ANAWS

Antonio Virdis

Email: antonio.virdis@unipi.it

Webpage: www.iet.unipi.it/a.virdis/

Office hours: on Thursday 9:30-11:30 - by appointment via email

Lectures Structure and Organization

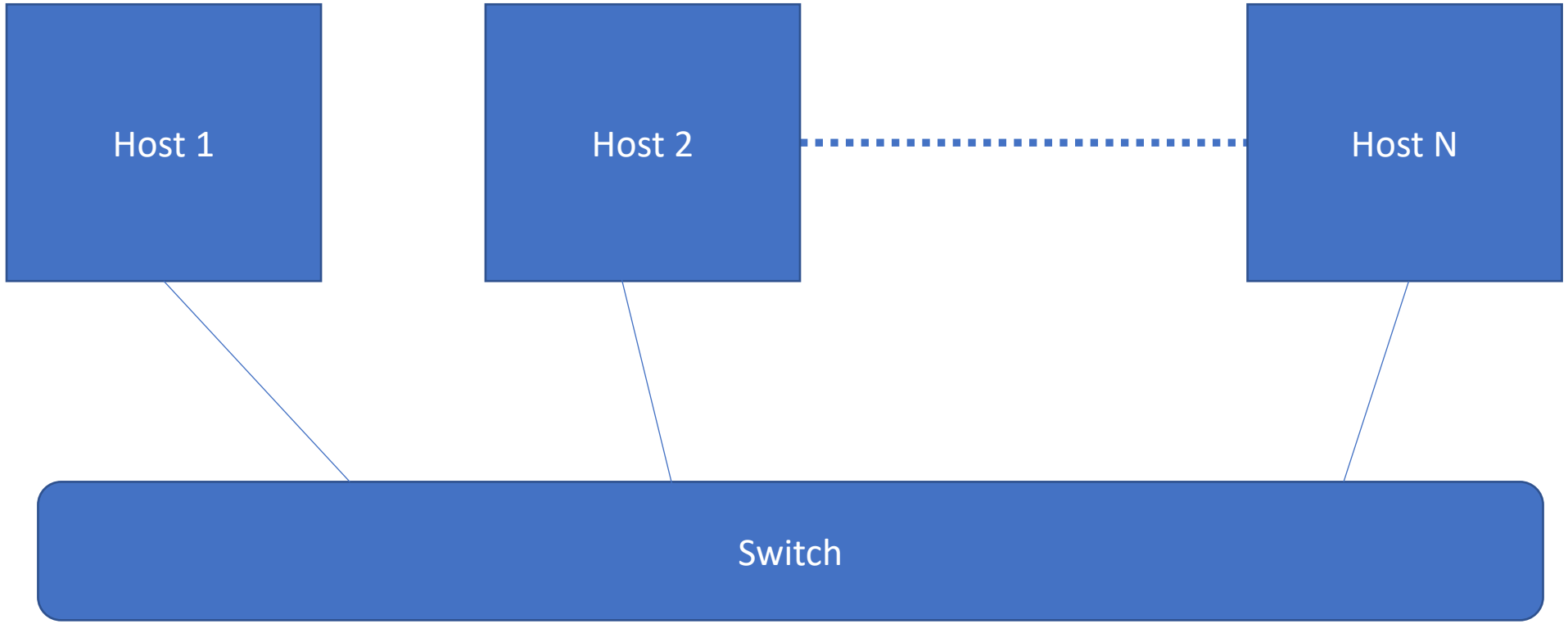
- Mixed theoretical/practical lectures:
  - Concepts on both networking theory and network configuration
  - Hands-on sessions
- Lectures will be held (typically) on Mondays 10.30-12.30 AM -> **ADII**
- We will work mostly through VMs. Please setup a virtual-machine environment. (I will refer to Virtual Box)
- The lab in ADII allows you to bring your own laptop (wifi, power outlets)

Topics

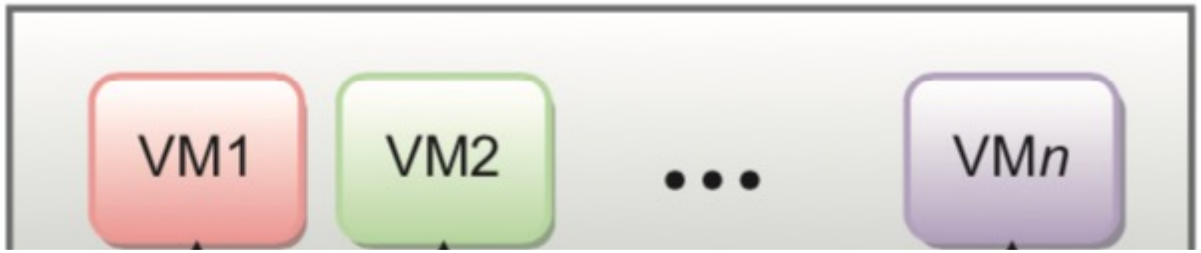
- Virtual Networking
- Software Defined Networking (SDN)
- Network Function Virtualization
- Network support to QoS
- Network Automation

