AP/ITEC 2210 3.0 A: System Administration Fall 2023

Instructor: Jamon Camisso
ITEC 2210 Chat: mattermost.itec2210.ca
Email: jamon@yorku.ca
Website: https://eclass.yorku.ca/

Date/Time: Wednesday, 19:00-22:00
Location: Zoom / ACE 003
Office hours: Via Mattermost any time

Lecture 3 - Virtualization, Containers

- Notes from last week:
  - Everyone should have a VM: 12345678.itec2210.ca
    - Username is ‘itec2210’
    - Password is ‘limoncelli’ - to change it:
      - Login, type ‘passwd’
  - Use Terminal on OSX or in Windows 10 terminal or Powershell prompt:
    - `ssh itec2210@12345678.itec2210.ca`

Lecture 3 - Virtualization, Containers

- Readings:
  - PSNA 13, section 13.3 (p227-235)
  - Chapter 3, *Selecting a Service Platform* from:

    Limoncelli, T., Chalup, S. R., Hogan, C. J., & Limoncelli, T. (2015). *The practice of cloud system administration: designing and operating large distributed systems.* Upper Saddle River, NJ: Addison-Wesley.

  - See eClass for link to chapter

Lecture 3 - Virtualization, Containers

Lecture 3 - Virtualization, Containers

- First some terms you'll encounter:
  - Cloud
    - Someone else's computers (usually) with an API
      - Company/organization internal and private clouds also count
    - PCSA chapter makes 3 distinctions :

Lecture 3 - Virtualization, Containers

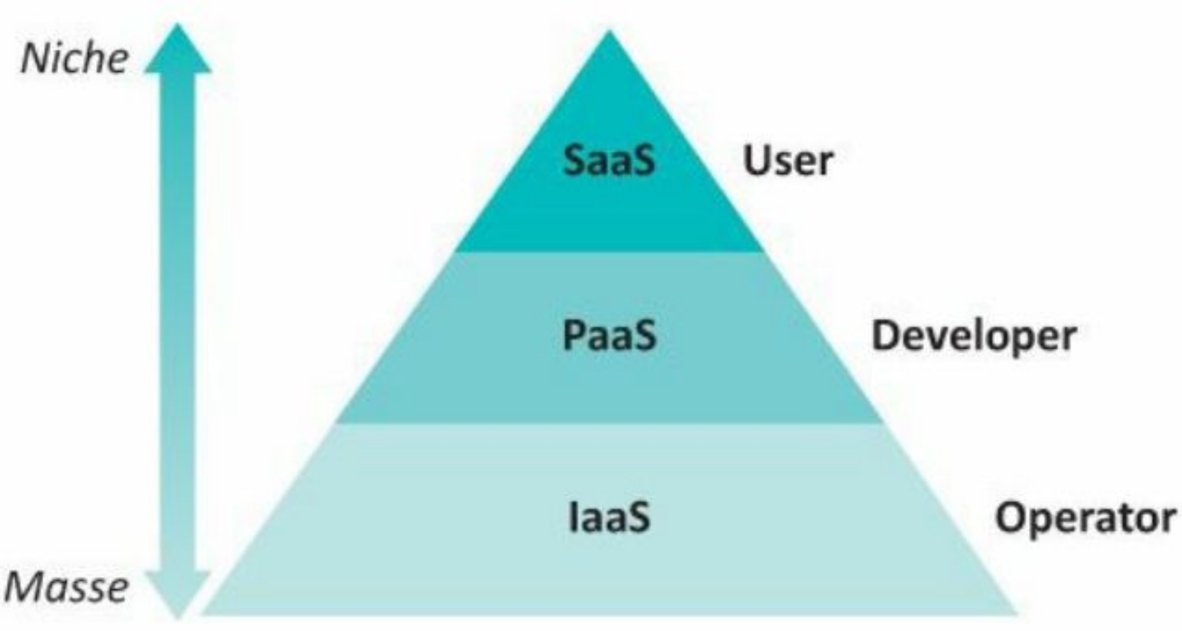
- First some terms you’ll encounter:
    - Cloud
        - Infrastructure as a Service (IaaS)
        - Platform as a Service (PaaS)
        - Software as a Service (SaaS)

