OpenStack for Non-Developers

Building a personal cloud with OpenStack

January 15th, 2015

Tim Potter <tpot@hp.com>

OpenStack is a hot new technology.

You're a Hacker and a Maker.

Hacker Rabbit, CC BY-NC-SA, tumbersault@flickr

There's little information available on small clouds.

You can install OpenStack at home using existing well-supported deployment thingies.

You need to have a personal cloud!

openstack™
CLOUD SOFTWARE

+ You = Awesome

How to build a single-node OpenStack compute server at home

What are we doing and what do we need?

Tools, CC BY, zzpza@flickr

Hardware Requirements

- A spare PC with a reasonable amount of memory
- A console can be very handy

Networking requirements

- A single NIC
- An undisturbed IP address range on your home network
  - 192.168.1.0/24, DNS and gateway at 192.168.1.1
  - DHCP server hands out addresses from .16 to .127
  - OpenStack will use addresses from .128 to .254
  - OpenStack host will live at 192.168.1.2
- General knowledge about Linux networking
  - DNS, DHCP and ARP
  - IP routing

What we end up with

- Installed
  - Horizon for the web UI
  - Nova and Glance for VMs and images
  - Neutron for networking
  - Cinder for block storage
  - Keystone for authentication

- Not installed
  - Swift
  - Heat
  - Ceilometer

Install OpenStack using Chef and Stackforge repositories

Tools, CC BY, zzpza@flickr

Detailed instructions on my wiki!

https://github.com/tpot/os4nd/wiki

Install Base OS

- Install Ubuntu 12.04 LTS (Precise)
- Add cloud-archive repository

```
$ sudo apt-get install python-software-properties –y
$ sudo apt-add-repository -y cloud-archive:icehouse
$ sudo apt-get update && sudo apt-get upgrade –y
```

- Install updated kernel

```
$ sudo apt-get install linux-image-generic-lts-trusty –y
$ sudo reboot
```


Install Chef and OpenStack

- Install Chef

```
$ wget http://opscode-omnibus-packages.s3.amazonaws.com
    /ubuntu/12.04/chefdk_0.3.2-1_amd64.deb
$ sudo dpkg –I chefdk_0.3.2-1_amd64.deb
```

- Check out StackForge umbrella repository

```
$ git clone –b stable/icehouse https://github.com/stackforge/openstack-chef-repo
$ cd openstack-chef-repo
$ berks vendor cookbooks
```

- Create configuration file and run Chef

```
$ vi environments/allinone-physical-host.json
$ sudo chef-client –z –E allinone-physical-host –o ‘role[allinone-compute]’
```

Sample configuration file

```
{
  "name": "allinone-physical-host",
  "override_attributes": {
    "openstack": {
      "developer_mode": true,
      "compute": {
        "network": {
          "service_type": "neutron"
        },
        "endpoints": {
          "host": "192.168.1.2",
          "bind-host": "0.0.0.0",
        }
  ...
```

Verify OpenStack Install

```
$ export OS_AUTH_URL=http://127.0.0.1:35357/v2.0
$ export OS_TENANT_NAME=admin
$ export OS_USERNAME=admin
$ export OS_PASSWORD=admin

$ keystone catalog
```

Browse to https://192.168.1.2 and poke around in the web interface

```
$ ps auxww
```


Configure networking

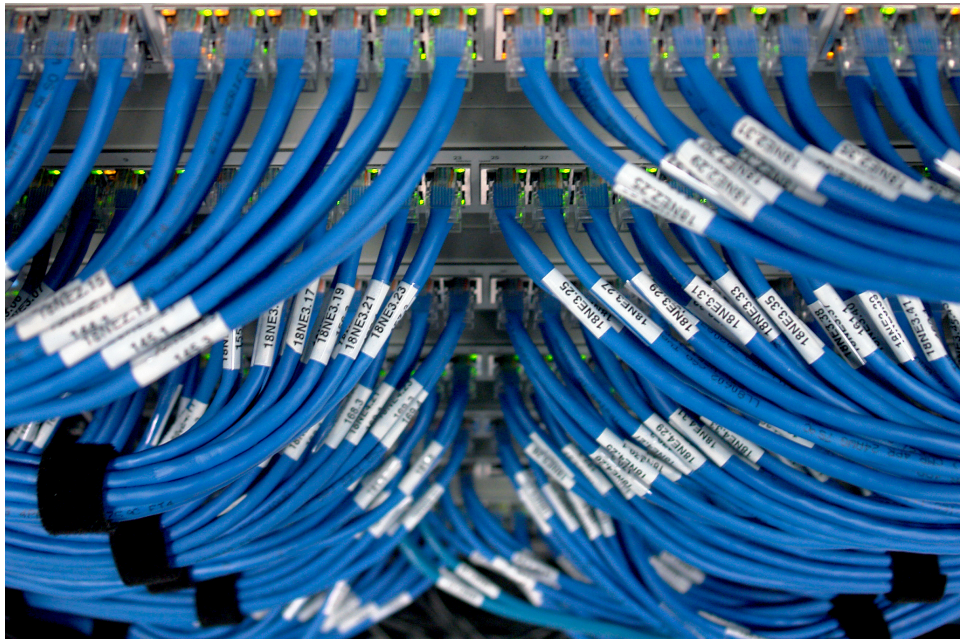
- Add eth0 to external bridge

```
$ sudo ovs-vsctl add-br br-ex
$ sudo ovs-vsctl add-port br-ex eth0
$ sudo ifconfig eth0 0.0.0.0
$ sudo ifconfig br-ex 192.168.1.2 netmask 255.255.255.0
```

