

Kernel Based Virtual Machine (KVM) es una solución de virtualización completa para linux que funciona sobre procesadores x86 con extensiones de virtualización por hardware. El componente principal de kvm es el módulo `kvm.ko` que fue incluido en el *kernel* linux 2.6.20.

KVM no proporciona una solución completa de virtualización, sino que reutiliza muchos componentes del proyecto de emulación y virtualización **qemu**.

Se puede utilizar KVM directamente o a través de la API de virtualización **libvirt**, como en el caso de OpenStack.

Lanzamiento de una instancia

Para lanzar una instancia utilizamos una instrucción del tipo con cualquier usuario de OpenStack:

```
$ nova boot --image 4ddb27ab-b3cc-4a65-ac52-f4ce7894e4ed \  
--flavor 1 \  
--key_name miclave \  
--nic net-id=0d43e5fe-99b9-46a8-b118-58136aa805b4 \  
test2
```

Si ejecutamos como administrador la instrucción:

```
$ nova show f69ab4d9-ed28-406e-8a78-67f62a5e4bc2  
-----  
---+  
| Property                                | Value  
|-----+-----  
---+  
| OS-DCF:diskConfig                       | MANUAL  
| OS-EXT-AZ:availability_zone             | nova  
| OS-EXT-SRV-ATTR:host                     | rdo2  
| OS-EXT-SRV-ATTR:hypervisor_hostname     | rdo2  
| OS-EXT-SRV-ATTR:instance_name           | instance-0000001c  
| OS-EXT-STS:power_state                   | 1  
| OS-EXT-STS:task_state                    | -  
| OS-EXT-STS:vm_state                      | active
```

```

| OS-EXT-SRV:vm_state | active
|
| OS-SRV-USG:launched_at | 2014-05-29T06:42:07.000000
|
| OS-SRV-USG:terminated_at | -
|
| accessIPv4 |
|
| accessIPv6 |
|
| config_drive |
|
| created | 2014-05-29T06:41:26Z
|
| flavor | m1.tiny (1)
|
| hostId | 4345a37408b6a85fb6a2eb6322d0edb5724aeaa2d1bcf903d67ee4
87 |
| id | f69ab4d9-ed28-406e-8a78-67f62a5e4bc2
|
| image | Cirros 0.3.2 (4ddb27ab-b3cc-4a65-ac52-f4ce7894e4ed)
|
| key_name | miclave
|
| metadata | {}
|
| name | test2
|
| os-extended-volumes:volumes_attached | []
|
| privada network | 10.0.0.5
|
| progress | 0
|
| security_groups | default
|
| status | ACTIVE
|
| tenant_id | 0912c80bef254d7b9352632793cf75b9
|
| updated | 2014-05-29T06:42:07Z
|
| user_id | 0f94f0bbcdee4715ba2750502bb0d63f
|
+-----+-----+
---+

```

Veremos algunas propiedades de esta instancia muy interesantes para obtener más información, en particular “id”, OS-EXT-SRV-ATTR:host, OS-EXT-SRV-ATTR:hypervisor_hostname, OS-EXT-SRV-ATTR:instance_name.

Si accedemos al nodo de computación en el que se está ejecutando esta máquina virtual (rdo2 según la información anterior), podemos ir al directorio /var/lib/nova/instances y listar el contenido del mismo:

```
$ ls -l /var/lib/nova/instances/
drwxr-xr-x. 2 nova nova 4096 may 22 17:35 065cdd8b-fb70-45ca-b309-8aa26be8bbc0
drwxr-xr-x. 2 nova nova 4096 may 27 21:22 3773a78d-c5ed-417a-a5a1-3dc89f3e836b
drwxr-xr-x. 2 nova nova 4096 may 29 08:41 _base
-rw-r--r--. 1 nova nova 26 may 29 08:35 compute_nodes
drwxr-xr-x. 2 nova nova 4096 may 29 08:42 f69ab4d9-ed28-406e-8a78-67f62a5e4bc2
drwxr-xr-x. 2 nova nova 4096 may 27 21:22 locks
drwxr-xr-x. 2 nova nova 4096 may 27 20:16 snapshots
```

Donde vemos un directorio para cada una de las instancias (el nombre de cada directorio se corresponde al id de la misma), además de los directorios `_base`, `lock` y `snapshots`.