Lecture 3 - Virtualization, Containers

- First some terms you'll encounter:
  - Cloud
  - **The NIST Definition of Cloud Computing**
    - Essential Characteristics
      - On-demand self-service
      - Broad network access
      - Resource Pooling
      - Rapid elasticity
      - Measured service

Lecture 3 - Virtualization, Containers

- Cloud



Lecture 3 - Virtualization, Containers

- Cloud

- Infrastructure as a Service (IaaS)

  ○ Physical or virtual compute resources that you can use as you see fit. Vendor provides infrastructure (CPU, RAM, Disk, Network, API) you do the rest yourself

  ○ AWS, GCE, Azure, Oracle, Rackspace Cloud, Linode, DigitalOcean, Softlayer are all IaaS

Lecture 3 - Virtualization, Containers

- Cloud

- Infrastructure as a Service (IaaS)

    - You’ll be charged for CPU, disk usage, and network ingress and egress on hourly, monthly, or per GiB basis

    - It is important to know how any application you deploy to an IaaS provider works
        - Does it use local storage or remote?
        - Can it deal with variable performance?

Lecture 3 - Virtualization, Containers

- Cloud

- Infrastructure as a Service (IaaS)
  - Most IaaS providers will have multiple regions, and zones within each

  - You may need to design your application to be aware of these and to be fault-tolerant should one go down

  - http://status.aws.amazon.com
  - https://azure.microsoft.com/en-gb/status/history/
  - https://status.cloud.google.com/incidents.json

Lecture 3 - Virtualization, Containers

- Cloud
- Infrastructure as a Service (IaaS)
  - Some provide VPN endpoints for integration with private clouds or networks (Virtual Private Network)
  - HTTP based storage APIs like Amazon S3, Google Cloud Storage
  - Most also include LBaaS and on demand scaling

Lecture 3 - Virtualization, Containers

- Cloud
- Platform as a Service (PaaS)
  - You run an application using a framework or environment that is specific to a vendor e.g. Heroku
  - Everything is managed for you
    - CPU allocation, scaling, load balancing
    - Network (public and private)
    - Even installed and available packages

Lecture 3 - Virtualization, Containers

- Cloud
- Platform as a Service (PaaS)
    - Most of the decision making is deciding on scaling thresholds to keep costs under control
    - Disadvantages
        - Available packages (vendor has to provide or support them)
        - Cost, out of date software, lack of control

Lecture 3 - Virtualization, Containers

- Cloud

- Software as a Service (SaaS)
  - a.k.a a website

- As an SA you will need to be able to evaluate SaaS offerings to see how they can integrate into your infrastructure

- Advice: try to pick systems that are well documented, and preferably ones with early access to forthcoming changes

Lecture 3 - Virtualization, Containers

- API aside - especially important
  - Refresher: Application Program Interface
    - A way to programmatically interact with any of the above (Iaas, Paas, SaaS)
  - When choosing any, look for vendors who have thoroughly documented, versioned APIs, e.g.
  - https://developer.github.com/v3 (Github is SaaS, or GaaS if you prefer. aaS acronyms abound)

Lecture 3 - Virtualization, Containers

- Types of Machines
- Physical
- Use when something needs maximum resources
- E.g. databases like physical machines

Lecture 3 - Virtualization, Containers

- Types of Machines
- Virtual
- Default to virtual when you can
- Overhead is slight, management and programmability are easy