- Create public network

```
$ neutron net-create public-net –shared –router:external=True
$ neutron subnet-create public-net 192.168.1.0/24 --name public-subnet \
    --allocation-pool=start=192.168.1.128,end=192.168.1.254 \
    --gateway 192.168.1.1 --enable_dhcp=False
```

Just enough networking

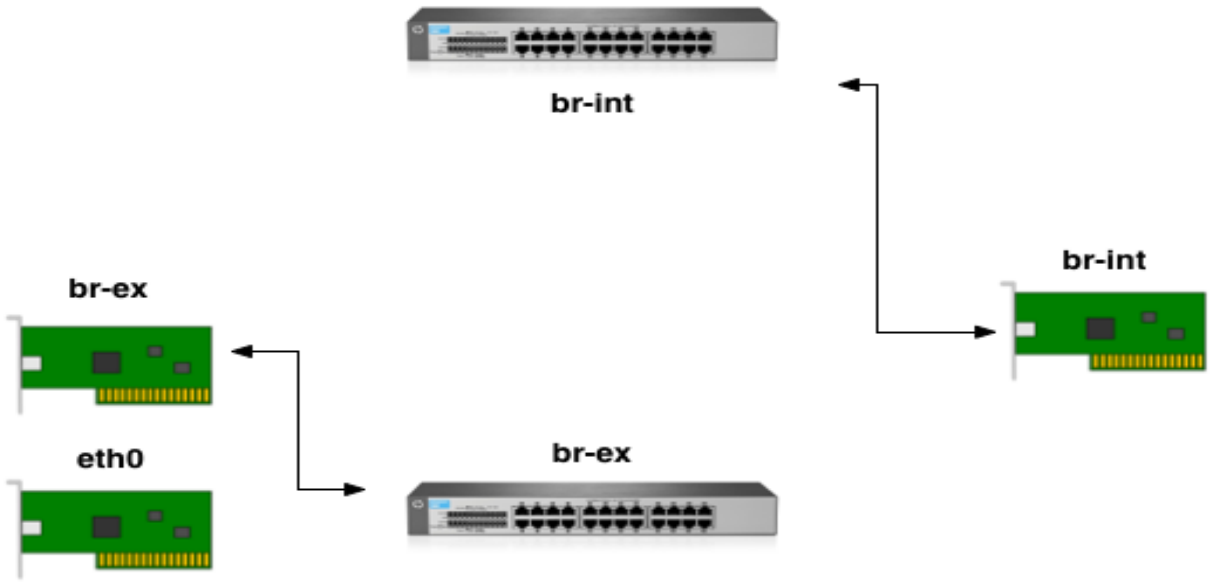


Switch!, CC BY-SA, andrewfhart@flickr

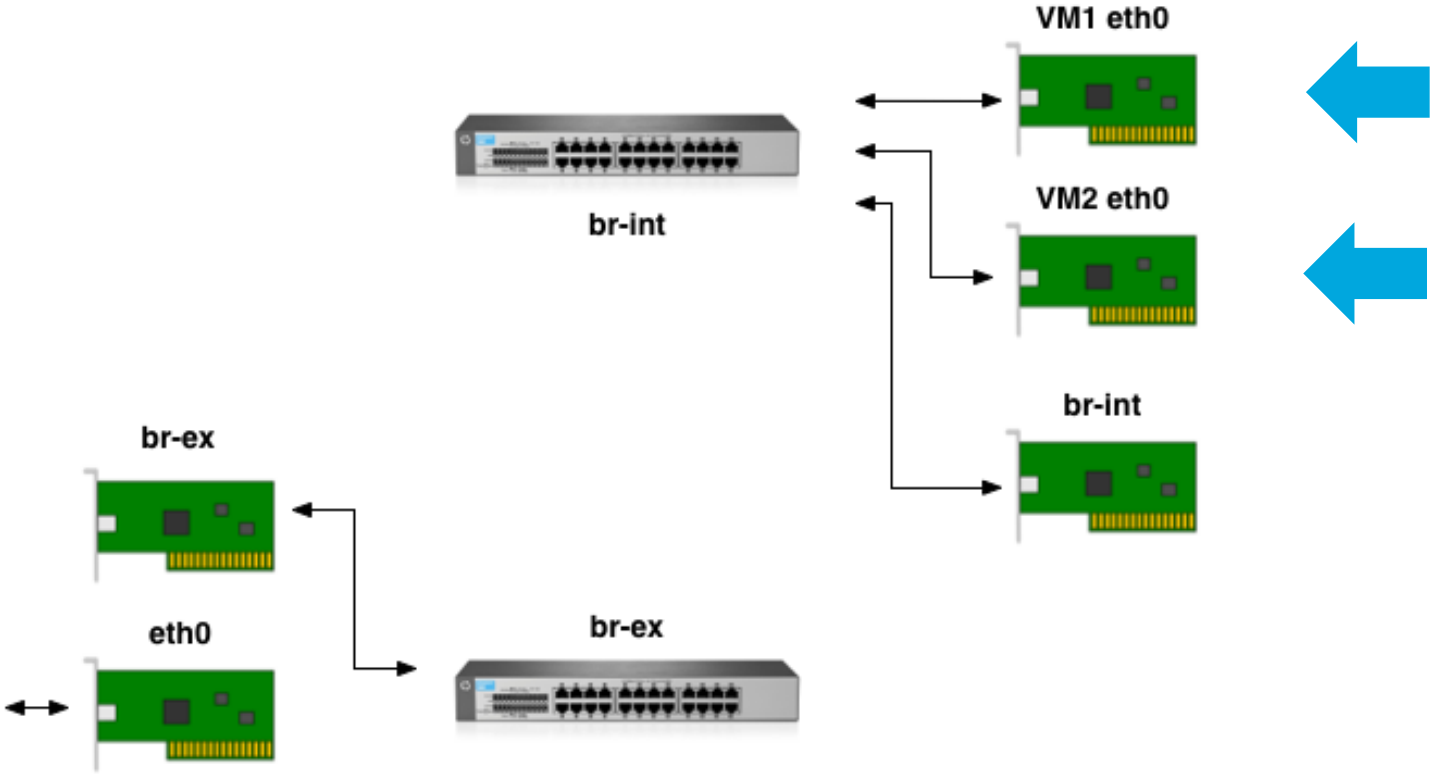
Open vSwitch

- An open source multi-layer virtual Ethernet switch
- Think of an Open vSwitch bridge just like a regular Ethernet switch
  - It has ports that you plug things in to
  - Broadcasts packets go to all ports
  - It learns which MAC address(es) are on which ports
- Our OpenStack install creates two virtual switches
  - `br-int` is the switch for OpenStack integration network
  - `br-ex` is the switch to access the external network