Server Virtualization and Networking

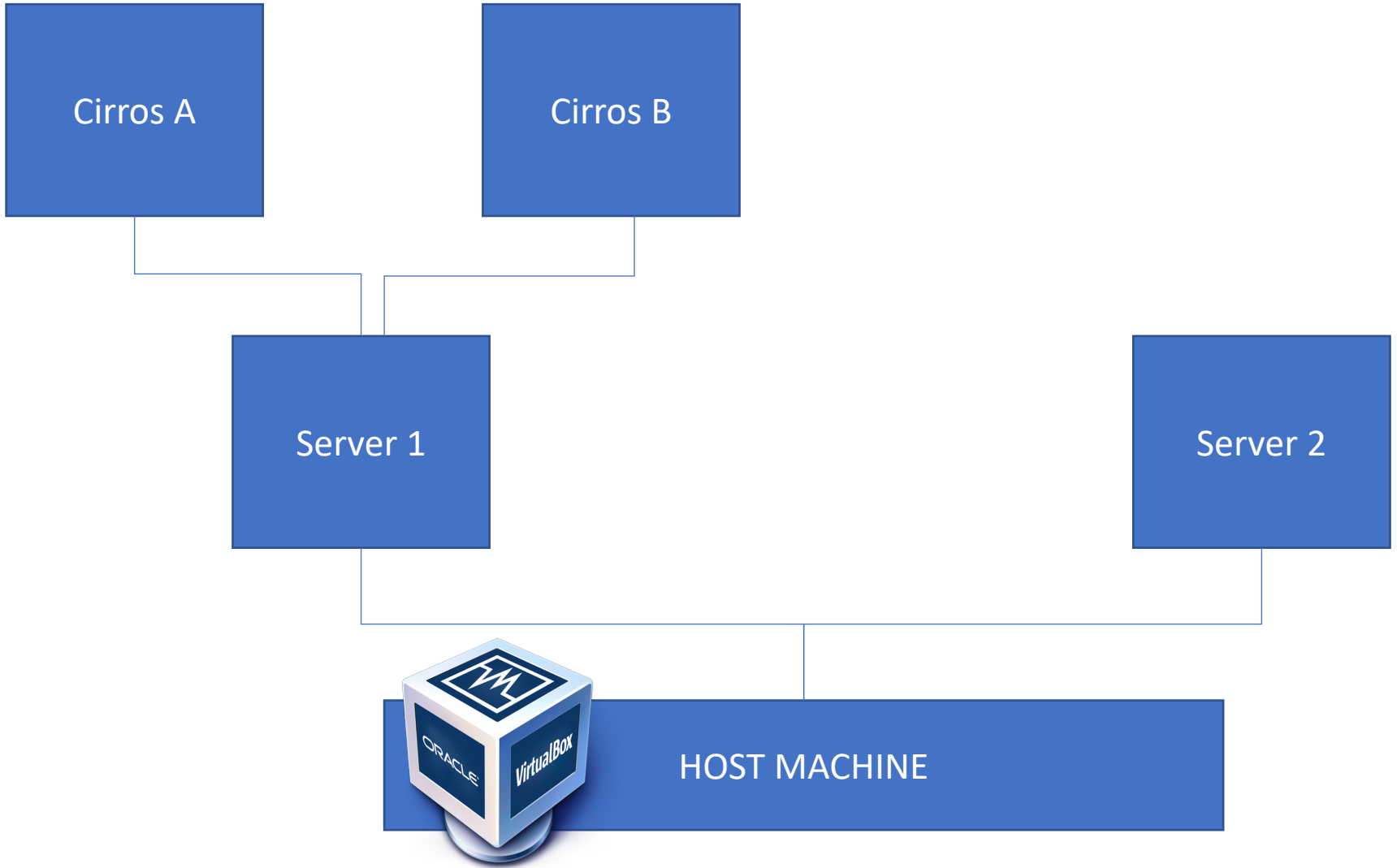
Network Switching



Virtual Switching

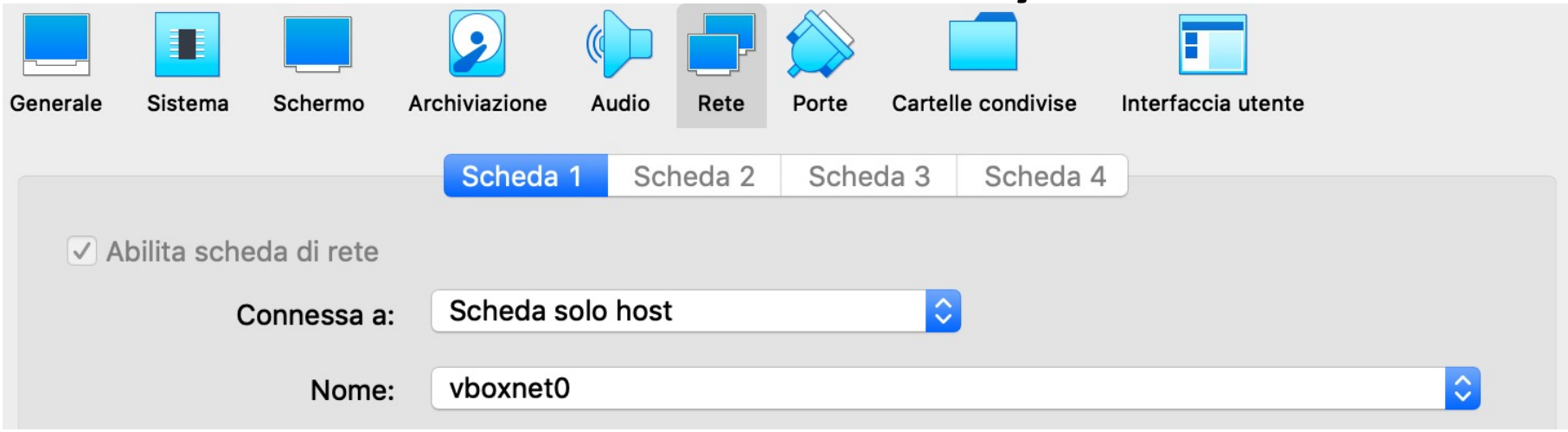


Lab configuration

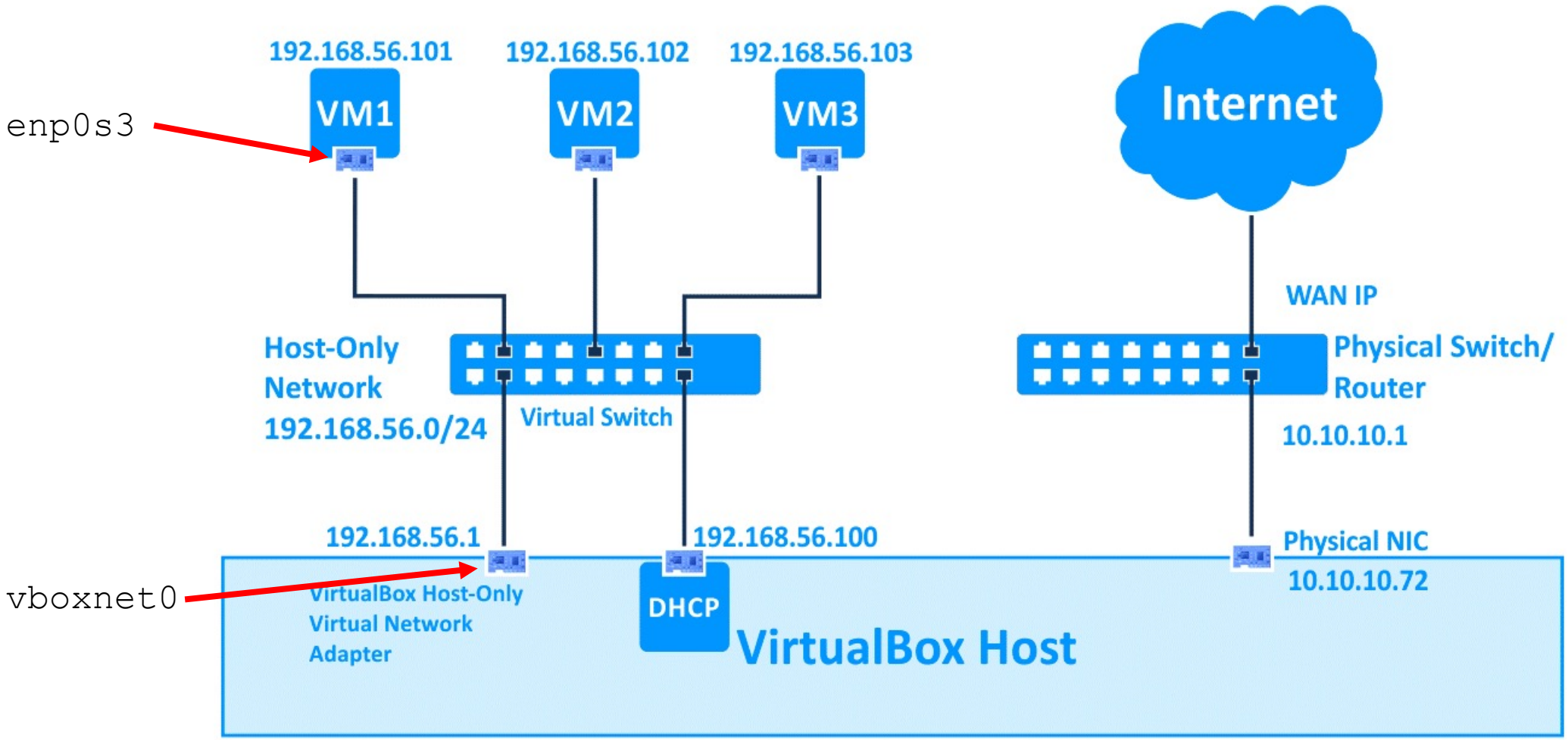


Setting up the environment

- Create a virtualbox network
- Create and start Server 1 using the image I provided you
- Configure a network card to work in **host-only** mode