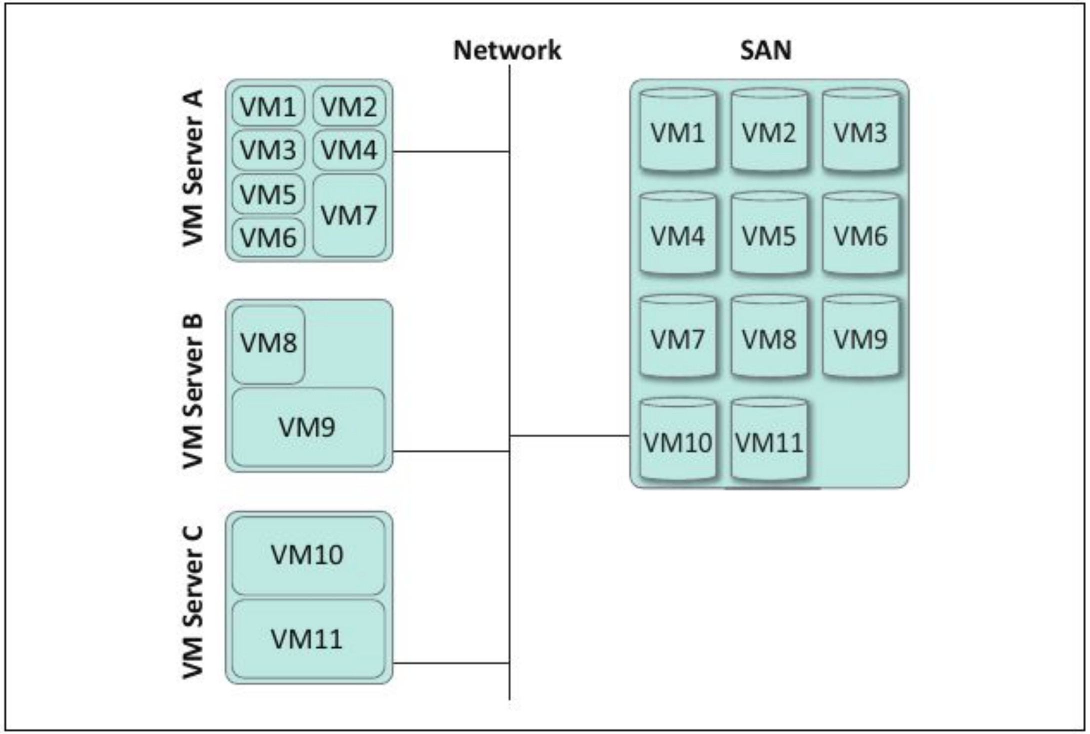


Figure 13.1: VM servers with shared storage

Lecture 3 - Virtualization, Containers

- Types of Machines
  - Virtual server
    - A portion of a physical machine that has been divided into multiple guest operating systems
    - Guests are usually unaware they are virtual
    - They do not usually have direct control over hardware, or access to other guests on a host

Lecture 3 - Virtualization, Containers

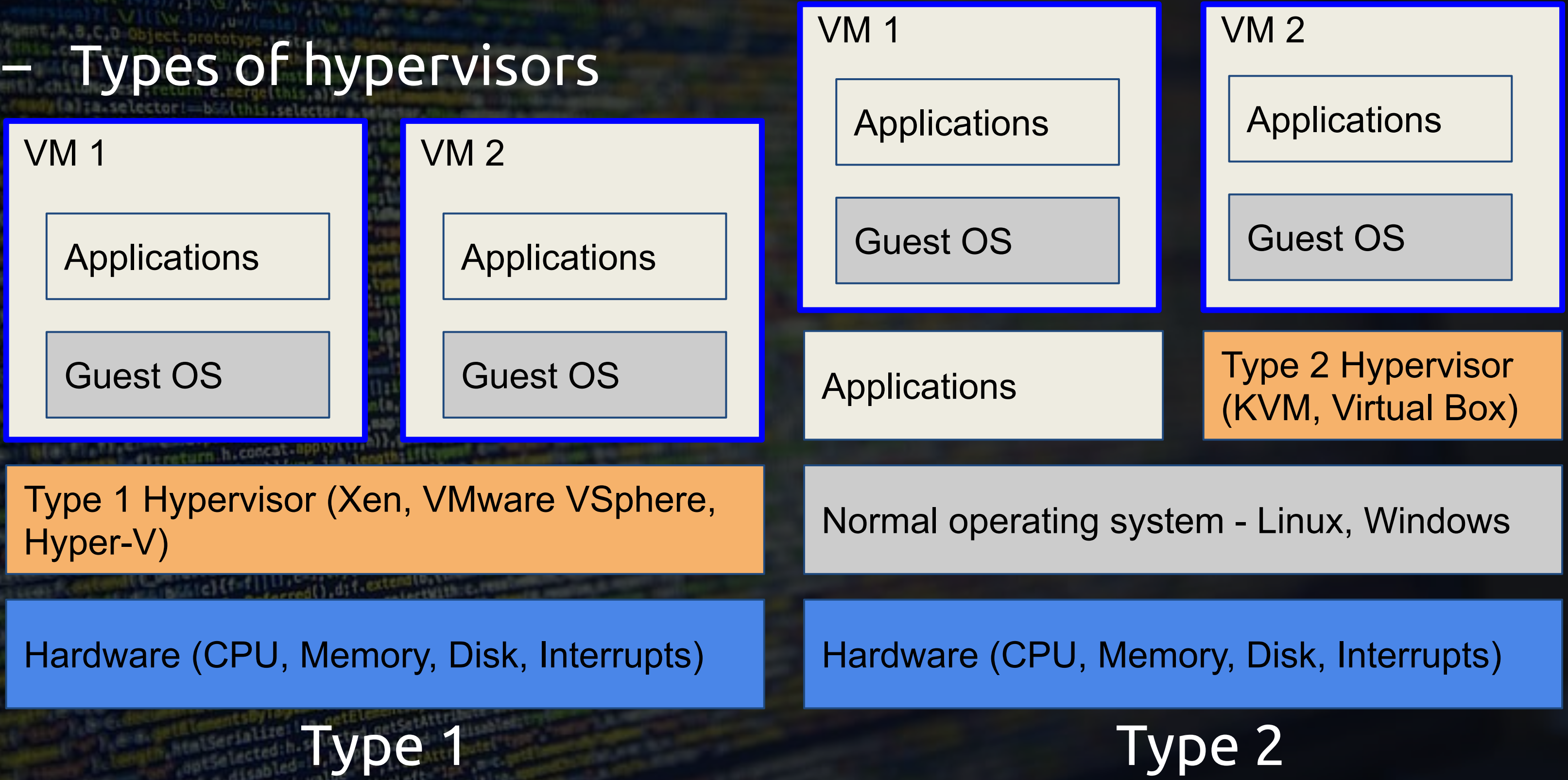
- Types of Machines
  - Virtual server components:
    - Consist of host physical server + **VMM - virtual machine monitor**
    - a.k.a **hypervisor**, the shim OS that runs guest operating systems, and translates VM requests for virtual hardware resources into requests to physical devices like network cards, disk writes