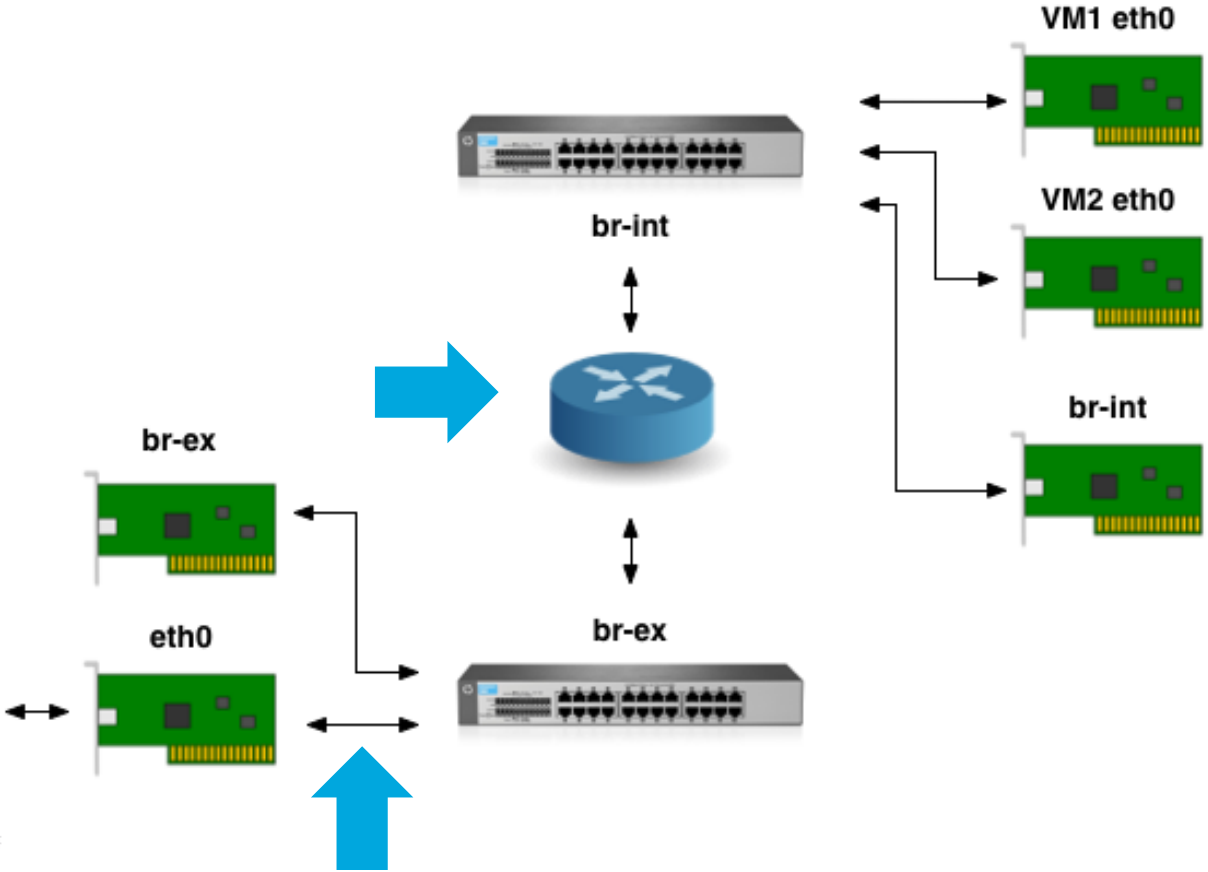
Networking



Networking with VMs



Networking with external access



Demonstration

Create and configure a user

Tools, CC BY, zzpza@flickr

Use web interface to create tenant and user

- As admin user, navigate to Admin tab, then Identity Panel
- Click on Projects tab, then Create Project button and create a tenant called “demo”
- Click on Users tab, then Create User button and create a user called “demo”
  - Set password to “demo”
  - Select “demo” as primary project
- Sign out as admin user

Use web interface to configure tenant network

- Log in as demo user
- Click on Network tab select Networks, then Create Network button to create tenant network
  - Name it “demo-net”
  - Create “demo-subnet” with address of `10.0.0.0/24`
  - Enable DHCP and set DNS servers to `8.8.8.8,8.8.4.4`

Use web interface to configure tenant router

- Select Routers tab then Create Router to create tenant router
  - Name it “demo-router”
  - Set gateway to “public-net”

- Click on “demo-router” and then Add Interface button
  - Select “demo-subnet” as the subnet for the interface

- That now links the tenant network to the external network
  - demo-net to public-net
  - demo-subnet to public-subnet
  - Layer 3 routing occurs inside demo-router

Create and test a VM

Tools, CC BY, j@flickr

Create VM using web interface

- Go to Project, Compute tab select Instances and click Launch Instance button
- Create VM named “demo-vm”
- Instance boot source “Boot from images”
- Image name “cirros”
- Click networking tab and select “demo-net” network
- Click launch button

