Virtualisation Intro

By Laban Mwangi (lmwangi _at_ gmail.com)
Based on notes by Joel Jaeggli
For AFNOG SS-E 2014
What is it?

• An abstraction that allows for easy subdivision and allocation of resources

• What Computing/Network resources can be virtualized?
  • OS virtualization
  • Application virtualisation
  • Service virtualisation
  • Network virtualisation
  • Storage virtualisation
  • And much more...
Anything?

- In the context of this course, we're interested in virtualization along two dimensions:
  - Resource virtualisation
  - OS virtualisation
Resource/Service virtualization

- Examples:
  - Load-balancers
  - DNS Based GLB
  - HTTP(S) Virtual Hosting
  - MX records
  - Virtual Switches
  - Virtual Routers
  - Virtual Firewalls
Resource Virtualization - Continued

- HTTP virtual hosts
  - Multiple websites on one system
- Load Balancing
  - One (or many sites or applications) across many systems
  - Can be done at Layer-3/4/7
Host Virtualization

- Examples
  - VMware
  - Virtual-Box (used in class)
  - KVM
  - XEN
  - FreeBSD and Linux Jails
  - Windows Hyper-V
What problem are we attempting to solve with host virtualization.

• Problem 1 – Idle capacity.
  • Most of the machines in your datacenter are idle most of the time.
  • Capacity you're not using:
    – Cost money up front
    – Cost money to operate
    – Reduces your return on capital
  • Packing discrete systems into a smaller number of servers provides savings along virtually every dimension.
Problems - Continued

• Problem 2 – Provisioning

  • Spinning up a new service involves:
    – Acquiring the hardware
    – Building the server
    – Integration with existing services

  • With virtualization we're aiming to short-circuit that
    – Capacity is a resource
    – Machine instances may be cloned or provisioned from common basic images
    – Resources are purchased in bulk and assigned to applications as necessary.
• Problem 3 – Hardware abstraction
  • Operating systems, servers, and applications evolve at different rates.
  • Providing a common set of infrastructure resources means, virtualized systems are portable across servers
  • Hardware failure can more easily be managed.
• Abstraction may come at a performance cost however. (some workloads are more expensive than others)
Examples – Desktop Virtualization
Desktop Virtualization

- **Uses**
  - Prototyping services or applications before deployment
  - Utilities that don't run on your operating system
  - Isolation of sandbox environments from your desktop
  - Maintaining multiple versions of an environment for support purposes.
  - Staying familiar with unix while running windows (consider compared to the alternative (dual-booting))

- **Issues**
  - Emulating multiple computers on your laptop/desktop is somewhat resource intensive

- **Vmware player and VirtualBox are free.**
  - [http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0?ie=UTF-8](http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0?ie=UTF-8)
Examples – Server Virtualization
Server Virtualization - Continued
Server Virtualization
Virtualized Servers as a Service (Amazon Web Services)

- Much as colocated servers, are available from a hosting provider, virtual servers are also available.

- Model is:
  - You pay for what you use.
  - Flexibility, need fewer servers today then you used, yesterday.
  - Leverage other amazon tools (storage/map-reduce/load-balancing/payments etc)
AWS
AWS Steps

- Select availability zone
- Launch new instance
- Select appropriate ami
- Associate with ssh key
- Launch instance
- Add ip
- SSH into new machine instance.
- t1-micro-instances run $54 a year + bandwidth
Try it for free...

- Free tier for the first Calendar year is (per month):
  - 750 hours of EC2 running Linux/Unix Micro instance usage
  - 750 hours of Elastic Load Balancing plus 15 GB data processing
  - 10 GB of Amazon Elastic Block Storage (EBS) plus 1 million IOs, 1 GB snapshot storage, 10,000 snapshot Get Requests and 1,000 snapshot Put Requests
  - 15 GB of bandwidth in and 15 GB of bandwidth out aggregated across all AWS services
- Which is not to say that, at scale EC2 is particularly cheap, (It isn't)
  - Limited capital at risk is in the context of prototyping or experimentation however.
AWS - Continued

• For provisioning purposes cli interaction is possible:
  • http://aws.amazon.com/developertools/351
• Along with tools to support the provisioning and destruction of virtual machines.
Provisioning and management

- Is the glue that makes virtualization usable
- In commercial virtualization environments the provisioning/management toolkits represent the bulk of the licensing cost (VMware) and the secret sauce (VMotion, disaster recovery, backup, etc)

Examples:

- XEN tools – a collection of perl scripts for spinning VMs [http://www.xen-tools.org/software/xen-tools/]
- KVM tools - [http://www.linux-kvm.org/page/Management_Tools]
- Cloud.com/cloud-stack (orchestration) - [http://www.cloudstack.org/]
- Rightscale – (orchestration multiple public/private clouds) [http://www.rightscale.com]
- Puppet (host / configuration management) - [http://puppetlabs.com/puppet/]
- PDSH – (Parallel Shell execution) [http://code.google.com/p/pdsh/]
- Salt/chef/ansible/parallel-ssh/libvirt....
Virtualbox Exercise

• Download virtualbox and install it.
• Download the openbsd install iso.
• Create an openbsd 32bit virtual machine.
• Adjust the virtual machine settings to boot off the iso.
• Install and accept the defaults at the prompts to install OpenBSD.
• After installation, type reboot and then intercept the reboot by shutting down the vm at the bios stage
• Change boot order in the VM settings (CD last)
• Boot into VM
Name and operating system

Please choose a descriptive name for the new virtual machine and select the type of operating system you intend to install on it. The name you choose will be used throughout VirtualBox to identify this machine.

