CSCI 5673
Distributed Systems

Lecture Set Nine

Virtual Machines and Cloud Computing

Lecture Notes by
Shivakant Mishra
Computer Science, CU Boulder
Last update: March 01, 2017

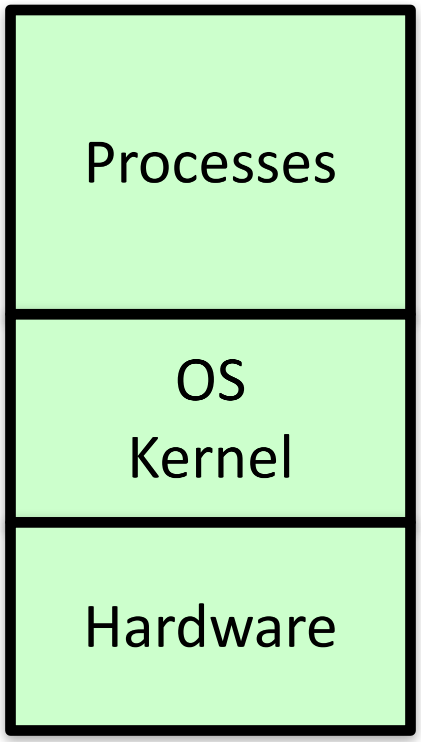
Virtual Machines

- A process already is given the illusion that it has its
  - Own memory, via virtual memory
  - Own CPU, via time slicing
  - Own I/O devices, via device-independent I/O
- Virtual machine extends this idea to give a process the illusion that it also has its own hardware
  - Usually, the process can itself be an OS

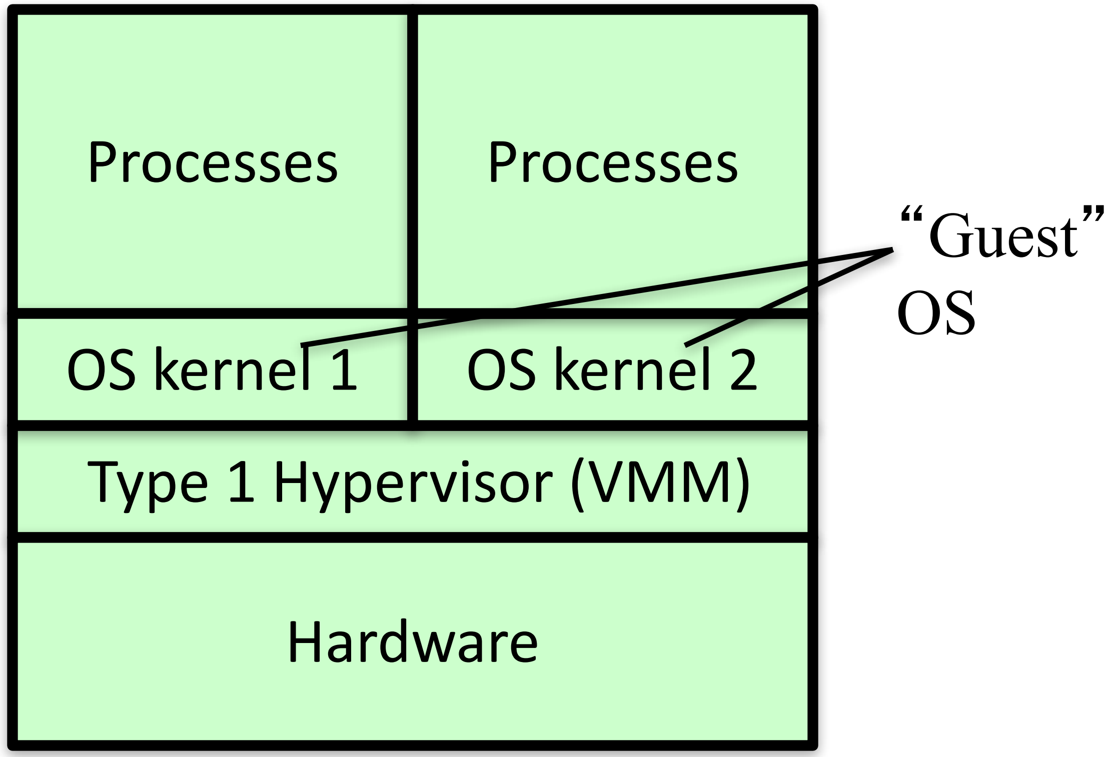
Hypervisor

- Also called virtual machine monitor (VMM)
- Allows multiple operating systems to run concurrently on a computer system
- Presents to the guest operating systems a virtual operating platform
- Monitors the execution of the guest operating systems

Type 1 Hypervisor



Traditional OS



Type 1 Hypervisor: runs directly on HW
Example: IBM's CP/CMS, Citrix XenServer, VMware ESXi, Microsoft Hyper-V