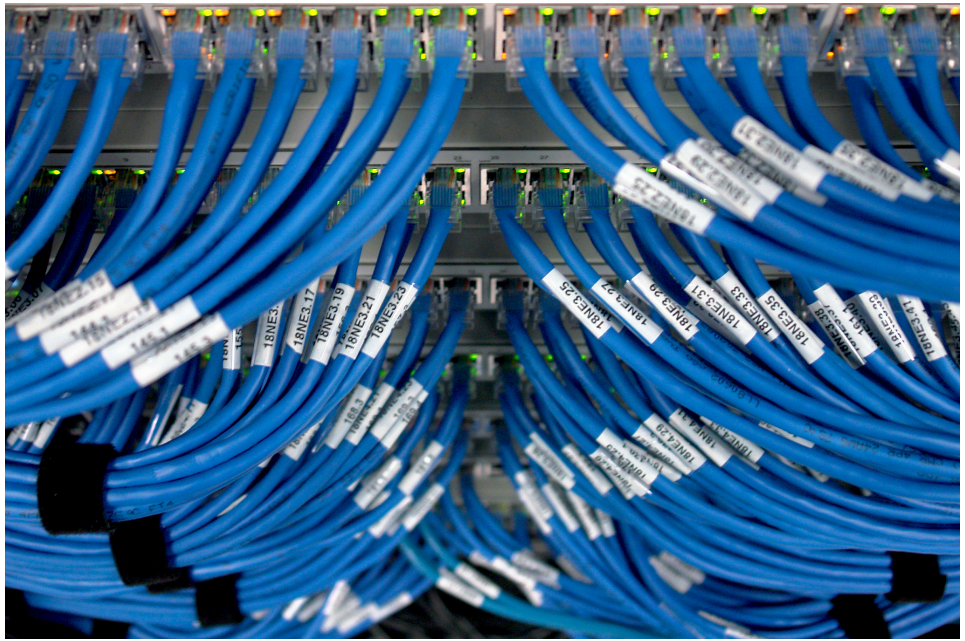
Test logging in and networking

- Use instance console to check VM has booted and log in
  - For Cirros username is “cirros”, password “cubswin:)”
- Should have NAT access to the internet via demo-router
- VMs by the same tenant should see each other at layer 2
  - Iff they are on the same network
- VMs of different tenants are isolated and cannot communicate at all

Testing floating IPs

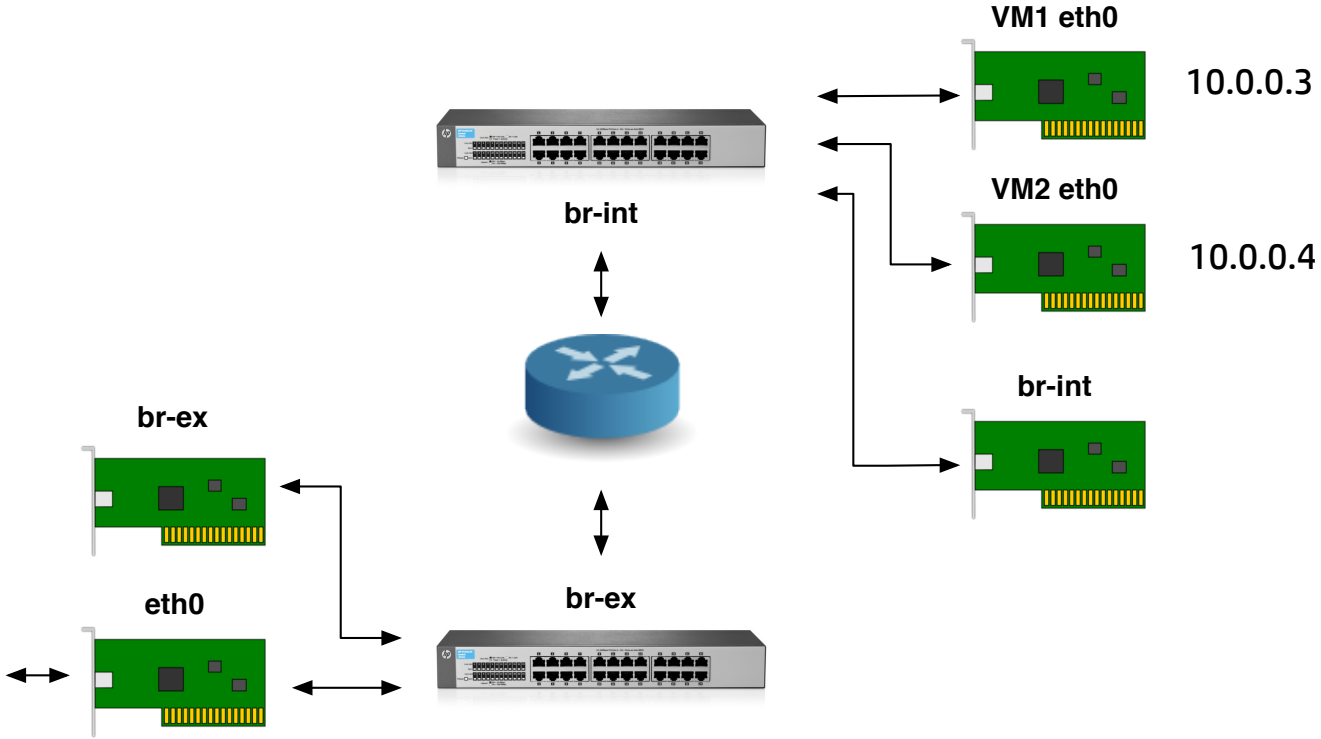
- Go to instance tab and select “Associate Floating IP” from the drop down menu
- Choose an IP address from the drop-down, or click the “+” button to allocate one from the public-net allocation pool
- Test public IP address can access the VM

