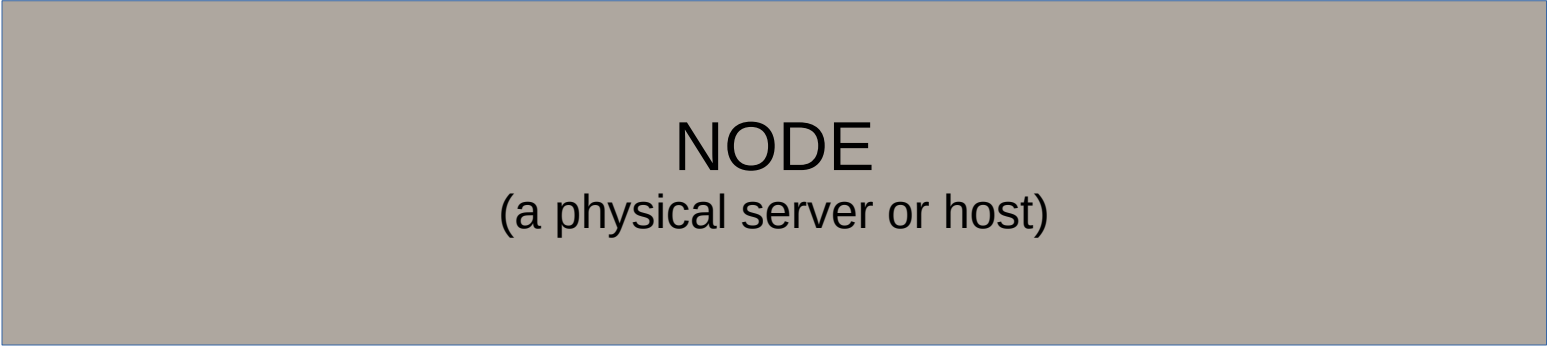# Working with libvirt

David Clinton

LINUX SYSTEM ADMINISTRATOR

@davidbclinton | www.bootstrap-it.com | www.bootstrap-it.com/blog

libvirt
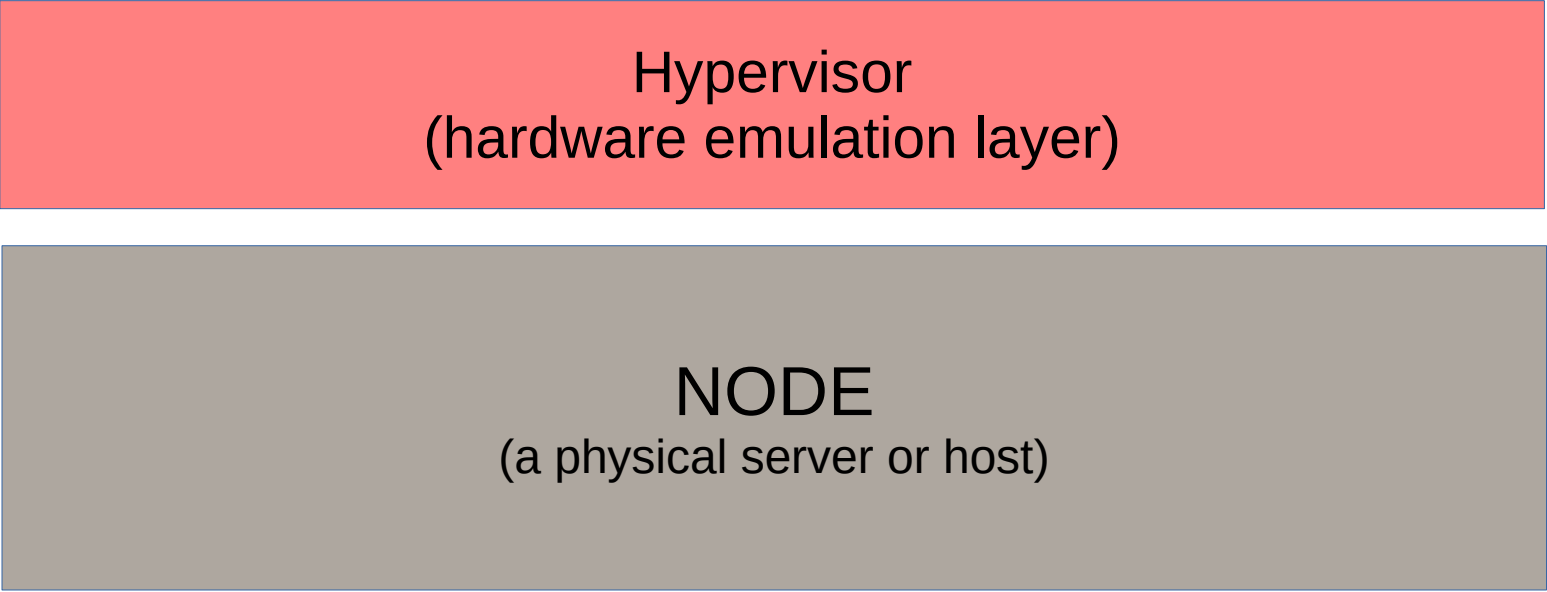Interfaces