Host-only network

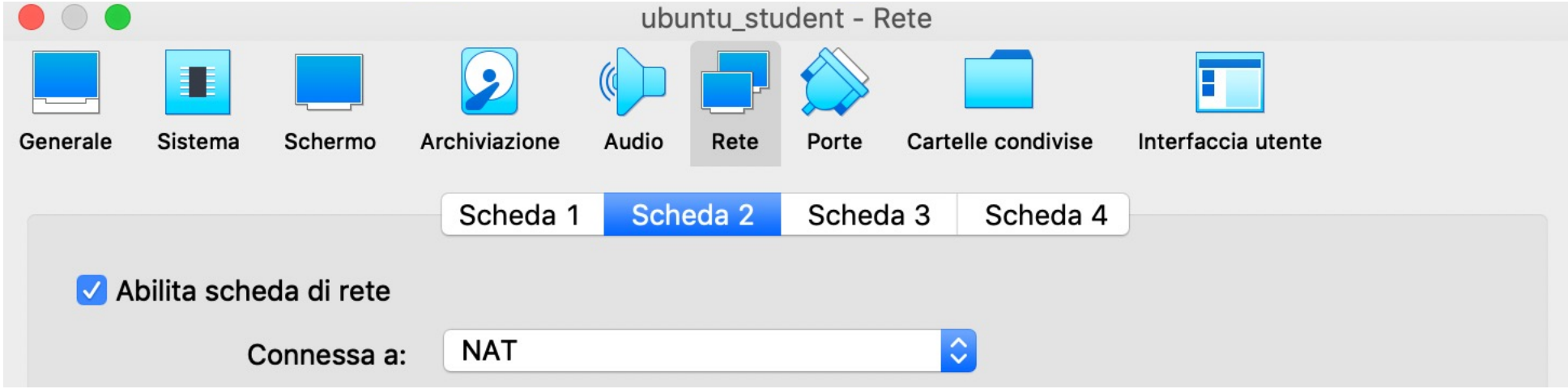


Setting up the environment (2)

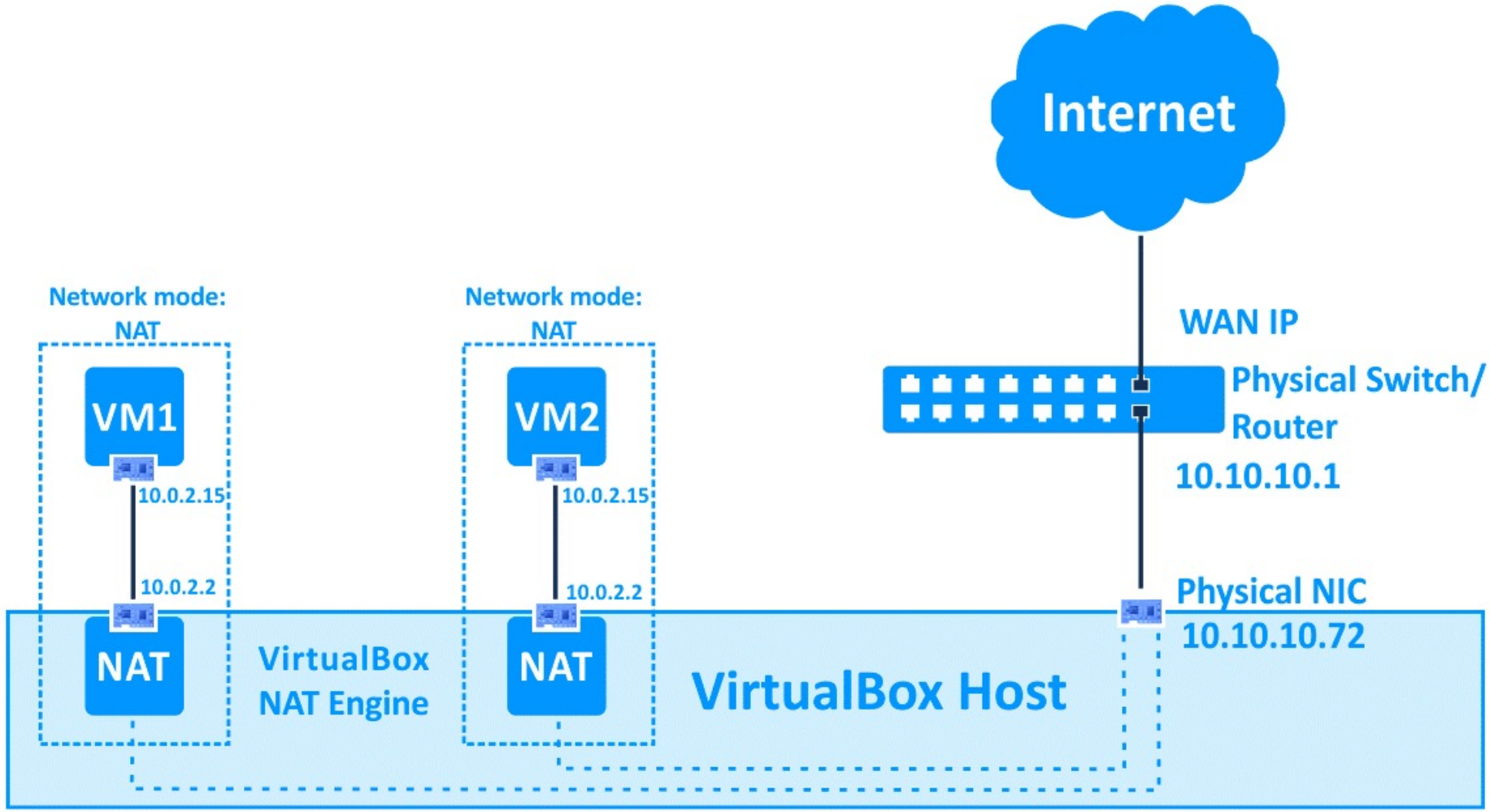
- Connect to Server 1 via SSH (*optional*, but will ease your work)
- Run ifconfig –a on both the host and the guest
- Verify the connectivity through pings between host and guest
- Repeat the process for student2

Setting up the environment (3)

- The current configuration does not allow accessing external networks
- One (quick) way to give internet access to your VMs is to configure a new interface and set it to work in **NAT** mode.