Networking review

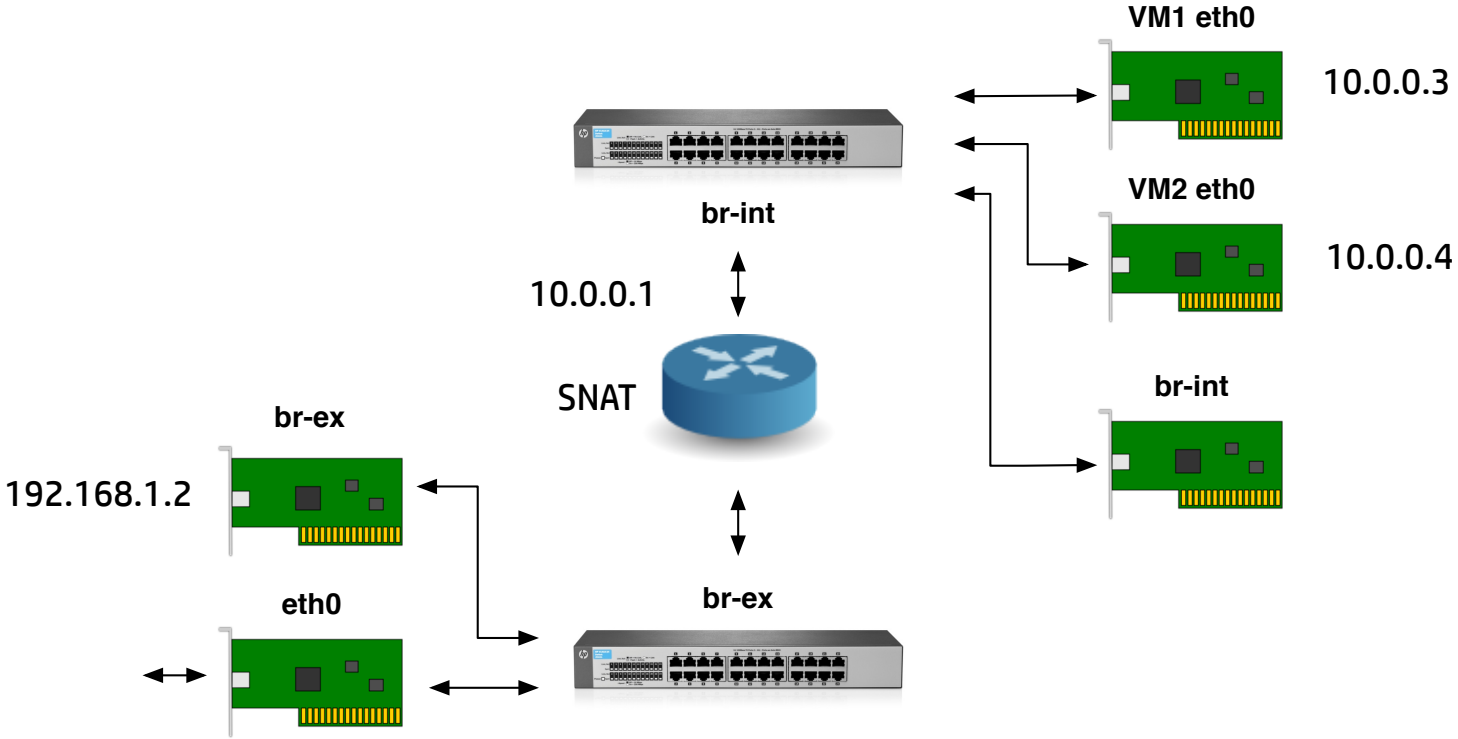


Switch!, CC BY-SA, andrewfhart@flickr

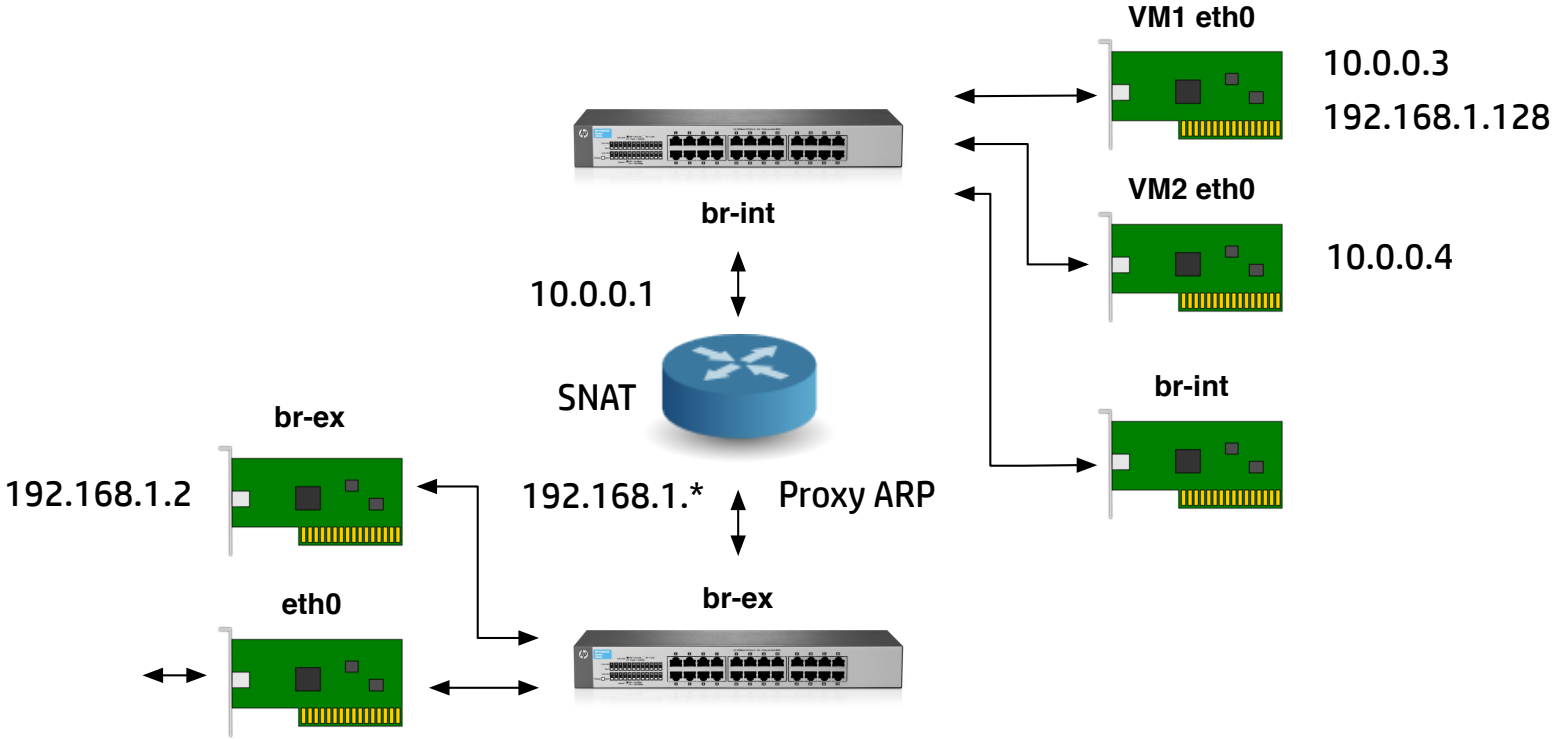
Inter-VM packet flow



Intra-VM packet flow

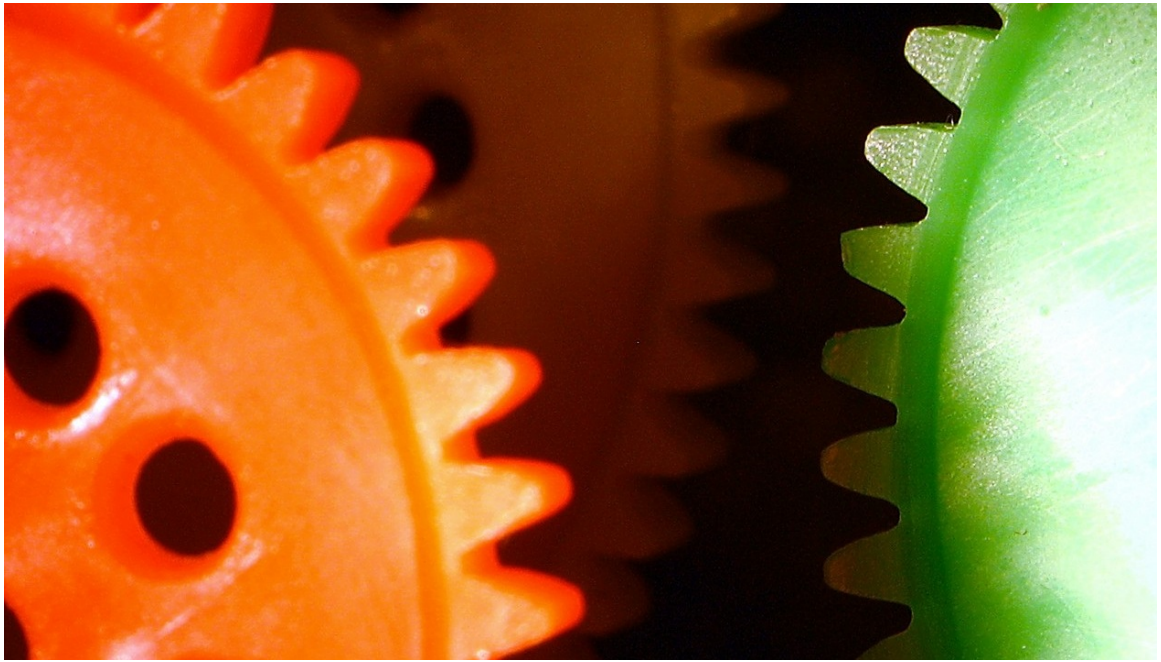


Public IP packet flow



Miscellaneous Tips & Tricks

Operating your cloud



Drawing gears..., CC BY-NC-ND, Heartlover1717@flickr

OpenStack is a collection of REST interfaces

Service	Function	Resource
Nova	Compute	libvirt, kvm, qemu
Keystone	Authentication	MySQL, LDAP, PAM, other
Cinder	Block storage	LVM2
Glance	Image management	Disk files, other
Neutron	Networking	iptables, Open vSwitch

Troubleshooting

- A lot of problems can be fixed with standard sysadmin skills
  - `/var/log/$SERVICE/$SERVICE.log`
  - `/etc/$SERVICE/$SERVICE.conf`
  - `http://stackoverflow.com/search?q=$SERVICE`
- Troubleshoot the underlying resource
- Various networking tricks can be used to troubleshoot networking problems
  - `ip netns`
  - `ovs-osctl, ovs-ofctl`

Tweaking config things with Chef