- Hypervisor drivers
- Command line shell (or GUI)
- API

The libvirt Naming Scheme

NODE
(a physical server or host)
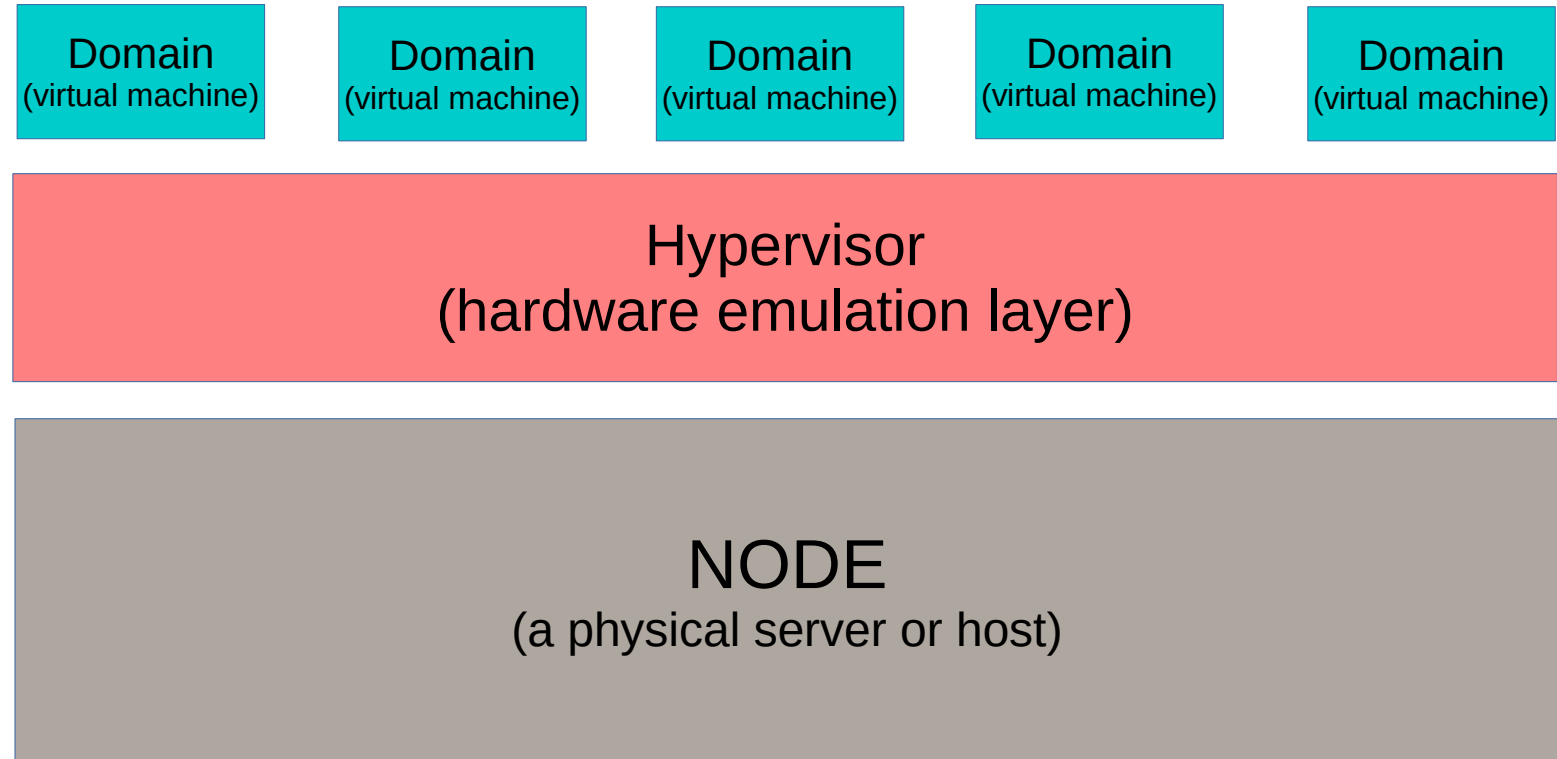
The libvirt Naming Scheme

