

## NOCIONES EXAMEN DE ASO

**VIRTUALIZACIÓN**→ TECNOLOGÍA QUE PERMITE CREAR ENTORNOS VIRTUALES O SIMULADOS EN UN SISTEMA FÍSICO, LO QUE SIGNIFICA QUE SE PUEDEN EJECUTAR MÚLTIPLES SISTEMAS OPERATIVOS O APLICACIONES EN UNA MISMA MÁQUINA FÍSICA. ESTO SE LOGRA MEDIANTE LA CREACIÓN DE VM O CONTENEDORES QUE FUNCIONAN DE MANERA INDEPENDIENTE DENTRO DE ESTE SISTEMA FÍSICO.

**MÁQUINA VIRTUAL**→ INSTANCIA AISLADA Y VIRTUALIZADA DE UN S.O. QUE SE EJECUTA EN UN SERVIDOR FÍSICO. CADA UNA TIENE SUS PROPIOS RECURSOS VIRTUALES, ALMACENAMIENTO, CPU, ADAPTADORES DE RED, ETC.

**HIPERVISOR**→ MONITOR DE VM. PROCESO QUE CREA Y EJECUTA MÁQUINAS VIRTUALES. PERMITE QUE UN ORDENADOR HOST PRESTE SOPORTE A VARIAS VM INVITADAS MEDIANTE EL USO COMPARTIDO VIRTUAL DE SUS RECURSOS. DOS TIPOS:

- **TIPO 1**→ (BARE METAL) SE EJECUTAN DIRECTAMENTE EN EL HARDWARE FÍSICO DEL HOST. (KVM)
- **TIPO 2**→ (ALOJADOS) SE EJECUTAN COMO UNA CAPA DE SOFTWARE SOBRE UN S.O., COMO OTROS PROGRAMAS INFORMÁTICOS (VIRTUALBOX).

## TIPOS DE VIRTUALIZACIÓN:

- **TOTAL**: LA CASI TOTAL SIMULACIÓN DEL HARDWARE REAL (VIRTUALBOX, KVM)
- **PARCIAL**: EL AMBIENTE DE DESTINO NO SE SIMULA TOTALMENTE, SINO PARCIALMENTE. PUEDE SER NECESARIA LA MODIFICACIÓN DE ALGUNOS PROBLEMAS QUE SE EJECUTEN EN EL GUEST
- **PARAVIRTUALIZACIÓN**: NO SE SIMULA NINGÚN AMBIENTE DE HARDWARE. CADA PROGRAMA INVITADO - GUEST SE EJECUTA EN UN DOMINIO AISLADO, COMO SI SE EJECUTARAN POR SEPARADO. ES NECESARIA UNA MODIFICACIÓN ESPECÍFICA DE CADA PROGRAMA GUEST PARA SU EJECUCIÓN DE ÉSTE TIPO DE AMBIENTE.

## VENTAJAS DE LA VIRTUALIZACIÓN

- **CONSOLIDACIÓN DE SERVIDORES**
- **UTILIZACIÓN DE RECURSOS DE FORMA EFICIENTE**
- **AISLAMIENTO**
- **FLEXIBILIDAD**

**RAW** □ FORMATO SIMPLE DE ACCESO RÁPIDO. NO REALIZA COMPRESIÓN DE DATOS NI INSTANTÁNEAS. SÓLO EN FRÍO

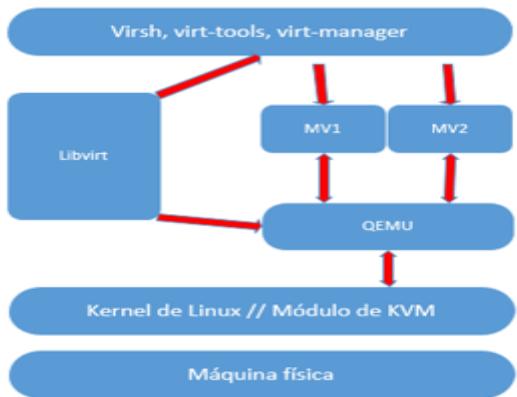
**QCOW2** □ FORMATO AVANZADO QUE PERMITE COMPRESIÓN DE DATOS E INSTANTÁNEAS Y COPY-ON-WRITE. (CLONACIÓN). PUEDE EN FRÍO Y EN CALIENTE.

**KVM** □ TECNOLOGÍA DE VIRTUALIZACIÓN COMPLETA DE CÓDIGO ABIERTO INTEGRADA EN LINUX, QUE PERMITE QUE ÉSTE ACTÚE COMO HIPERVISOR. PERMITE QUE UNA MÁQUINA FÍSICA EJECUTE MÚLTIPLES SISTEMAS OPERATIVOS DE FORMA SIMULTÁNEA. CON ELLA UTILIZAREMOS ADEMÁS QEMU PARA EMULAR DIFERENTES CARACTERÍSTICAS O PARTES DE HARDWARE.