NAT mode


Internet
WAN IP
Physical Switch/
Router
10.10.10.1
Physical NIC
10.10.10.72
Network mode:
NAT
VM1
10.0.2.15
10.0.2.2
NAT
Network mode:
NAT
VM2
10.0.2.15
10.0.2.2
NAT
VirtualBox
NAT Engine
VirtualBox Host

Check connectivity (from guest)

- Ping 8.8.8.8

- run ifconfig –a

- You might need to explicitly query the DHCP server:

```
dhclient enp0s8
```

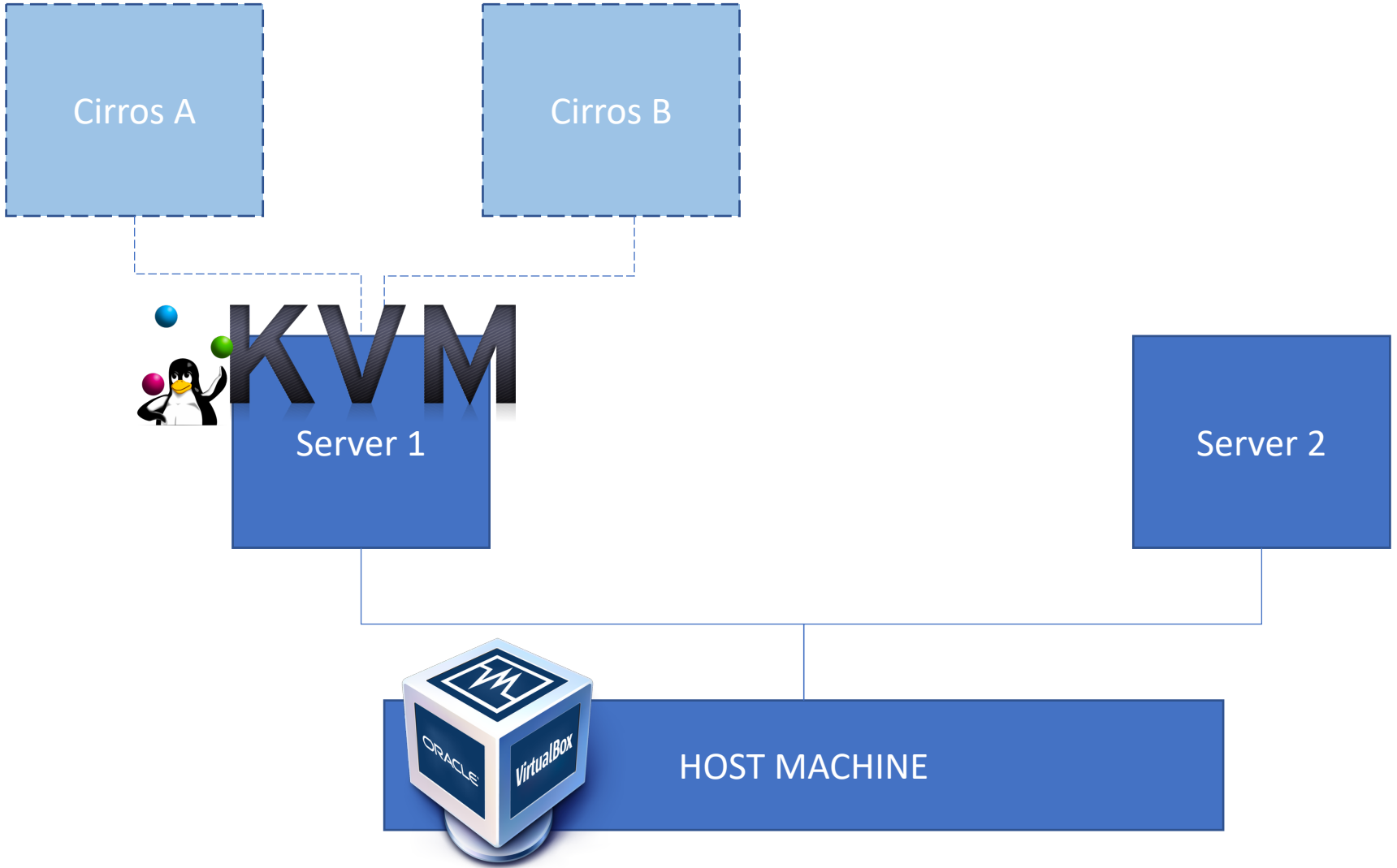
Is it NATed?

- Try a ping from student1 to 8.8.8.8
- Monitor the traffic through the “internet” interface of the host

```
tcpdump -i enp0s3 dst 8.8.8.8
```

- Do the same for NATed interface in student1

Lab configuration



Quick Recap: VM creation

Already done

```
sudo apt-get update


sudo apt-get install qemu-kvm libvirt-daemon-system libvirt-clients
bridge-utils virtinst
```

```
wget http://download.cirros-cloud.net/0.4.0/cirros-0.4.0-x86_64-disk.img
```

- `mv cirros-0.4.0-x86_64-disk.img /var/lib/libvirt/images/CirrosA.img`
- `cp /var/lib/libvirt/images/CirrosA.img /var/lib/libvirt/images/CirrosB.img`
- `sudo virt-install --name CirrosA --description 'CirrosA' --ram=512 --vcpus=1 --os-type=Linux --os-variant=ubuntu18.04 --disk path=/var/lib/libvirt/images/CirrosA.img,bus=virtio,format=raw --graphic none --import`

Quick Recap: useful commands [VM management]

```
virsh list                          // list running VMs
virsh console CirrosA               // connects to the VM console
virsh start CirrosA                 // starts/shutdown the VM
virsh shutdown CirrosA

virsh edit CirrosA                  // edit VM configuration

virsh destroy CirrosA               // remove VM
virsh undefine CirrosA
```

Quick Recap: useful commands [networking]

virsh net-create *VNET.xml* *// create and start a non-persistent network*
virsh net-define *VNET.xml* *// create only*

virsh net-start VNET
virsh net-destroy VNET
virsh net-undefine VNET

brctl show
virsh net-list // list available networks
virsh net-dumpxml [network_name]

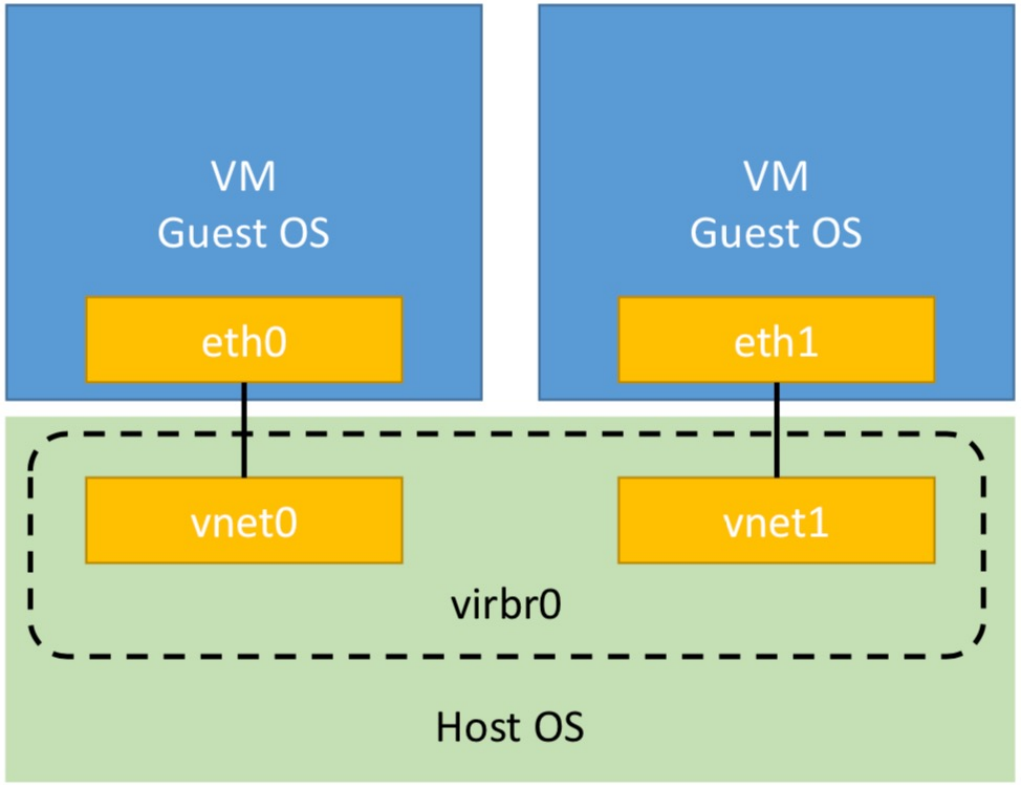
virsh domifaddr Cirros // list of interfaces of a running domain
virsh domiflist Cirros // brief info of all interfaces of a given domain
virsh domifstat Cirros vnet0 // statistics of a given interface