Lecture 3 - Virtualization, Containers

- Types of Machines
  - Hypervisor types, 2 main kinds:
    - **Type 1**: PV - **paravirtualization** . I/O calls from a guest are patched to use hypervisor methods. A bit faster because there's no emulation to do
    - **Type 2**: **HVM** hardware virtual machine (Linux, KVM). Guest OS makes I/O calls to what it thinks is an actual device. VMM intercepts call, does it itself, then returns the result back to the VM

Lecture 3 - Virtualization, Containers

– Types of hypervisors



VM 1

Applications

Guest OS

VM 2

Applications

Guest OS

Type 1 Hypervisor (Xen, VMware VSphere, Hyper-V)

Hardware (CPU, Memory, Disk, Interrupts)

Type 1

VM 1

Applications

Guest OS

VM 2

Applications

Guest OS

Applications

Type 2 Hypervisor (KVM, Virtual Box)

Normal operating system - Linux, Windows

Hardware (CPU, Memory, Disk, Interrupts)

Type 2

Lecture 3 - Virtualization, Containers

– Types of ‘Machines’

- Virtual server advantages:
    - Increased compute efficiency/density
    - Instead of dedicating a whole server to one application, the host can run multiple isolated virtual machines, each with their own program
    - e.g, nodejs, single threaded, 1CPU:process

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Virtual server advantages:
    - You can split infrastructure into separate VMs and manage each separately, even though they might all be on the same physical host, or different ones
    - Programmable: APIs to manage virtual servers are much more robust than physical servers:
      - proprietary hardware, out of date firmware etc.

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Virtual server advantages:
    - Fast to create and destroy identical VMs
    - Machines can be migrated between physical hosts with little to no downtime for the services in them

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
    - Virtual server migrations:
        - PSNA provides a good overview of how to plan for migrating VMs, with a number of different scenarios and architectures:

Lecture 3 - Virtualization, Containers

- Virtual server migrations:
  N + redundancy
- (a) spare server
- (b) distributed capacity
- (c) insufficient capacity
- (d) Towers of Hanoi repacked
  capacity

