Sistemas Informáticos II
Práctica 3

Prácticas 2020/2021

El objetivo de esta práctica es mostrar la utilización de grupos de servidores *(clusters)* como herramienta para aumentar la disponibilidad de las aplicaciones distribuidas

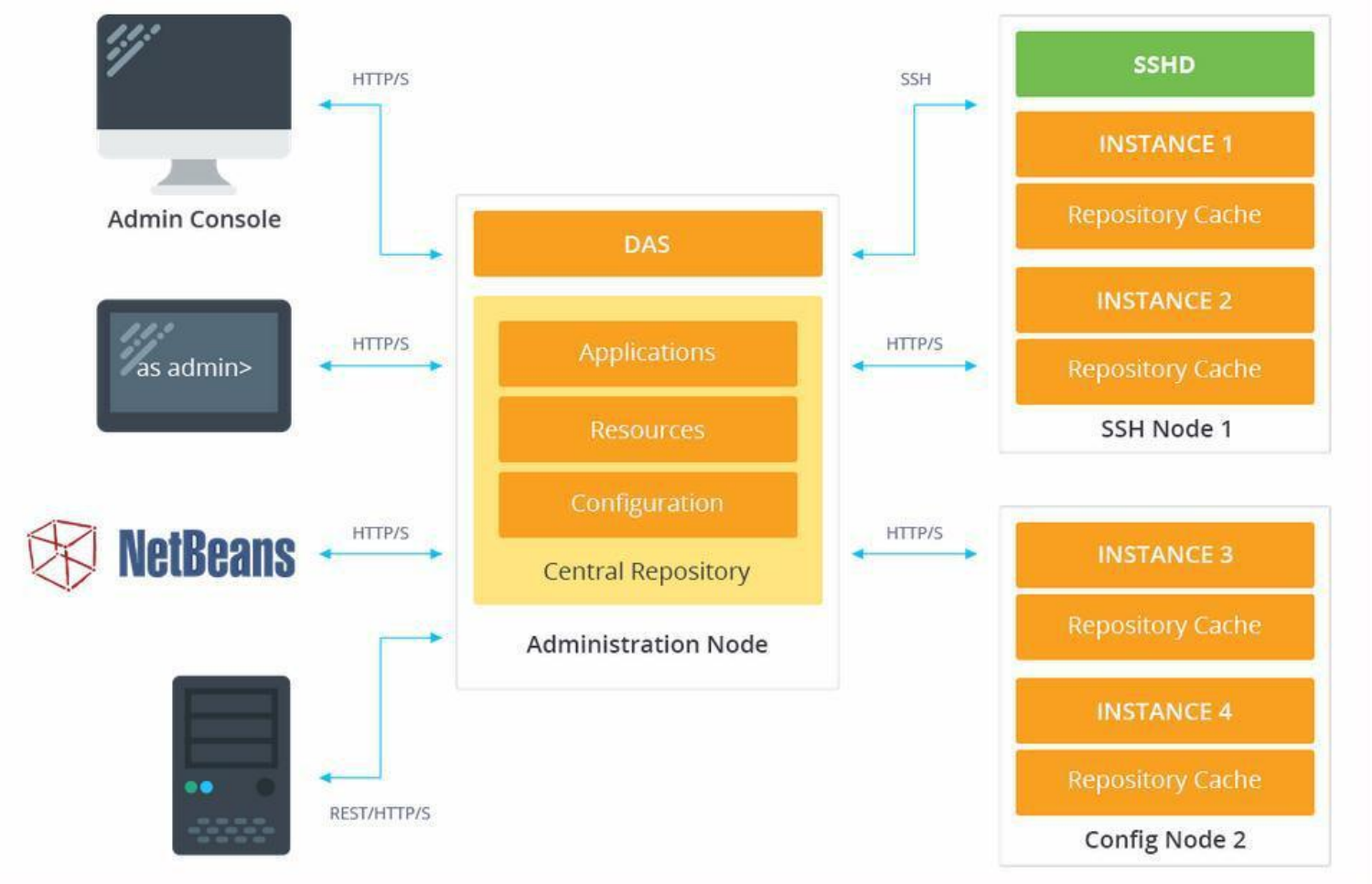
Para lograr este objetivo se realizarán una serie de pasos:

- Conceptos básicos de seguridad: cifrado asimétrico, claves DSA para automatización de comandos remotos SSH
- Aspectos generales de *clustering*.
- Configuración de un *cluster*.
- Pruebas de disponibilidad de una aplicación J2EE.