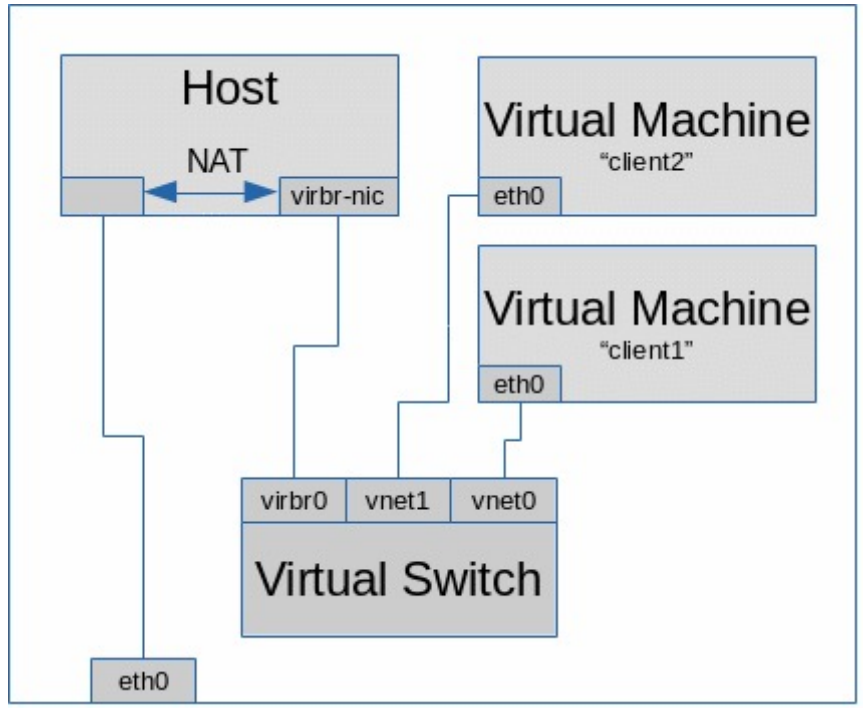
Custom network

- The VM-creation command we saw, connects the VM to the default “network”
- You can specify a custom network through the following option for the `virt-install` command:

```
--network network=custom
```

Linux-Kernel support (libvirt)





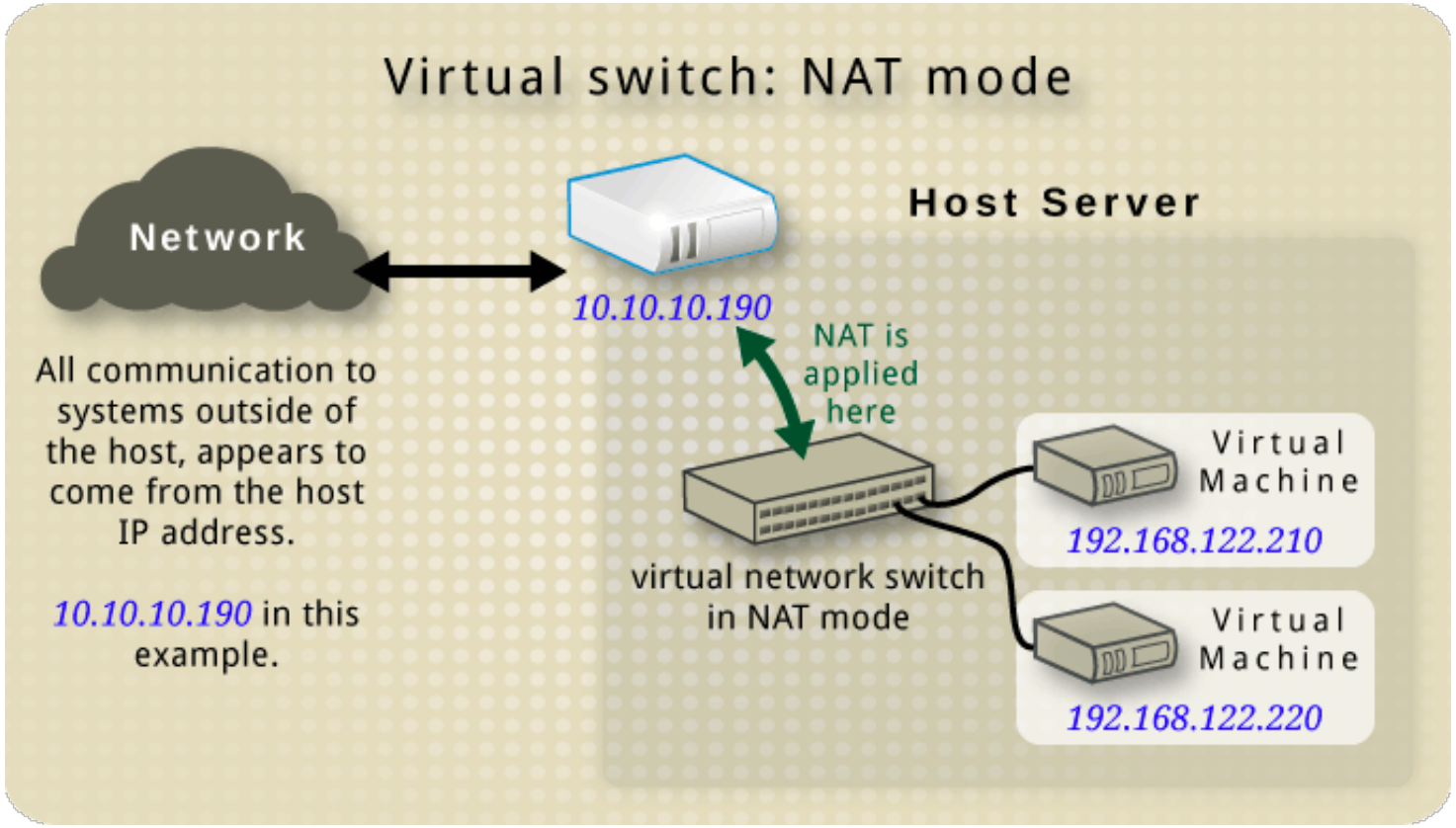
Quick Recap: network configuration

```
virsh net-dumpxml default
```

```
[student@odl_server:~$ virsh net-dumpxml default
<network connections='1'>
  <name>default</name>
  <uuid>bbca96ec-78a3-465c-ab36-13cfc1047c29</uuid>
  <forward mode='nat'>
    <nat>
      <port start='1024' end='65535'/>
    </nat>
  </forward>
  <bridge name='virbr0' stp='on' delay='0'/>
  <mac address='52:54:00:84:10:1f'/>
  <ip address='192.168.122.1' netmask='255.255.255.0'>
    <dhcp>
      <range start='192.168.122.2' end='192.168.122.254'/>
    </dhcp>
  </ip>
</network>
```

