: Pago instancias

Ejercicio4

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: ZZZZZZZZZZZZZZZZZZZZZ
Domain: 10.X.Y.1
Path: /P3
Mostrar las pantallas y comentar: las diferencias en el contenido de las cookie respecto a jvmRoute, cómo esta diferencia afecta a la afinidad y por qué.

Configuramos el balanceador de carga y comprobamos que todo se ha hecho correctamente:

Load Balancer Manager for 10.1.3.1

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://10.1.3.2:28080	Instance01		1	0	Ok	0	0	0
http://10.1.3.3:28080	Instance02		1	0	Ok	0	0	0

Apache/2.2.14 (Ubuntu) Server at 10.1.3.1 Port 80

Figura 3: Página de status del balanceador

Una vez hecho esto realizamos las actividades del ejercicio 4:

- Borramos las cookies:

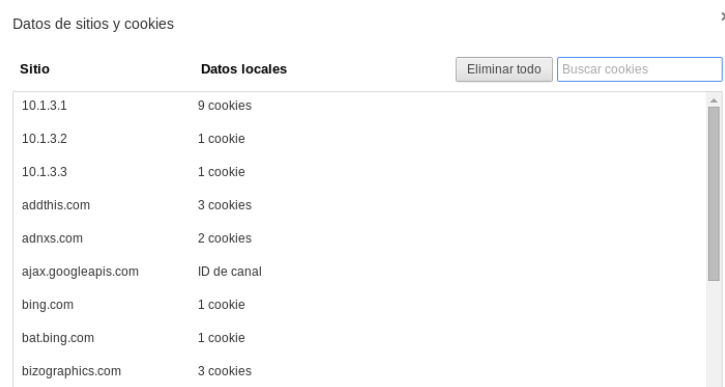


Figura 4: Cookies

Ejercicio5

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.

Ejercicio6

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`) 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.

Ejercicio7

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. **Consulte los apéndices para información detallada de comandos de gestión individual de las instancias.**

Ejercicio8

Fallo en el transcurso de una sesión.

- Desde un navegador, comenzar una petición de pago introduciendo los valores del mismo en la pantalla inicial y realizando la llamada al servlet `ComienzaPago`.
- Al presentarse la pantalla de “Pago con tarjeta”, leer la instancia del servidor que ha procesado la petición y detenerla. Se puede encontrar la instancia que ha procesado la petición revisando la cookie de sesión (tiene la instancia como sufijo), el `balancer-manager` o el `server.log` de cada instancia.
- Completar los datos de la tarjeta de modo que el pago fuera válido, y enviar la petición.
- Observar la instancia del cluster que procesa el pago, y razonar las causas por las que se rechaza la petición.

Ejercicio9

Modificar el script de pruebas JMeter desarrollado durante la P2. (P2.xml)
Habilitar un ciclo de 1000 pruebas en un solo hilo contra la IP del cluster y nueva URL de la aplicación: `http://10.X.Y.1/P3` Eliminar posibles pagos previos al ciclo de pruebas. Verificar el porcentaje de pagos realizados por cada instancia, así como (posibles) pagos correctos e incorrectos. ¿qué algoritmo de reparto parece haber seguido el balanceador? Comente todas sus conclusiones en la memoria de prácticas.

Apéncices