Definiciones

- **Instancia de servidor Glassfish**
  - Representa una JVM compatible con Java EE dentro de un nodo en el cual Glassfish se está ejecutando. Cada instancia pertenece a un solo dominio, y tiene su propia estructura de directorios, configuración y aplicaciones desplegadas.
- **Nodo**
  - Máquina en la cual está instalado el software Glassfish Server. Hay 2 tipos de nodos:
    - Nodos SSH: soportan comunicación remota por SSH. Permiten la administración centralizada desde el DAS
    - Nodos de configuración: No soportan comunicación remota, deben administrarse de forma local. Cada dominio contiene un nodo de configuración por defecto llamado *localhost-domain*.
- **Dominio**
  - Grupo de instancias que se administran de forma común. Cada dominio tiene su propia configuración, ficheros de log y áreas de despliegue.
- **Domain Administration Server (DAS)**
  - Instancia de servidor Glassfish con capacidades adicionales de administración de los recursos del dominio. También se le llama *admin server* o *default server*, ya que es la instancia que viene por defecto en cualquier instalación de Glassfish.

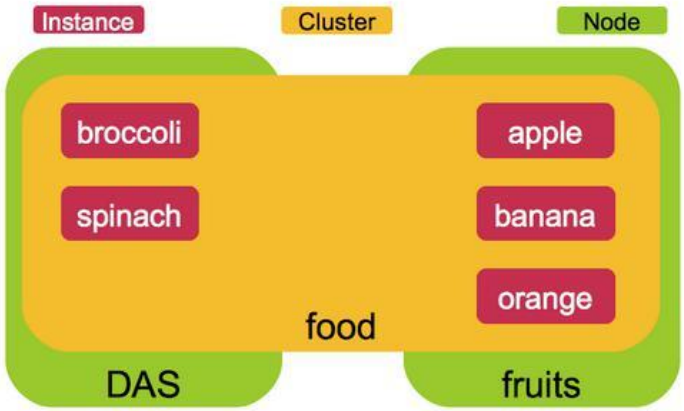
Definiciones



- *Fuente: https://jelastic.com/blog/how-to-configure-glassfish-cluster-with-automatic-load-balancing/*