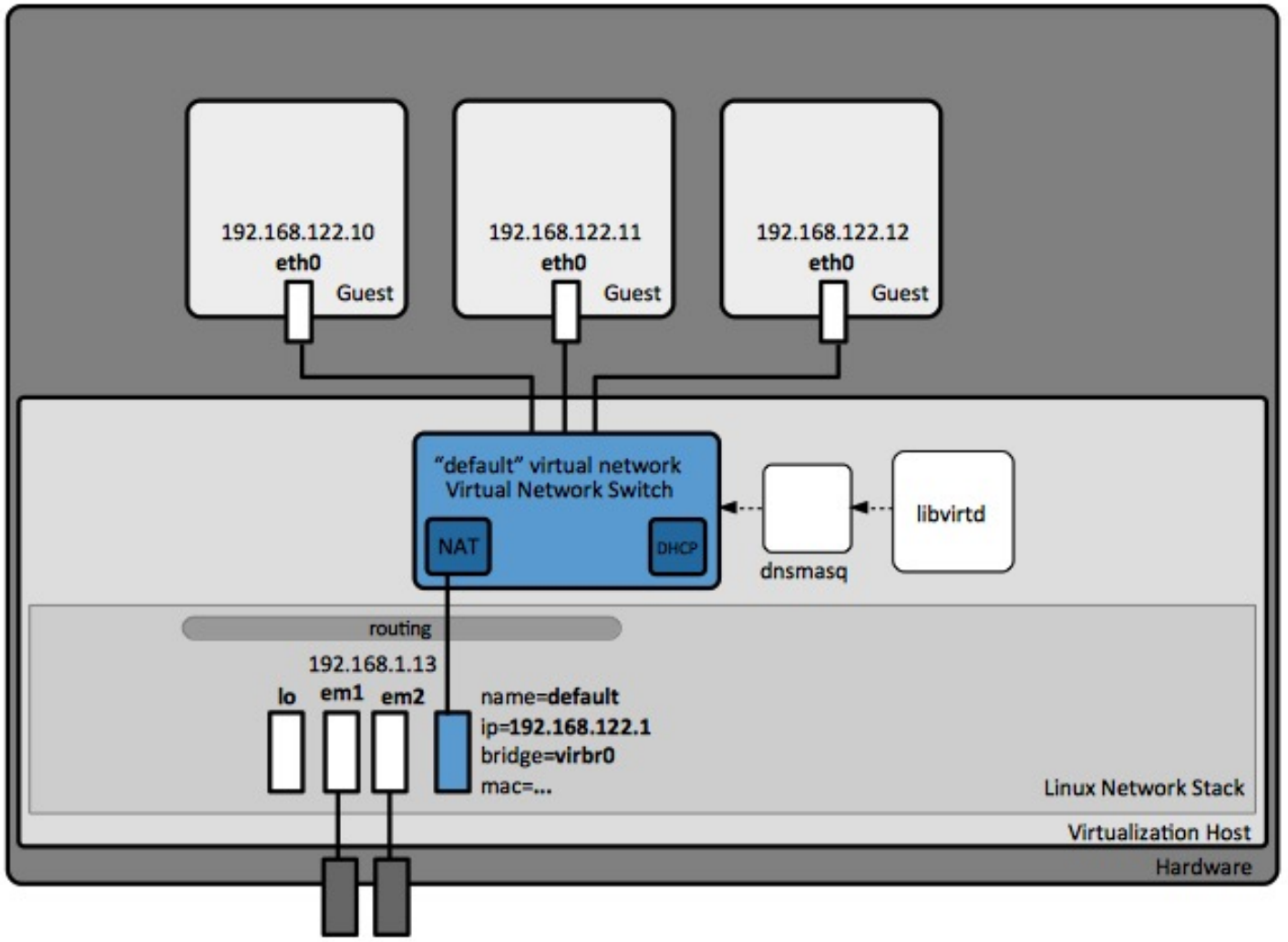
- Nat
- isolated
- Routed
- bridged

Network Modes: NAT



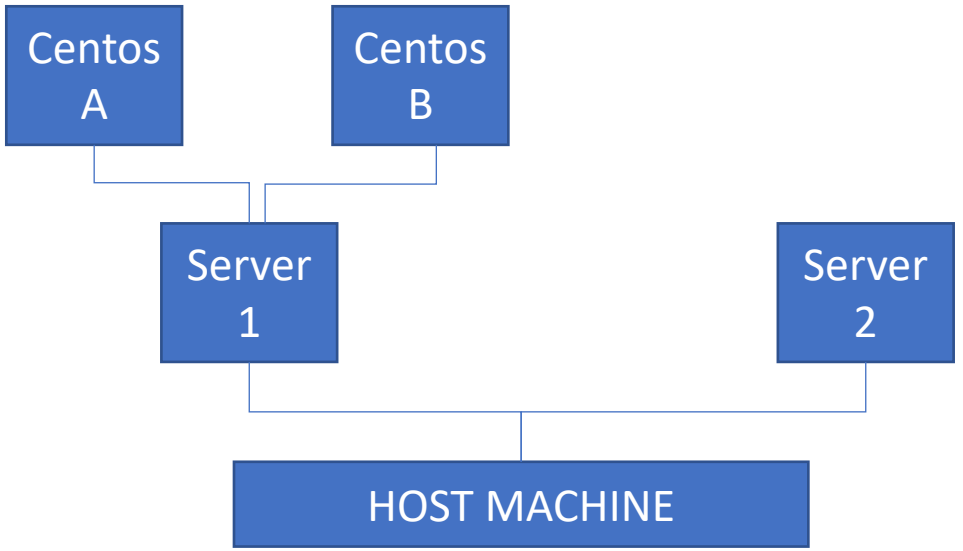
Network Modes: NAT

```
brctl
ifconfig -a
ip a s
```



