**Intro**

Cloud Computing Enablers - Drivers:

* Business: capacity planning, cost reduction, organizational agility.
* Tech: computing clusters, grid computing, virtualization.

Estimating IT Costs:

* Cap-ex (Capital expenditure) – one-time expense: physical building.
* Op-ex (Operational expenditure) – reoccurring expense: people, utility.

Scaling:

* Horizontal: increase number of servers. Not expensive. Instantaneous. No setup-time.
* Vertical: increase quality of servers. Expensive. Not Instantaneous. ++setup-time.

++ of cloud computing: -- capital and costs, ++ scalability, availability, and reliability.

-- of cloud computing: ++ vulnerabilities, legal issues, -- control.

Cloud computing attributes:

* On-demand usage: self-provisioning resources.
* Ubiquitous access.
* Multi-tenancy and Resource Pooling.
* **Elasticity**: ability to scale. Manuel/Automated.
* Measured Usage.
* Resiliency.

Cloud delivery Models:

* Primary:
  + Infrastructure-as-a-Service (IaaS) ex. Virtual servers.
  + Platform-as-a-Service (PaaS).
  + Software-as-a-Service (SaaS).
* Specialized:
  + Storage-as-a-Service
  + Database-as-a-Service
  + Security-as-a-Service

Cloud deployment Models: public, community, private, hybrid

**Tech**

Cloud Enabling Tech:

* Data center
  + Standardization and Modularity.
  + Automation. Tools: Puppet, Ansible, and Chef.
  + Security – usually outsourced because of its costs.
  + Hardware.
* Virtualization.
* Web Technology.
* Service Technology.

Data center classification:

1. Basic site – uptime: 99.671%
2. Redundant component Site – uptime: 99.741%
3. Concurrently maintainable site – uptime: 99.982%
4. Fault Tolerant Site Infrastructure – uptime: 99.995%

**Virtualization Intro**

Characteristics of Virtualization: abstraction, replication, isolation.

Levels of Virtualization:

* Application: multi-tenancy. Ex. Servers serving an IP address.
  + ++ efficient, stronger isolation
  + -- limited customization, hard to implement, single point failure.
  + Security: Application dependent.
* OS: containers. Ex. OpenVZ, LXC, LMCTFY, Docker, and Windows containers.
  + Originated from ‘chroot’
  + cgroup: allows for CPU and memory restrictions.
  + Namespaces: allows for resources isolation: mnt, ust, IPC, PID, net, user.
  + Hardware: virtual machines.
  + Other: networking, communication, infrastructure.
  + LXC:
    - Management system is built on cgroups and namespaces.
    - File System: is union-based. No duplicates. Copies base system and adds an overlay on top with any differences.
    - Networking: bridges, taps, and veth.
    - Security: as secure as a VM, however, earlier version had a lot of vulnerabilities.
  + Docker:
    - -- single process containers, impersistent storage
* Hardware: virtual machines
  + VMM: Virtual Machine Monitor – Hypervisor. Properties:
    - Equivalence(fidelity)
    - Resource control (safety)
    - Efficiency (performance)
  + Processor modes: user/supervisor.
  + Instruction types:
    - Privileged instructions: only runs in supervisor mode, traps otherwise.
    - Control-sensitive instructions: change resource or processor mode.
    - Behavior-sensitive instructions.
  + Virtualization Generations:
    - 1st Emulators and partial emulators. Slow.
    - 2nd Paravirtualization
    - 3rd Hardware assisted
  + Hypervisors:
    - Type 1: runs on bare metal. Ex. Linux KVM, VMware ESXi, Hyper-V
    - Type 2: runs on OS. Ex. VMware Workstation, VirtualBox, Parallels.
  + Virtualization of Resources: memory. device I/O virtualization, storage.