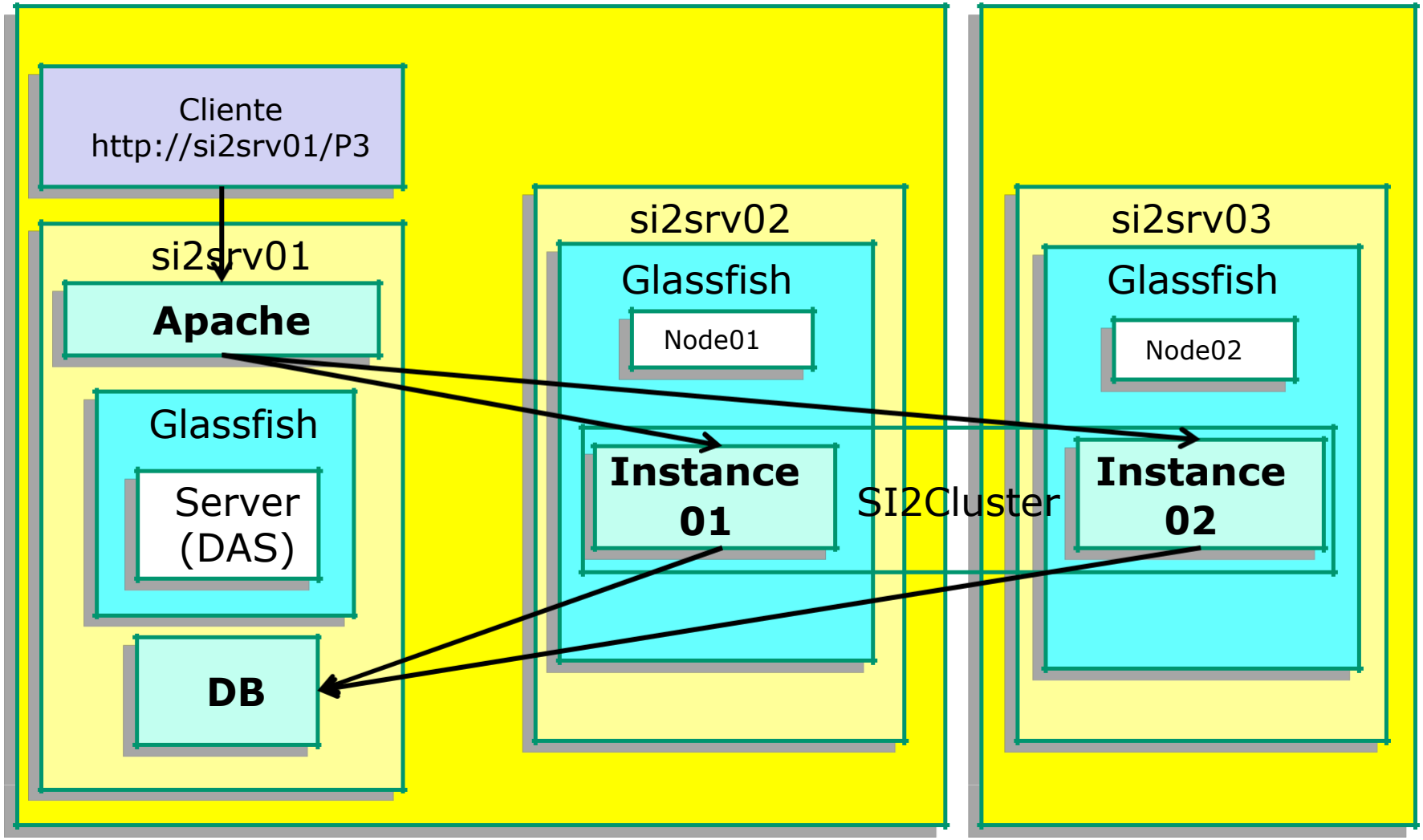
Glassfish Server Clusters

- Un **cluster** es un grupo de instancias de servidor Glassfish que trabajan juntas como una sola entidad lógica. Comparten una misma configuración, recursos y aplicaciones. Proporciona alta disponibilidad mediante:
  - Escalabilidad
  - Protección ante fallos
  - Balanceo de carga
- Las instancias de un cluster pueden residir en el mismo nodo o en nodos nodos distintos



- *Fuente: https://blogs.oracle.com/arungupta/totd-142:-glassfish-31-ssh-provisioning-and-startstop-instancecluster-on-localremote-machines*

En cuanto a la disponibilidad, la práctica consiste en la preparación de un *cluster* con dos instancias de servidores que reciben peticiones a través de un balanceador de carga, sobre los que se desplegará y probará una aplicación