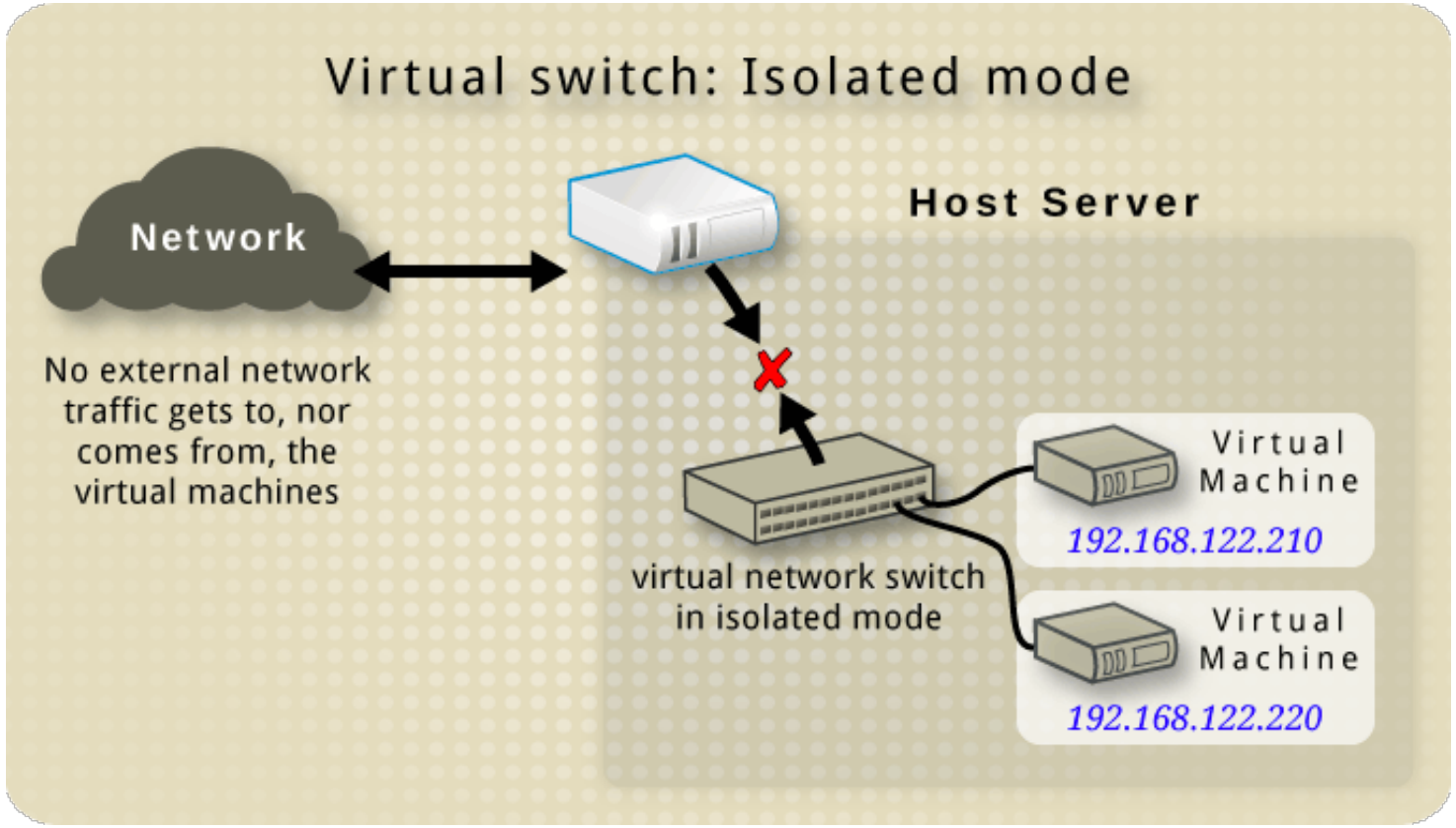
Test: NAT

1. Create VMs CentosA, CentosB
2. Verify IPs are properly set
3. Do the following test



- Ping: A <-> B
- Ping: A,B <-> Server 1
- Ping: A,B -> Server 2
- Ping: Server 2 -> A,B
- Check traffic through Server1

Network Modes: isolated



Test: isolated Mode

CHECK VM

- `virsh edit CirrosA`
- search for section:

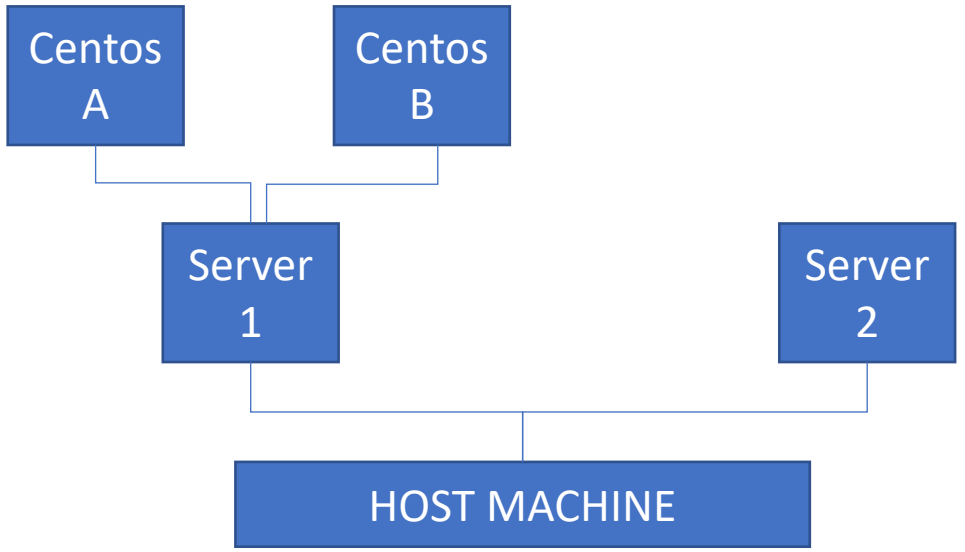
```
<interface type='network'>
      <source network='default'/>
      <mac address='00:16:3e:1a:b3:4a'/>
  </interface>
```

REMOVE NAT

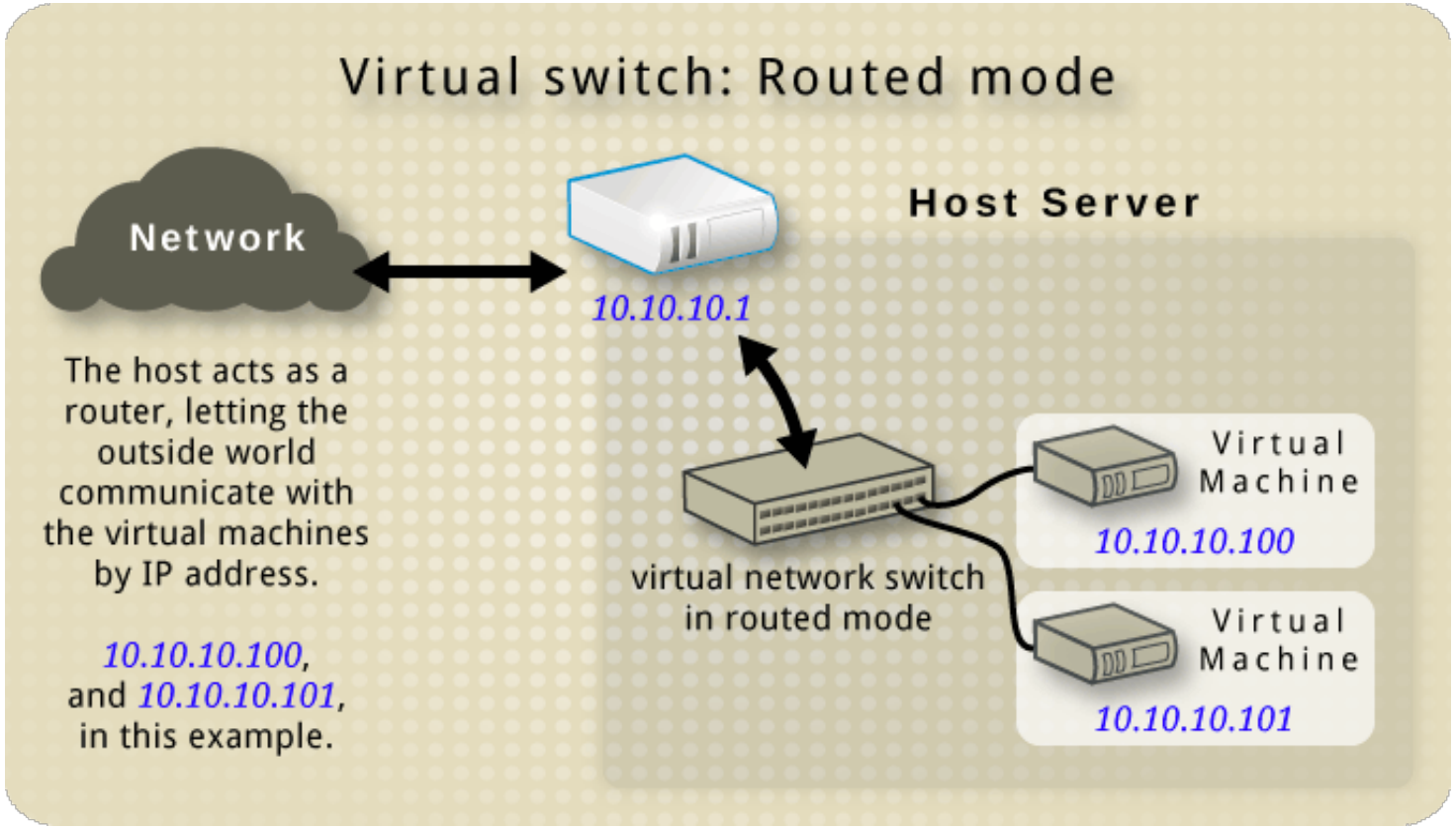
- `# sudo virsh net-edit default`
- //remove <forward .. section
- `# sudo virsh net-destroy default ; sudo virsh net-start default`
- `# service libvirtd restart`
- `# sudo virsh console vm_name`