- Process for tweaking configuration files
  - Identify which config file you want to change, e.g `nova.conf`
  - Find template file in Chef cookbooks, e.g `nova.conf.erb`
  - Add JSON attribute in environment file and re-run Chef
- Example, use qemu virtualisation instead of kvm
  - Change `virt_type` from `kvm` to `qemu in nova.conf.erb`
  - Attribute is `node["openstack"]["compute"]["libvirt"]["virt_type"]`
  - Re-run Chef to effect the change

New config file

```
{
  "name": "allinone-physical-host",
  "override_attributes": {
    "openstack": {
      "developer_mode": true,
      "compute": {
        "libvirt": {
          "virt_type": "qemu"
        },
        "network": {
          "service_type": "neutron"
        }
   ...
}
```


Customisations

This Old House, CC BY-SA, Ed Schipul@flickr

Add useful VM images

- Download QCOW2 disk images from vendor page
  - `https://cloud-images.ubuntu.com`
  - `https://openstack.redhat.com/Image_resources`
- Upload to glance server as admin

```
$ glance image-create --name $IMAGE_NAME --is-public=true \
    --container-format=bare --disk-format=qcow2 \
    --file=$IMAGE.qcow2
```


Configure Block Storage

- Default driver for the Cinder block storage server is LVM2, using a volume group called `cinder-volumes`

```
$ sudo pvcreate /dev/sdb
$ sudo vgcreate cinder-volumes /dev/sdb
$ sudo service cinder-volume restart
```

- Block storage gives us persistent storage for VMs
  - Snapshots
  - Boot from volume

Multi-node setup

- Not considered in this presentation, but you're welcome to try it out (-:
  - Make one node a controller node
  - Apply compute-only Chef roles to compute nodes
- Multi-node networking is more complicated
  - `br-tun` virtual switch creates a mesh for compute nodes to communicate with each other
  - Bridges together the `br-int` switches on each node

Conclusion

Hacker Rabbit, CC BY-NC-SA, tumbersault@flickr

openstack™
CLOUD SOFTWARE

+ You = Awesome

Thank you!

tpot@hp.com

https://github.com/tpot/os4nd/wiki