Si accedemos a continuación al directorio de la instancia que acabamos de lanzar y listamos el contenido:

```
-rw-rw----. 1 qemu qemu 18803 may 29 08:43 console.log
-rw-r--r--. 1 qemu qemu 1835008 may 29 08:43 disk
-rw-r--r--. 1 nova nova 79 may 29 08:41 disk.info
-rw-r--r--. 1 nova nova 1548 may 29 08:42 libvirt.xml
```

El fichero `libvirt.xml` se corresponde con la definición de la máquina virtual utilizada por `libvirt` y su contenido es:

```
<domain type="kvm">
  <uuid>f69ab4d9-ed28-406e-8a78-67f62a5e4bc2</uuid>
  <name>instance-0000001c</name>
  <memory>524288</memory>
  <vcpu>1</vcpu>
  <sysinfo type="smbios">
    <system>
      <entry name="manufacturer">Red Hat Inc.</entry>
      <entry name="product">OpenStack Nova</entry>
      <entry name="version">2013.2.3-1.el6</entry>
      <entry name="serial">e21da79e-b1eb-11de-b487-00e01884fce4</entry>
      <entry name="uuid">f69ab4d9-ed28-406e-8a78-67f62a5e4bc2</entry>
    </system>
  </sysinfo>
  <os>
    <type>hvm</type>
    <boot dev="hd"/>
    <smbios mode="sysinfo"/>
  </os>
  <features>
    <acpi/>
    <apic/>
  </features>
  <clock offset="utc">
    <timer name="pit" tickpolicy="delay"/>
    <timer name="rtc" tickpolicy="catchup"/>
  </clock>
  <cpu mode="host-model" match="exact"/>
```

```

<devices>
  <disk type="file" device="disk">
    <driver name="qemu" type="qcow2" cache="none"/>
    <source file="/var/lib/nova/instances/f69ab4d9-ed28-406e-8a78-67f62a5e4bc2/disk"/>
    <target bus="virtio" dev="vda"/>
  </disk>
  <interface type="bridge">
    <mac address="fa:16:3e:a0:07:c9"/>
    <model type="virtio"/>
    <source bridge="qbrd440070c-5a"/>
    <target dev="tapd440070c-5a"/>
  </interface>
  <serial type="file">
    <source path="/var/lib/nova/instances/f69ab4d9-ed28-406e-8a78-67f62a5e4bc2/console.log"/>
  </serial>
  <serial type="pty"/>
  <input type="tablet" bus="usb"/>
  <graphics type="vnc" autoport="yes" keymap="en-us" listen="192.168.0.68"/>
</devices>
</domain>

```

En este fichero vemos todos los detalles de la máquina virtual y en particular el significado de los ficheros disk y console.log. El primero es un fichero de tipo qcow2 que se corresponde con el disco /dev/vda que verá la instancia y el segundo es un fichero que muestra la salida por consola:

```

...
ip-route:default via 10.0.0.1 dev eth0
ip-route:10.0.0.0/24 dev eth0 src 10.0.0.5
=== datasource: ec2 net ===
instance-id: i-0000001c
name: N/A
availability-zone: nova
local-hostname: test2.novalocal
launch-index: 0
=== cirros: current=0.3.2 uptime=64.37 ===

  ____          _      _      _      _      _
 /  _/  _  _  _  _  _  /  _  \  _/
/  _/  //  //  //  //  //  //  //  \  \
\  _/  //  //  //  //  \  _/  _/

  http://cirros-cloud.net

login as 'cirros' user. default password: 'cubswin:)' . use 'sudo' for root.

```

Del disco podemos obtener más información con la instrucción qemu-img:

```

# qemu-img info /var/lib/nova/instances/f69ab4d9-ed28-406e-8a78-67f62a5e4bc2/disk
image: /var/lib/nova/instances/f69ab4d9-ed28-406e-8a78-67f62a5e4bc2/disk
file format: qcow2
virtual size: 1.0G (1073741824 bytes)
disk size: 1.7M

```

```
cluster_size: 65536
backing file: /var/lib/nova/instances/_base/24664fbf8133385a2743951bde673f75d7991092)
```

Vemos que se trata de un disco que virtualmente tiene 1 GiB pero que actualmente el fichero donde se encuentra solo ocupa 17 MiB. Además vemos que este fichero utiliza un fichero de respaldo ubicado en el directorio `/var/lib/nova/instances/_base/`. Esto es una técnica utilizada por OpenStack a través de `qemu` para utilizar el mismo disco base para todas las instancias que tengan una imagen de partida común.

Definición de la máquina virtual

Uno de los parámetros que vimos al mostrar los detalles de la instancia era “OS-EXT-SRV-ATTR:instance_name” que tenía el valor `instance-0000001c`. Si vemos las máquinas virtuales definidas en el nodo de computación:

```
# virsh list --all
 Id      Nombre                               Estado
-----
 1       instance-0000001c                    ejecutando
-       instance-00000016                    apagado
-       instance-0000001b                    apagado
```

Podemos comprobar que en estos momentos se está ejecutando la instancia que hemos arrancado recientemente y hay otras dos definidas y apagadas. Aunque el fichero XML que se utiliza como base para crear las instancias sobre es el anteriormente visto `libvirt.xml`, los ficheros XML que se utilizan en cada momento en `libvirt` se sitúan en el directorio `/etc/libvirt/qemu`:

```
# ls -l /etc/libvirt/qemu
total 16
-rw-----. 1 root root 3017 may 22 17:35 instance-00000016.xml
-rw-----. 1 root root 3019 may 28 16:29 instance-0000001b.xml
-rw-----. 1 root root 3017 may 29 08:42 instance-0000001c.xml
```