Hypervisor
(hardware emulation layer)

NODE
(a physical server or host)

The libvirt
Naming
Scheme

| Domain (virtual machine) | Domain (virtual machine) | Domain (virtual machine) | Domain (virtual machine) | Domain (virtual machine) |

Hypervisor
(hardware emulation layer)

NODE
(a physical server or host)

Environments
with libvirt
Emulation
Drivers

- Xen
- QEMU
- LXC
- OpenVZ
- UML (User Mode Linux)
- VirtualBox
- VMware ESX
- VMware Workstation/Player
- Microsoft Hyper-V
- IBM PowerVM (phyp)
- Virtuozzo
- Bhyve (the BSD hypervisor)
- Test (used for testing)

Storage Backends with libvirt Emulation Drivers

- Directory
- Local filesystem
- Network filesystem
- Logical Volume Manager (LVM)
- Disk
- iSCSI
- SCSI
- Multipath
- RBD (RADOS Block Device)
- Sheepdog

Working with VirtualBox

Installing libvirt and Creating a Domain

virt-install Parameters

```
virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

virt-install
Parameters

```
virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

virt-install Parameters

```
virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

virt-install
Parameters

```
virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

virt-install
Parameters

```
virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

virt-install Parameters

```
virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

virt-install
Parameters

virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v

virt-install Parameters

```
virt-install -n my_vm -r 256
--disk path=/var/lib/libvirt/images/my_vm.img,bus=virtio,size=4 \
-c ubuntu-14.04.4-server-amd64.iso \
--network network=default,model=virtio \
--graphics vnc,listen=0.0.0.0 --noautoconsole -v
```

Working with Virsh

Sample XML Configuration File

```xml
<domain type='qemu'>
  <name>QEmu-fedora-i686</name>
  <uuid>c7a5fdbd-cdaf-9455-926a-d65c16db1809</uuid>
  <memory>219200</memory>
  <currentMemory>219200</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='i686' machine='pc'>hvm</type>
    <boot dev='cdrom'/>
  </os>
  <devices>
    <emulator>/usr/bin/qemu-system-x86_64</emulator>
    <disk type='file' device='cdrom'>
      <source file='/home/user/boot.iso'/>
      <target dev='hdc'/>
      <readonly/>
    </disk>
    <disk type='file' device='disk'>
      <source file='/home/user/fedora.img'/>
      <target dev='hda'/>
    </disk>
    <interface type='network'>
      <source network='default'/>
    </interface>
    <graphics type='vnc' port='-1'/>
```

Sample XML Configuration File

```xml
<domain type='qemu'>
 <name>QEmu-fedora-i686</name>
 <uuid>c7a5fdbd-cdaf-9455-926a-d65c16db1809</uuid>
 <memory>219200</memory>
 <currentMemory>219200</currentMemory>
 <vcpu>2</vcpu>
 <os>
   <type arch='i686' machine='pc'>hvm</type>
   <boot dev='cdrom'/>
 </os>
 <devices>
   <emulator>/usr/bin/qemu-system-x86_64</emulator>
   <disk type='file' device='cdrom'>
     <source file='/home/user/boot.iso'/>
     <target dev='hdc'/>
     <readonly/>
   </disk>
   <disk type='file' device='disk'>
     <source file='/home/user/fedora.img'/>
     <target dev='hda'/>
   </disk>
   <interface type='network'>
     <source network='default'/>
   </interface>
   <graphics type='vnc' port='-1'/>
```