- **VENTAJAS:** CÓDIGO ABIERTO, SIN CONFLICTOS DE DRIVERS, ESCALABILIDAD (MÚLTIPLES MÁQUINAS EN UN SOLO SERVIDOR FÍSICO), FLEXIBILIDAD (SO SE EJECUTAN DE FORMA INDEPENDIENTE), TRASLADO RÁPIDO, GRAN RENDIMIENTO POR SER PARTE DEL NÚCLEO DE LINUX
- **DESVENTAJAS:** SÓLO PARA LINUX, POCO INTUITIVO, DEPENDENCIA DEL HARDWARE SUBYACENTE, GESTIÓN DE ALMACENAMIENTO COMPLEJA, ALTO RIESGO DE PERDER LOS DATOS EN CASO DE ACCIDENTE O AVERÍA.

#### TIENE DOS COMPONENTES

- **KVM.KO** → módulo que actúa como componente central, responsable de gestionar la infraestructura de virtualización. Permite al S.O. anfitrión funcionar como hipervisor, creando y controlando máquinas virtuales.
- **QEMU** → trabaja conjuntamente con el módulo de kernel KVM para proporcionar un soporte de virtualización completo. funciona como un proceso de espacio de usuario, que emula dispositivos hardware. SU INTERFAZ GRÁFICA → VIRT MANAGER



## **QEMU □ EMULADOR Y VIRTUALIZADOR DE MAQUINAS DE CODIGO ABIERTO**

**LIBVIRT** □ ES UN TOOLKIT. PAQUETE DISEÑADO PARA LA ADMINISTRACIÓN Y GESTIÓN DE REDES VIRTUALES. BIBLIOTECA QUE PROPORCIONA UNA API PARA GESTIONAR LA VIRTUALIZACIÓN.

**LIBVIRTD** □ COMPONENTE DEMONIO, PROCESO EN SEGUNDO PLANO, QUE SE EJECUTA EN SERVIDORES HOST. ES UNA API (**interfaz de programación de aplicaciones**)

**VIRSH** □ INTERFAZ CLI PARA ADMINISTRAR MVS (CREAR, ENUMERAR, EDITAR...). APPLICACIÓN CONSTRUIDA SOBRE LA API LIBVIRT, ALTERNATIVA A VIRT-MANAGER

**VIRT-MANAGER** □ INTERFAZ DE USUARIO DE ESCRITO

**DISPOSITIVO VIRTIO** → dispositivos paravirtualizados o emulados. Específicos para KVM. Interfaz independiente del hipervisor diseñada para el desarrollo de drivers que puedan ser utilizados sobre varios hipervisores.

**POOL** □ ARCHIVO, DIRECTORIO O DISPOSITIVO DE ALMACENAMIENTO ADMINISTRADO PARA PROPORCIONAR ALMACENAMIENTO A VM. UN CONTENEDOR VIRTUAL DONDE SE ALMACENAN LOS DISCOS DUROS DE LAS MV.

- SIMPLIFICA, OPTIMIZA Y ALTA DISPONIBILIDAD, GESTIÓN FLEXIBLE

**MIGRACIÓN** □ MOVER UNA VM A OTRO SERVIDOR FÍSICO O NUBE PUBLICA

**COPIA SEGURIDAD** □ CREAR UNA COPIA DE VM

**DISTRIBUCIÓN** □ COMPARTIR VM CON OTROS USUARIOS O EQUIPOS

**CLONACIÓN** □ CREAR MÚLTIPLES INSTANCIAS DE UNA MISMA MV

**PROCESO EXPORTACIÓN** □ PARADA VM, EXPORTAR XML, EXPORTAR DISCO Y VERIFICAR

**REDES** □ **2 GRUPOS**

1. **REDES VIRTUALES (PRIVADAS)**: CON BRIDGE INTERNO: VMBRX
  - TIPO NAT (LA DEFAULT). LAS VM PUEDEN COMUNICARSE CON EL EXTERIOR PERO EL EXTERIOR CON ELLAS NO.
  - TIPO AISLADAS (ISOLATED): NO TIENEN CONECTIVIDAD CON EL EXTERIOR. RED PRIVADA ENTRE EL HIPERVISOR Y LA VM.
  - TIPO MUY AISLADAS (VERY SOLATED): EL HOST NO ESTÁ CONECTADO A LAS VM, POR LO QUE NO HAY SERVIDOR DNS NI DHCP, TAMPOCO TIENE SALIDA AL EXTERIOR.
  - EN MODO ABIERTO: LAS VM SE COMUNICAN ENTRE SI DENTRO DE LA RED PRIVADA, PERO SIN AISLAMIENTO
- NAT=PUENTE DIRECTO CON ACCESO EXTERNO:  
2. **REDES PUENTES (PÚBLICAS)**: CON BRIDGE EXTERNO: BRX
  - NO TIENEN EL AISLAMIENTO DE LA NAT, COMPORTAMIENTO SIMILAR A UN PUENTE DIRECTO CON ACCESO EXTERNO.
    - o LAS VM PUEDEN ACCEDER A LA RED EXTERNA DIRECTAMENTE
    - o NO TIENEN AISLAMIENTO
  - COMPARTEN LA INTERFAZ FÍSICA DEL HOST. USA UNA CONEXIÓN MACVTAP