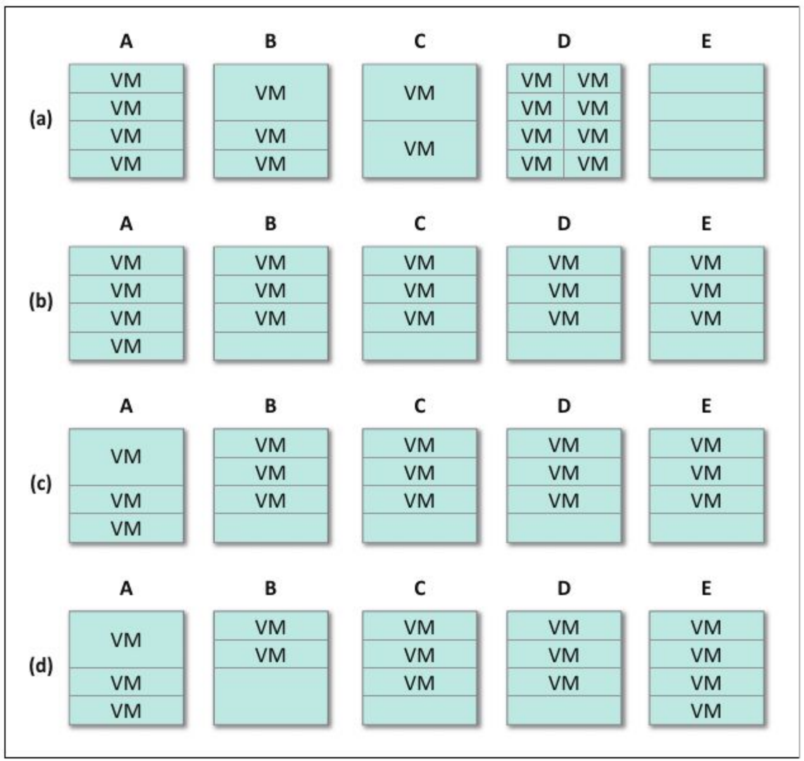


Figure 13.3: VM packing

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Virtual server disadvantages:
    - Can be very prone to over committing available resources, e.g. CPU steal (aka noisy neighbour)
    - Chapter describes Netflix’s strategy to deal with the issue: delete the VM and provision another
      - Cross your fingers and hope it works

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Virtual server disadvantages:
    - Network and disk are usually common resources in a host, one VM with lots of network traffic can cause dropped packets for others
    - Likewise, hard drives have upper read/write limits, and one VM can use too much I/O causing everyone to suffer slow requests

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Containers
    - Isolated processes with resource controls for each group
    - On Linux, a container is just a collection of related process and network namespaces
    - Many containers run on one physical or virtual host

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Containers
    - Docker, rkt, LXD, Solaris Zones, BSD Jails. PaaS providers usually use containers
    - Each container system relies on the host OS to provide process isolation, defined resources
    - An application in a container is like any other on the host when viewed from outside the container

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
    - Container advantages
        - Different versions of an application can run on the same host independent of each other
        - Fewer dependency problems, no OS footprint
        - Highest resource usage per host, less **stranded resources** across physical hosts

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Container advantages
    - Fine grained resource allocation - docker run --help
    - Fast snapshotting
    - Easy incorporation into continuous integration (CI) tools and environments for beta & prod deploys

Lecture 3 - Virtualization, Containers

- Types of ‘Machines’
  - Container disadvantages
  - Containers can become stale and out of date easily
    - Use git & CI to version control & build containers
  - Physical host maintenance still requires downtime or container migration
    - Tools like mesos, kubernetes mitigate this by shifting units across a cluster as needed

Lecture 3 - Virtualization, Containers

- Takeaways

- Book says default to virtual - but containers run on VMs

- Plan for costs up front, especially with IaaS & PaaS providers, which can get very costly very quickly

- Build an MVP using cloud provider to quickly gauge the usefulness of an idea. Saves building your own private infrastructure if it isn’t a good idea.

Lecture 3 - Virtualization, Containers

- Takeaways
- I say default to containers if you can - they're taking over infrastructure very quickly
- Don't deploy containers without some kind of version control (git), and CI process (travis-ci, jenkins, concourse-ci)
  - Your images will get stale quickly
- docker runs on Mac, Windows, & Linux, use a container for a fast MVP to gauge effort

Lecture 3 - Virtualization, Containers

- Takeaways

- VMs suffer configuration drift, with CI, your containers will match production & staging bit for bit

- Learn container orchestration tools - kubernetes, mesos, docker swarm, even docker-compose locally

    - A career awaits the sysadmin who puts in the effort learning about containers and orchestration