Test: isolated Mode

- Ping: A <-> B
- Ping: A,B <-> Server 1
- Ping: A,B -> Server 2
- Ping: Server 2 -> A,B



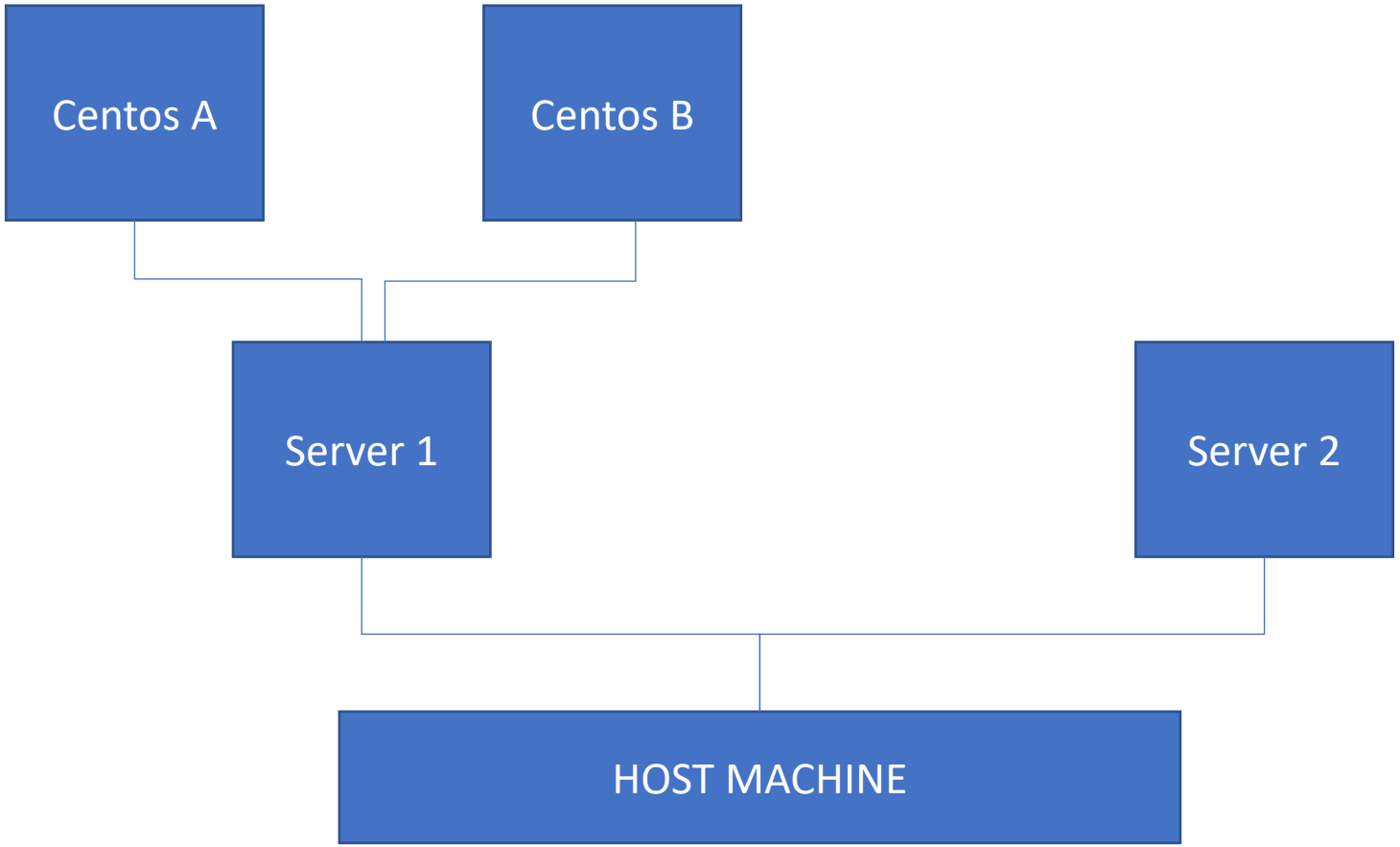
Network Modes: routed



Configure Route mode

- `# sudo virsh net-edit default`
- //ADD section <forward mode="route"/>
- `# sudo virsh net-destroy default`
- `# sudo virsh net-start default`
- `# service libvirtd restart`
- `# sudo virsh console CirrosA`

Does it work?



Configure a static route

At this point the VM can reach other networks but cannot ping, as there is no route for return packets. We should adjust routing at various points of the network. For the sake of this example, we can place a static route pointing to the host server, as follows.

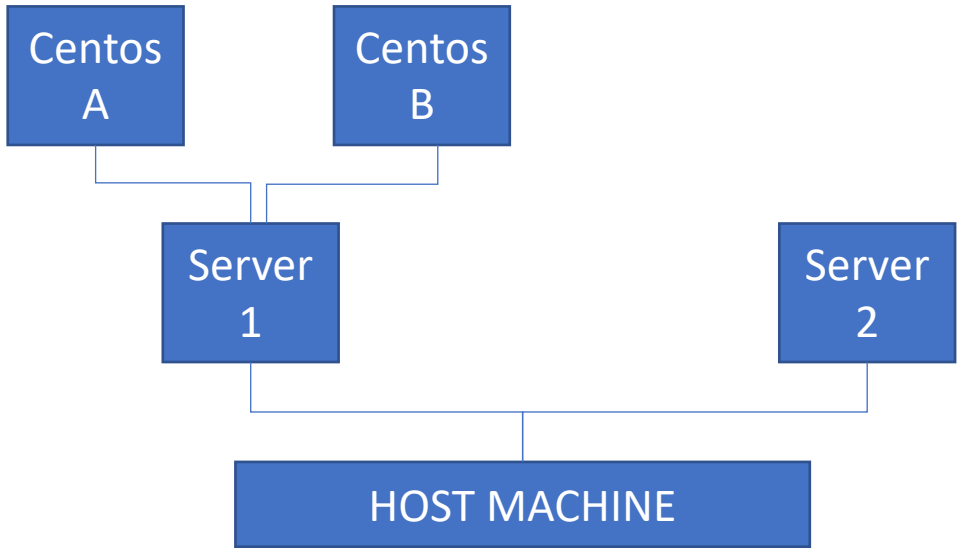
```
sudo ip -4 route add 192.168.122.0/24 via 192.168.56.3
```

You can browse the routes and delete any line using the following commands:

```
ip route list
sudo ip route del 192.168.122.0/24 via 192.168.56.3 dev eth0
```

Test: Routed

- Ping: A <-> B
- Ping: A,B <-> Server 1
- Ping: A,B -> Server 2
- Ping: Server 2 -> A,B



Exercise