Sample XML Configuration File

```xml
<domain type='qemu'>
  <name>QEmu-fedora-i686</name>
  <uuid>c7a5fdbd-cdaf-9455-926a-d65c16db1809</uuid>
  <memory>219200</memory>
  <currentMemory>219200</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='i686' machine='pc'>hvm</type>
    <boot dev='cdrom'/>
  </os>
  <devices>
    <emulator>/usr/bin/qemu-system-x86_64</emulator>
    <disk type='file' device='cdrom'>
      <source file='/home/user/boot.iso'/>
      <target dev='hdc'/>
      <readonly/>
    </disk>
    <disk type='file' device='disk'>
      <source file='/home/user/fedora.img'/>
      <target dev='hda'/>
    </disk>
    <interface type='network'>
      <source network='default'/>
    </interface>
    <graphics type='vnc' port='-1'/>
```

Sample XML Configuration File

```xml
<domain type='qemu'>
  <name>QEmu-fedora-i686</name>
  <uuid>c7a5fdbd-cdaf-9455-926a-d65c16db1809</uuid>
  <memory>219200</memory>
  <currentMemory>219200</currentMemory>
  <vcpu>2</vcpu>
  <os>
    <type arch='i686' machine='pc'>hvm</type>
    <boot dev='cdrom'/>
  </os>
  <devices>
    <emulator>/usr/bin/qemu-system-x86_64</emulator>
    <disk type='file' device='cdrom'>
      <source file='/home/user/boot.iso'/>
      <target dev='hdc'/>
      <readonly/>
    </disk>
    <disk type='file' device='disk'>
      <source file='/home/user/fedora.img'/>
      <target dev='hda'/>
    </disk>
    <interface type='network'>
      <source network='default'/>
    </interface>
    <graphics type='vnc' port='-1'/>
```

Virsh
Editing
Tools

Virtual networks:
   net-edit
   net-dumpxml
Storage pools:
   pool-edit
   pool-dumpxml
Storage volumes:
   vol-edit
   vol-dumpxml
Interfaces:
   iface-edit
   iface-dumpxml

Working with GUI Administration Managers

The libvirt API

A Python API Call

```python
import libvirt
conn = libvirt.open('qemu:///system')
for id in conn.listDomainsID():
    dom = conn.lookupByID(id)
    print "Dom %s  State %s" % ( dom.name(), dom.info()[0] )
    dom.suspend()
    print "Dom %s  State %s (after suspend)" % ( dom.name(), dom.info()[0] )
    dom.resume()
    print "Dom %s  State %s (after resume)" % ( dom.name(), dom.info()[0] )
    dom.destroy()
```

A Python
API Call

```python
import libvirt
conn = libvirt.open('qemu:///system')
for id in conn.listDomainsID():
    dom = conn.lookupByID(id)
    print "Dom %s  State %s" % ( dom.name(), dom.info()[0] )
    dom.suspend()
    print "Dom %s  State %s (after suspend)" % ( dom.name(), dom.info()[0] )
    dom.resume()
    print "Dom %s  State %s (after resume)" % ( dom.name(), dom.info()[0] )
    dom.destroy()
```

A Python
API Call

```python
import libvirt
conn = libvirt.open('qemu:///system')
for id in conn.listDomainsID():
    dom = conn.lookupByID(id)
    print "Dom %s  State %s" % ( dom.name(), dom.info()[0] )
    dom.suspend()
    print "Dom %s  State %s (after suspend)" % ( dom.name(), dom.info()[0] )
    dom.resume()
    print "Dom %s  State %s (after resume)" % ( dom.name(), dom.info()[0] )
    dom.destroy()
```

A Python API Call

```python
import libvirt
conn = libvirt.open('qemu:///system')
for id in conn.listDomainsID():
    dom = conn.lookupByID(id)
    print "Dom %s  State %s" % ( dom.name(), dom.info()[0] )
    dom.suspend()
    print "Dom %s  State %s (after suspend)" % ( dom.name(), dom.info()[0] )
    dom.resume()
    print "Dom %s  State %s (after resume)" % ( dom.name(), dom.info()[0] )
    dom.destroy()
```

API Access
by Object

- network   (virNetworkPtr)
- storage volume (virStorageVolPtr)
- storage pool   (virStoragePoolPtr)