**PUENTE** □ DISPOSITIVO DE INTERCONEXIÓN DE REDES

**POOL** □ ARCHIVO, DIRECTORIO O DISPOSITIVO DE ALMACENAMIENTO, QUE PROPORCIONA ALMACENAMIENTO A VM. ES COMO UN CONTENEDOR DONDE SE ALMACENAN LOS DISCOS DUROS DE LAS VM □ SIMPLIFICACIÓN, OPRIMIZACIÓN, ALTA DISPONIBILIDAD

**STORAGE POOLS** □ CONCEPTO DE VIRTUALIZACIÓN PARA GESTIONAR LOS VOLÚMENES DE ALMACENAMIENTO

**DIFERENCIAS:**

- **POOL CREATE** □ DEFINE, CREA Y ACTIVA INMEDIATAMENTE
- **POOL-DEFINE** □ SOLO DEFINE. HAY QUE ACTIVARLO MANUAL.

**VOLUMEN** □ DIVISIÓN DE LOS POOLS. SON UTILIZADOS COMO DISCOS EN VM. SON LOS MEDIOS DE ALMACENAMIENTO QUE UTILIZAN LA MV

## **TECNOLOGIAS DE ALMACENAMIENTO DE POOLS**

- SISTEMA DE FICHERO O DISPOSITIVO DE BLOQUE
- ALMACENAMIENTO COMPARTIDO
- SNAPSHOT: MIGRACIÓN
- APROVISIONAMIENTO LIGERO: AHORRO DE ESPACIO

### **POOL TIPO DIR □ DIRECTORIO HOST. FICHEROS DE FORMATO:**

- RAW □ FORMATO SIMPLE Y DE ACCESO RÁPIDO. OCUPA TODO SU ESPACIO. NO SOPORTA NI SNAPSHTOS NI APROVISIONAMIENTO LIGERO
- QCOW2 □ SOLO OCUPA EL ESPACIO QUE ESTÁ OCUPADO. IRÁ CRECIENDO CONFORME ESCRIBAMOS EN ÉL.

### **POOL DE ALMACENAMIENTO, TIPOS:**

- LOGICAL
- NETFS

**SNAPSHOT** □ INSTANTÁNEA. GUARDA EL ESTADO DE UNA MAQUINA VIRTUAL EN UN DETERMINADO MOMENTO. SE GUARDAN EL ESTADO DEL DISCO Y DE LA MEMORIA.

**Revertir** □ vuelve al punto en que se hizo la instantánea

**Consolidar** □ borra la instantánea y se queda como máquina original por dónde iba

**PROCESO DE EXPORTACIÓN** □ PARAR MV, EXPORTAR CONFIGURACIÓN, EXPORTAR DISCOS, VERIFICAR EXPORTACIÓN.

**SSH** □ PROTOCOLO DE RED SEGURO QUE PERMITE CONEXIONES REMOTAS A UN SERVIDOR UTILIZANDO UN NOMBRE DE USUARIO Y UNA CONTRASEÑA

### **DIFERENCIAS ENTRE SNAPSHOT INTERNOS Y EXTERNOS:**

Característica	Snapshots Internos	Snapshots Externos
Ubicación	Dentro del sistema	En un dispositivo externo
Rendimiento	Rápido	Más lento
Espacio de Almacenamiento	Ocupa espacio adicional	No ocupa espacio adicional
Dependencia	Dependiente del sistema	Independiente del sistema
Uso	Copias de seguridad rápidas, revertir cambios	Copias de seguridad a largo plazo, protección contra desastres

**PLANTILLA** □ IMAGEN PRECONFIGURADA DE UN SO QUE PUEDE UTILIZARSE PARA DESPLEGAR RAPIDAMENTE VM. CON ELLA, POEMOS:

- **CLONACIÓN COMPLETA (FULL)** □ COPIA COMPLETA INDEPENDIENTE DE LA PLANTILLA. REQUIERE MISMO ESPACIO MV ORIGINAL
- **CLONACIÓN ENLAZADA (LINKED)** □ IMAGEN SOLO LECTURA Y VINCULA LA IMAGEN DE COPIA EN ESCRITURA PARA ALMACENAR LOS NUEVOS DATOS ALMACENADOS. NECESITA LA PLANTILLA BASE PARA EJECUTARSE.

### **RUTAS:**

Fichero de redes □ **/etc/network/interfaces** □ archivo que contiene la configuración de las interfaces de red.

Fichero redes xml □ **/etc/libvirt/qemu/networks/default.xml**: Contiene las definiciones de las redes virtuales.

Ruta pool □ **/var/lib/libvirt/images** □ donde se almacenan imágenes de discos virtuales utilizados por VMs administradas por libvirt

Ruta xml para volcar pool □ **/etc/libvirt/storage** □ contiene las configuraciones de almacenamiento de libvirt. Archivos xml que definen grupos de almacenamiento.

Ruta de las iso □ **/var/lib/libvirt/images/.qcow2**

Ruta ssh.conf □ **/ssh/sshd.conf** □ el estándar normalmente es **/etc/ssh/sshd\_config** □ configura el servidor SSH