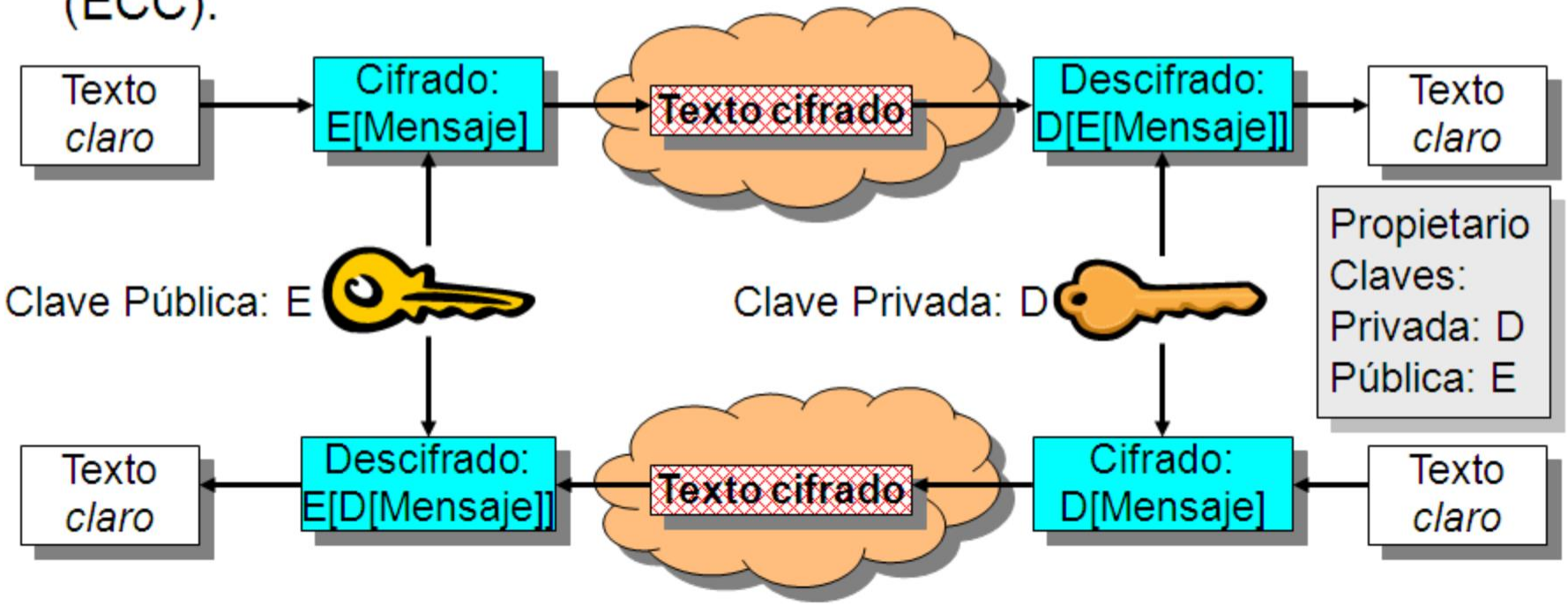


Los pasos a seguir para realizar la configuración del *cluster* se proporcionan en detalle en el enunciado de la práctica. En líneas generales, hay que realizar las siguientes tareas:

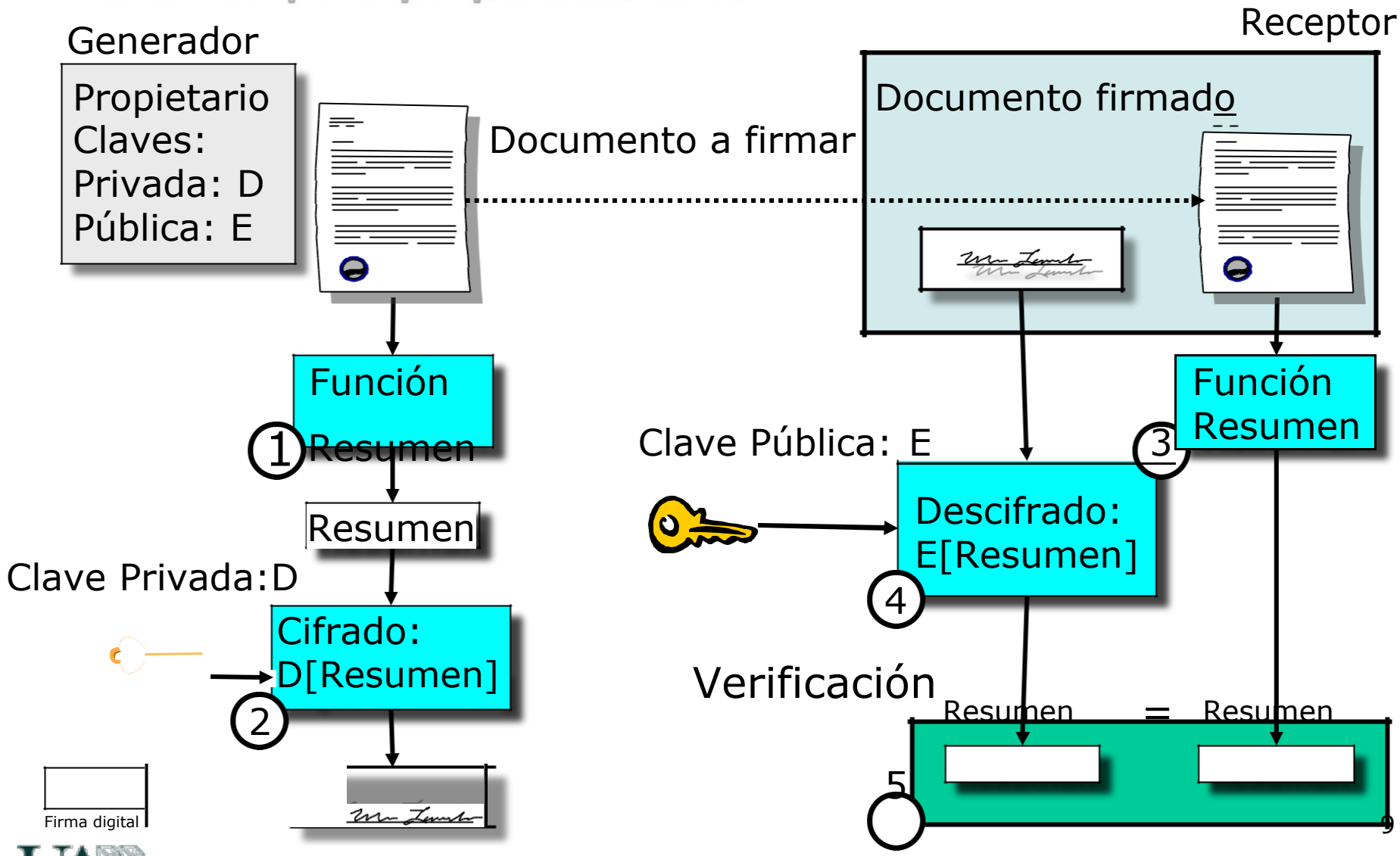
- Preparar la infraestructura de máquinas virtuales sobre PCs físicos de los laboratorios.
- Configurar la autenticación automática (SSH) de si2srv01 => si2srv02, si2srv03
- Creación de los nodos SSH.
- Creación del *cluster* y de las instancias de servidor que lo componen.
- Modificar la configuración de las instancias del *cluster* para adaptarlas a los requisitos de la práctica.
- Configurar un servidor Apache como balanceador de carga para las instancias creadas del *cluster*.