Name: openbsd
Type: BSD
Version: OpenBSD (32 bit)
Memory size

Select the amount of memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended memory size is 64 MB.

4 MB  

256 MB  

3072 MB

Go Back  
Continue
• Choose interactive install (I)
• Choose defaults by pressing <enter> on the prompts below
• Note that we configured the default interface to acquire an address via DHCP
• Elect not to run X windows

• Elect to enable ssh

• Type in a root password (afnog). You will need this password to log in once the installation is complete.
- Auto layout and auto partition...
- Note the resource consumption for OpenBSD is extremely low!
- Ignore the SHA256 signature verification prompt!
- Image below should give you an idea...

```
Let's install the sets!
Location of sets? (cd disk ftp http or 'done') [cd]
Available CD-ROMs are: cd0.
Which CD-ROM contains the install media? (or 'done') [cd0]
Pathname to the sets? (or 'done') [5.5/i386]

Select sets by entering a set name, a file name pattern or 'all'. De-select
sets by prepending a '-' to the set name, file name pattern or 'all'. Selected
sets are labelled '[X]'.

[X] bsd     [X] etc55.tgz  [X] xbase55.tgz  [X] xserv55.tgz
[X] bsd.rd  [X] comp55.tgz [X] xetc55.tgz
[ ] bsd.mp  [X] man55.tgz  [X] xshare55.tgz
[X] base55.tgz [X] game55.tgz  [X] xfont55.tgz

Set name(s)? (or 'abort' or 'done') [done]
Directory does not contain SHA256.sig. Continue without verification? [no] yes
```
- Wait for the installation to complete.
- Reboot the instance
- Intercept the reboot at the bios stage and shutdown the instance.
- Stop the instance and adjust the settings of the VM
- Make sure that CD/DVD comes after Hard disk as shown below
- Restart the instance
dev/wd0a (e508ca3ad951b0d5.a): file system is clean; not checking
dev/wd0e (e508ca3ad951b0d5.e): file system is clean; not checking
dev/wd0d (e508ca3ad951b0d5.d): file system is clean; not checking
setting tty flags
f enabled
tarting network
HCPREQUEST on em0 to 255.255.255.255 port 67
HCPACK from 10.0.2.2 (52:54:00:12:35:02)
ound to 10.0.2.15 -- renewal in 43199 seconds.
tarting early daemons: syslogd pflogd.
tarting RPC daemons:.
avecore: no core dump
hecking quotas: done.
earning /tmp
tarting pre-securelevel daemons:.
etting kernel security level: kern.securelevel: 0 -> 1
reating runtime link editor directory cache.
reserving editor files.
tarting network daemons: sshd sendmail sndiod.
tarting local daemons: cron.
ue May 27 10:49:11 EDT 2014

penBSD/i386 (mine.mtg.afnog.org) (ttyC0)
ogin: _
Class Exercise

- Good engineers are lazy, lazy, lazy, lazy!
- *vagrant* profiter de la vie!!
Vagrant

• Install vagrant from the local server on your laptop
• Make a vagrant file by issuing vagrant init
• Edit the vagrantfile
• Change the line:
  • From: `config.vm.box = "base"
  • To: `config.vm.box = "http://mini1.sse.ws.afnog.org/~inst/vagrant_boxes/hashicorp/precise32/version/1/provider/virtualbox.box"`
- Create a new dir and change to it
- Run vagrant init
- Run ee Vagrantfile to edit the generated file

```bash
$ vagrant init
```

- Change the line `config.vm.box` to the one below

```bash
  config.vm.box = "http://mini1.sse.ws.afnog.org/~inst/vagrant_boxes/hashicorp/precise32/version/1/provider/virtualbox.box"
```

# Every Vagrant virtual environment requires a box to build off of.

# Disable automatic box update checking. If you disable this, then
t# boxes will only be checked for updates when the user runs...
Run `vagrant up` to start the instance
• Run `vagrant ssh` to connect to the instance

```
$ vagrant ssh
Welcome to Ubuntu 12.04 LTS (GNU/Linux 3.2.0-23-generic-pae i686)

Documentation: https://help.ubuntu.com/
Welcome to your Vagrant-built virtual machine.
```

```
Last login: Fri Sep 14 06:22:31 2012 from 10.0.2.2
vagrant@precise32:~$ uname
Linux
vagrant@precise32:~$ uptime
12:06:58 up 2 min, 1 user, load average: 0.72, 0.79, 0.34
vagrant@precise32:~$
```

• You just provisioned an instance in minutes

• You can now configure this instance for a service

• You can change to another directory and repeat the process to get another instance

  • Run: `vagrant init`
  • Edit the file: `Vagrantfile`
  • Run: `vagrant up`

• Exercise: Create a new FreeBSD vm using this box:
Other vagrant commands

- List instances: `vagrant global-status`
- Stop an instance: `vagrant halt`
- Suspend an instance: `vagrant suspend`
- Resume an instance: `vagrant resume`
- Terminate an instance: `vagrant destroy`
- Snapshot and store/share an instance: `vagrant package`