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager
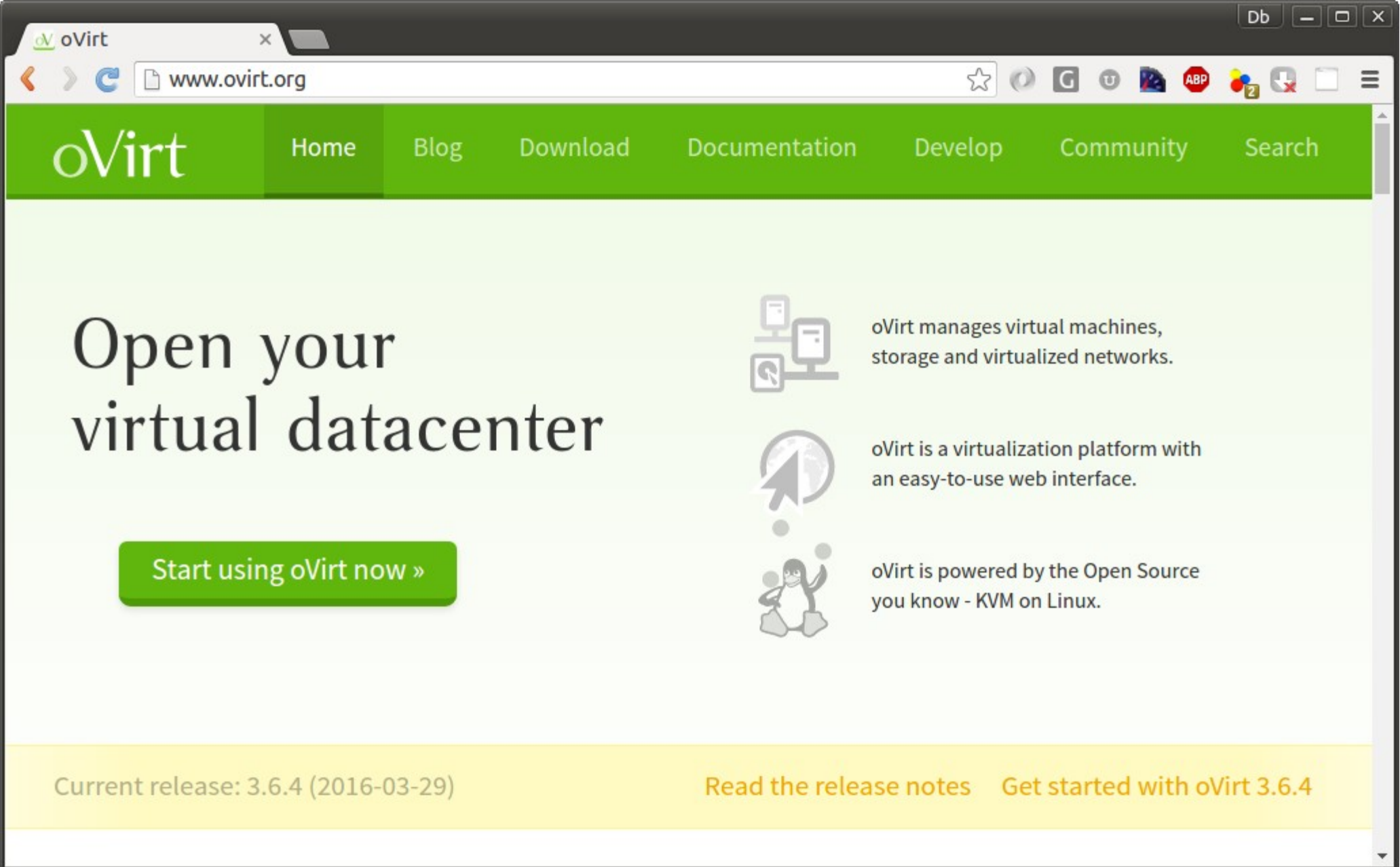
import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

Review

Node: physical host (server)

Hypervisor: VM management software

Domain: virtual machine

Configuration file: /etc/libvirt/

File systems: /var/lib/libvirt

virt-install -n my_vm -r 256 ...

--network network=default

--disk path=/var/lib/libvirt/images/my_vm.img

sudo virsh create /etc/libvirt/qemu/filename.xml

sudo virsh edit my_vm

sudo virsh dumpxml my_vm

sudo virsh iface-dumpxml

virt-manager

import libvirt

conn = libvirt.open('qemu:///system')

API Access
by Object

- network (virNetworkPtr)
- storage volume (virStorageVolPtr)
- storage pool (virStoragePoolPtr)

libvirt API documentation:
www.libvirt.org/html/index.html

oVirt

oVirt
Interface