Gestión del cluster con SSH (DSA) - Cifrado asimétrico

- Dos claves. El propietario publica una de ellas, y mantiene en secreto la otra.
- Lo que se cifre con una de ellas sólo se puede descifrar con la otra, y viceversa.
- Algoritmos más empleados: RSA, ElGamal, Curvas Elípticas (ECC).



Gestión del cluster con SSH (DSA) – Firmas digitales
Permiten verificar la autenticidad e integridad de un documento y evitar su repudio por parte del autor.



Para detectar mejor los nodos del cluster que procesan las peticiones se realizarán las siguientes modificaciones sobre el código de la práctica 1, creando así la aplicación que denominaremos P3:

- Modificar la tabla *pago*: para que contenga dos nuevas columnas: *instancia* e *ip.* > create.sql
- Modificar el bean *Pago* para que pueda incorporar las dos nuevas propiedades anteriores. > Incluir nuevas propiedades en PagoBean, y métodos get/set.
- Modificar los *servlet ComienzaPago y ProcesaPago* para que a la hora de crear el bean Pago > Método *creaPago* de *ComienzaPago*
  - Instancia: `System.getProperty("com.sun.aas.instanceName")`
  - IP: `java.net.InetAddress.getLocalHost().getHostAddress()`
- Modificar *VisaDAO* para que a la hora de insertar el pago se incluyan los dos parámetros anteriores en la tabla *pago* > *VisaDAO.java*

También es necesario realizar modificaciones en los archivos de configuración de ant para lograr el despliegue sobre el *cluster* creado:

Archivo `build.properties`:

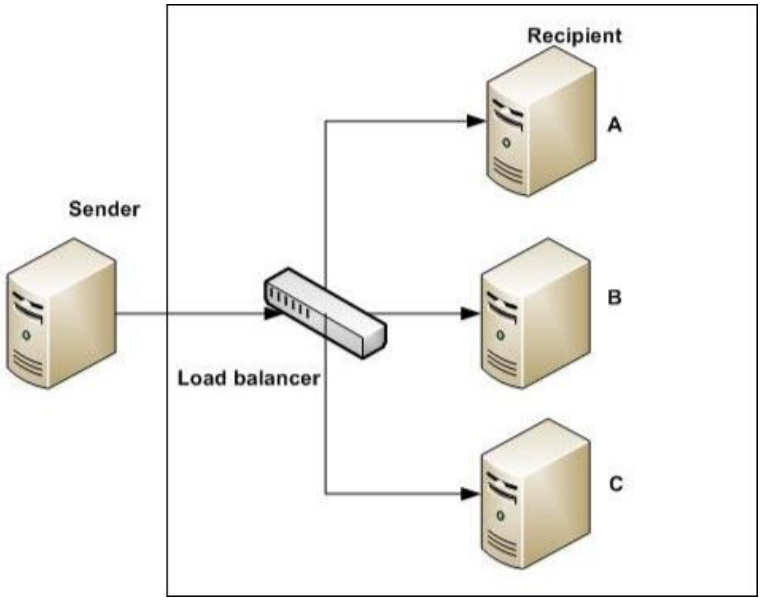
Original	Nuevo
... nombre=P1-base ... as.host=localhost ... as.target=server ...	nombre=**P3** ... as.host=**10.X.Y.1** ... as.target=**SI2Cluster** ...

