CygnusCloud – Esbozo de la arquitectura

# OpenStack y OpenNebula

Descartamos usar OpenNebula y OpenStack: son soluciones demasiado grandes, y entender cómo funcionan nos llevaría un mínimo de dos meses. Por eso, plantearemos e implementaremos nuestra propia arquitectura.

Se trata de una arquitectura bastante simple, que proporciona una base lo suficientemente sólida como para desarrollar todo el proyecto.

# Modelo básico

En nuestro sistema, podremos distinguir tres clases de máquinas:

* **Clientes**. Son las máquinas que usarán los alumnos para trabajar con las máquinas virtuales.
* **Servidor de máquinas virtuales**. En estas máquinas residirán las imágenes de disco de las distintas máquinas virtuales. Además, las máquinas virtuales también se ejecutarán en ellas.
* **Servidor central**. Esta máquina se encargará de la autenticación, de la asignación de un servidor de máquinas virtuales a cada usuario y de permitir la administración del sistema.

Es importante notar que el tráfico del servidor VNC no pasará en ningún momento a través del servidor central: este derivará las peticiones de cada usuario a un servidor de máquinas virtuales, que será quien se comunique con la máquina cliente en lo sucesivo.

# Funcionamiento básico del sistema

Para explicar el funcionamiento del sistema, mostraremos mediante un diagrama de secuencia qué equipos intervienen en el arranque y en el uso posterior de una máquina virtual.

Nos limitaremos al caso más sencillo, en el que un único usuario arranca y utiliza una máquina virtual. Los pasos que se seguirán son:

1. El usuario se conecta al servidor para proceder a la autenticación.
2. Una vez autenticado, el usuario solicitará al servidor el uso de cierta máquina virtual.
3. El servidor enviará todo lo necesario para establecer la comunicación con el usuario a un servidor de máquinas virtuales. Por ahora, no consideraremos el balanceado de carga: la petición se asignará de forma muy sencilla (por ejemplo, siguiendo el algoritmo *Round Robin*). Aquí termina la intervención del servidor central.
4. El servidor de máquinas virtuales crea la máquina virtual, asignándole una dirección IP en la red virtual, un puerto en el servidor VNC y una contraseña.

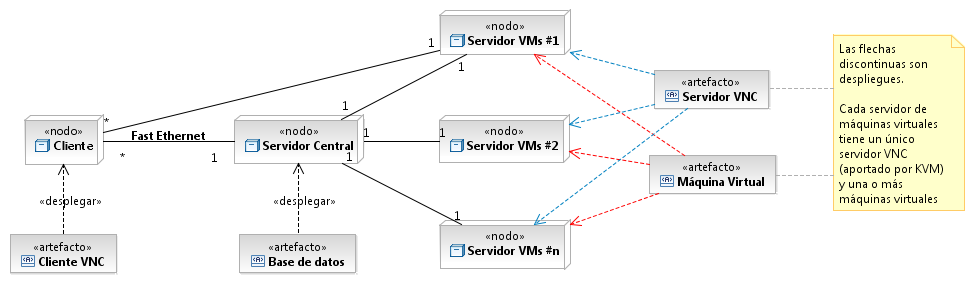


Figura Diagrama de despliegue básico

1. El servidor de máquinas virtuales envía todo lo necesario para realizar la conexión al cliente (por ahora, su IP, el puerto en el que escucha el servidor VNC y la contraseña).
2. El cliente ejecuta el cliente VNC y se conecta a la máquina virtual. A partir de este momento, puede utilizarla normalmente.

Todos estos pasos aparecen resumidos en un diagrama de secuencia.

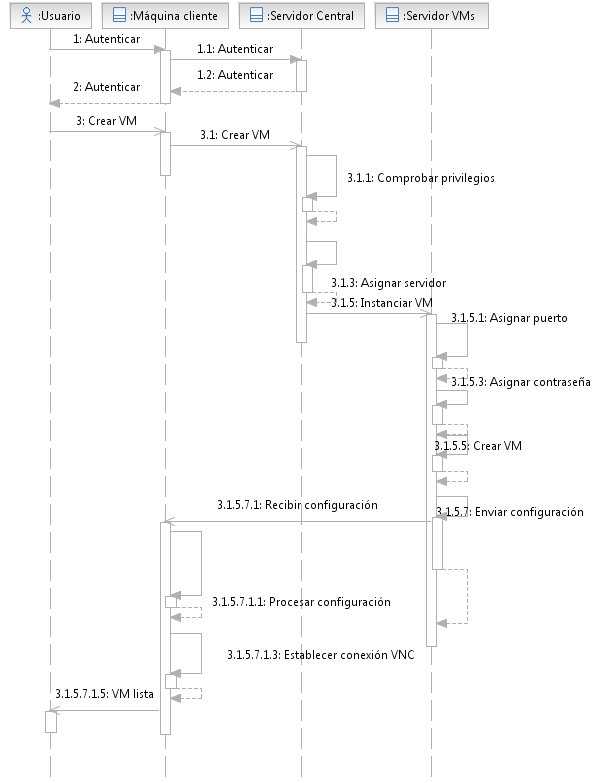


Figura Creación de una máquina virtual: diagrama de secuencia

El apagado de una máquina virtual también es bastante simple:

1. El usuario solicita el apagado de la máquina virtual.
2. El servidor de máquinas virtuales detecta el apagado de la máquina virtual.
3. El servidor de máquinas virtuales indica al servidor principal que una de sus máquinas se ha apagado normalmente.
4. El servidor principal registra el apagado.

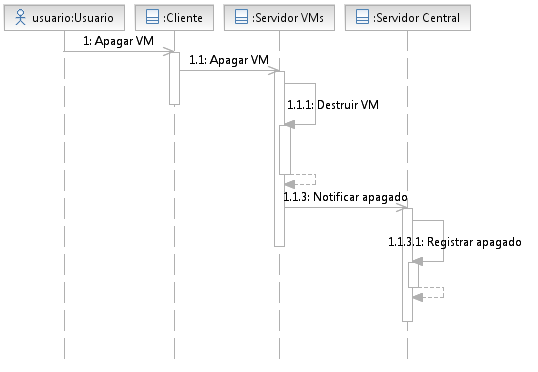


Figura Apagado de una máquina virtual: diagrama de secuencia

# Distribución de responsabilidades

Las funciones del servidor principal son las siguientes:

* Hospedar la web que los usuarios utilizarán para interactuar con el sistema.
* Realizar la autenticación de los usuarios.
* Registrar la creación y destrucción de máquinas virtuales.
* Detectar las caídas de los servidores de máquinas virtuales.
* Distribuir las peticiones entre los distintos servidores de máquinas virtuales (balanceado de carga).
* Permitir la administración del sistema, incluyendo:
  + Despliegue de imágenes
  + Borrado de imágenes ya existentes
  + Altas y bajas de usuarios
  + Altas y bajas de servidores de máquinas virtuales
  + Mostrar estadísticas en tiempo real

Por otra parte, los servidores de máquinas virtuales deben:

* Crear y destruir máquinas virtuales bajo petición del servidor principal.
* Proporcionar información relativa a su estado al servidor principal.
* Transferir imágenes a un cliente bajo petición del servidor central.
* Transferir los parámetros necesarios para establecer la conexión al servidor central.
* Detectar las desconexiones abruptas de los usuarios.
* Registrar y borrar imágenes bajo petición del servidor central.
* Enviar periódicamente estadísticas al servidor central.
* Incorporar las máquinas virtuales a su red virtual.

# Primer esquema de implementación

Para comunicar el servidor central con los distintos servidores de máquinas virtuales, utilizaremos *sockets*. En sus extremos habrá siempre un proceso a la escucha.

En el caso del servidor central, tendríamos tres tipos de proceso (uno por cada servidor de máquinas virtuales):

* Procesamiento de las peticiones recibidas desde la web.
* Recopilación del estado de los servidores de máquinas virtuales. Además de modificar la estructura de datos utilizada para mostrar estadísticas en tiempo real, este proceso también detectaría la caída de servidores de máquinas virtuales, el apagado de máquinas virtuales,…
* Envío de órdenes a los servidores de máquinas virtuales. Entre las órdenes consideradas, se encuentran la creación y destrucción de máquinas virtuales y la creación y destrucción de imágenes de disco.
* Transferencia de imágenes desde/hacia el servidor de máquinas virtuales.

Por otra parte, en los servidores de máquinas virtuales tendríamos cinco tipos de procesos, aunque en este caso sólo existiría un proceso de cada tipo:

* Creación y destrucción de máquinas virtuales.
* Envío de datos relacionados con el estado al servidor principal.
* Recepción de órdenes desde el servidor central.
* Transferencia y recepción de imágenes.
* Envío de la configuración de las máquinas virtuales a los clientes. Este proceso establecería la conexión con cada cliente asignado al servidor para enviarle los ajustes requeridos para establecer la conexión con el servidor VNC.

En ambos casos, el primer tipo de proceso no utiliza ningún *socket*. Para comunicar los distintos procesos de cada máquina, utilizaríamos colas.