Archivo `postgresql.properties`:

Original	Nuevo
... db.client.host=localhost db.host=localhost ...	db.client.host=**10.X.Y.1** Db.host=**10.X.Y.1** ...

Balanceo de carga

- Ofrece transparencia ante el cluster, protección ante fallos, aumento del rendimiento
- Problema: Cómo resolvemos el problema de que 2 instancias distintas procesen peticiones de una misma sesión?
  - Replicación de memoria (costoso)
  - **Afinidad de sesión**



- *Fuente: https://phptechsolutions.wordpress.com/2012/07/28/load-balancing-with-apache/*

Uso de jvmRoute para afinidad de sesión:

Propiedad del cluster "jvmRoute" hace que cada una de las instancias del *cluster* modifique la generación del identificador de sesión para incluir en él información de la instancia que ha procesado la petición.

• *El identificador de la instancia que ha procesado la petición se incluye en la cookie de sesión de J2EE "**JSESSIONID**".*

La práctica incluye una prueba de concepto que requiere acceder al gestor de cookies del navegador utilizado (para ver y borrar cookies del navegador):

Firefox:

"Preferencias" -> "Privacidad" -> "Mostrar cookies…"

Chrome:

"Configuración" -> "Mostrar opciones avanzadas...". En "Privacidad"->"Configuración de contenido....". En la sección "Cookies" -> "Todas las cookies y los datos de sitios"

Apache mod_proxy_balancer

- Módulo de Apache Server dedicado a configurar balanceadores de carga simples.
- Se encuentra instalado en las máquinas virtuales
- Fichero de configuración */etc/apache2/mods-available/proxy_balancer.conf* con el siguiente contenido:

A	`ProxyRequests Off`

B	`<Proxy balancer://SI2Cluster>`
	`  BalancerMember http://10.x.y.2:XXXXX route=Instance01`
	`  BalancerMember http://10.x.y.3:XXXXX route=Instance02`
	`</Proxy>`

C	`<Location /P3>`
	`  Order allow,deny`
	`  Allow from all`
	`  ProxyPass balancer://SI2Cluster/P3 stickysession=JSESSIONID|jsessionid scolonpathdelim=On`
	`  ProxyPassReverse balancer://SI2Cluster/P3`
	`</Location>`

D	`<Location /balancer-manager>`
	`  SetHandler balancer-manager`
	`</Location>`

Las pruebas a realizar con la nueva aplicación buscan validar el funcionamiento del *cluster* y hacer un breve análisis sobre la mejora en el rendimiento que produce su uso frente a los resultados obtenidos en la prueba de rendimiento.

- Prueba del correcto balanceo de carga
- Prueba del proceso de *fail-over*.
- Prueba del proceso de *fail-back*.
- Prueba de fallo de un nodo del *cluster* en el transcurso de una sesión.
- Prueba masiva (Jmeter) contra el cluster, para identificar el algoritmo de balanceo.
- Análisis de resultados y conclusiones.

La entrega de la práctica incluye todos los archivos necesarios para reproducir la prueba realizada por los alumnos.

- Código completo de la aplicación P3, preparado para el despliegue sobre el *cluster* SI2Cluster, sin incluir los directorios build y dist.
- Ficheros de configuración de Glassfish, domain.xml, de los tres servidores:
  DAS: /opt/glassfish4/glassfish/domains/domain1/config/domain.xml
  Instance01: /opt/glassfish4/Node01/Instance01/config/domain.xml
  Instance02: /opt/glassfish4/Node02/Instance02/config/domain.xml
- Fichero de configuración del balanceador
  /etc/apache2/mods-available/proxy_balancer.conf
- Memoria de la ejecución de la práctica.
  - **Detallar respuestas y evidencias a todas las cuestiones planteadas.**
- La entrega se realizará la semana del **10 al 13 de Mayo de 2021** antes del comienzo del examen de prácticas de la asignatura.