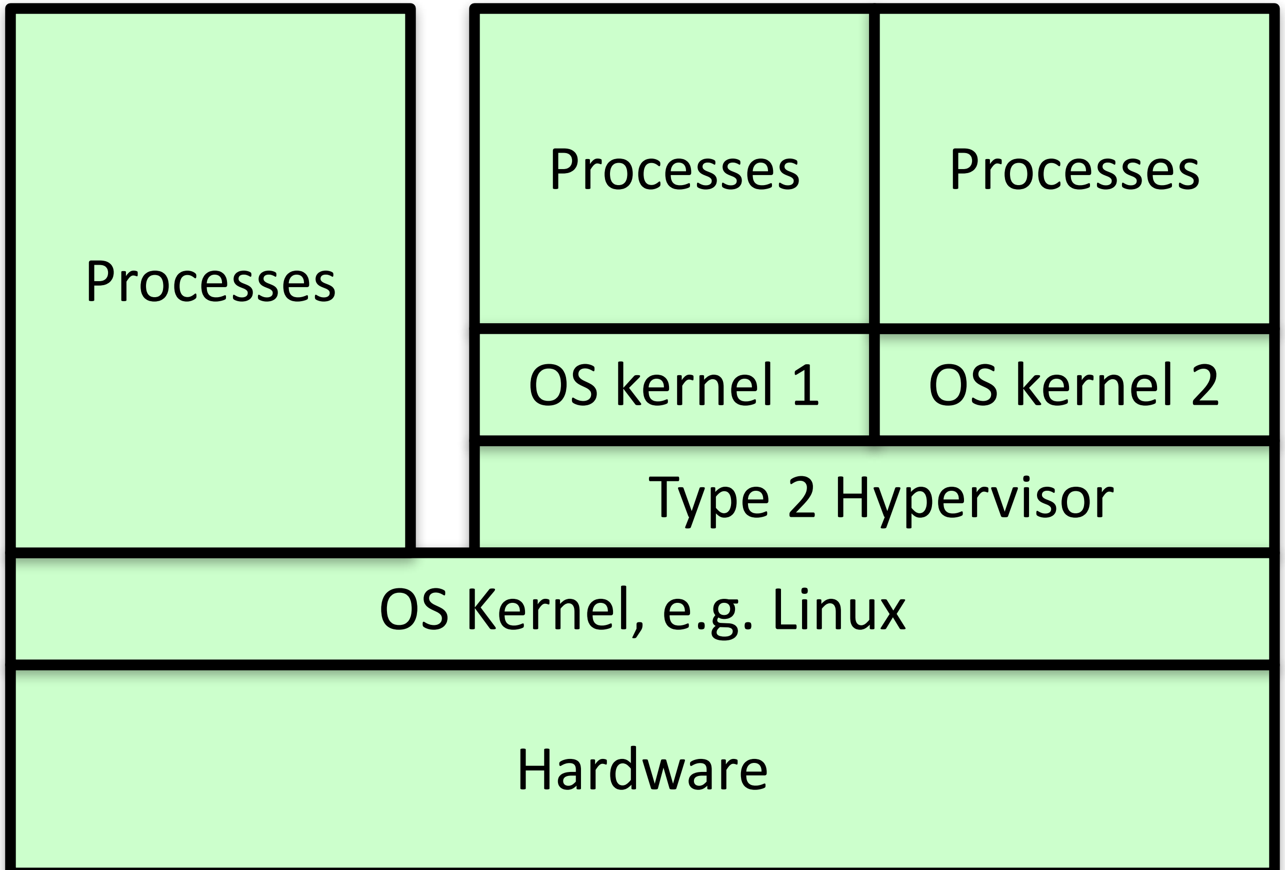
Type 1 Hypervisor

- *Type 1* (or *native*) hypervisors run directly on the host's hardware
  - A guest operating system thus runs on another level above the hypervisor
- Also known as Full virtualization
- The hypervisor provides illusion of identical hardware as the real system
  - Guest OS doesn’t have to be recompiled, and doesn’t even realize it’s executing on a VM
  - Example VMs: Microsoft Virtual Server, VMWare ESX Server
  - <u>Disadvantage</u>: Virtualized applications can run much slower, because of the many layers of fully virtualized abstraction between them and the hardware

Full Virtualization

- There are two approaches in full virtualization
  - Hardware assisted virtualization
    - Provides architectural support that facilitates building a virtual machine monitor and allows guest OSs to be run in isolation
  - Software assisted virtualization
    - the virtual machine simulates enough hardware to allow an unmodified "guest" OS to be run in isolation.

Type 2 Hypervisor



Type 2 Hypervisor: runs on an OS
Example: KVM, Virtualbox

Type 2 Hypervisor

- *Type 2* (or *hosted*) hypervisors run within a conventional operating system environment
  - The hypervisor layer as a distinct 2nd software level
  - A guest operating system runs at the third level
- There are two approaches
  - Operating system-level virtualization
    - Allows multiple isolated and secure virtualized servers to run on a single physical server.
    - The OS kernel is used to implement the "guest" environment
      - Applications running in a given "guest" environment view it as a stand-alone system
  - Paravirtualization

Paravirtualization

- *Paravirtualization* addresses the performance issue:
  - Paravirtualization provides specially defined 'hooks' to allow the guest(s) and host to request and acknowledge certain tasks, which would otherwise be executed in the hypervisor
  - An OS no longer sees full virtualization of hardware, but instead must be specially recompiled to work with this particular hypervisor, exploiting special optimizations to speed performance

Paravirtualization

- Involves modifying the OS to run in the virtualized environment as a VM → needs to be recompiled
- Virtual machine does not necessarily simulate hardware, but instead (or in addition) offers a special API that can only be used by modifying the "guest" OS
  - VMware tries to run code whenever possible directly on CPU without emulation, e.g. user code
- Example VMs: Xen, Parallels, VMware Workstation

Virtual Machines

- Process VMs, e.g. Java VMs
  - Differ from System VMs in that the goal is not to try to run multiple OS's on the same host, but to provide portable code execution of a single application across different hosts
- Java applications are compiled into Java byte code for running on any Java VM
  - Java VM acts as an interpreter of byte code, translating each byte code instruction into a local action on the host OS

Virtual Machines

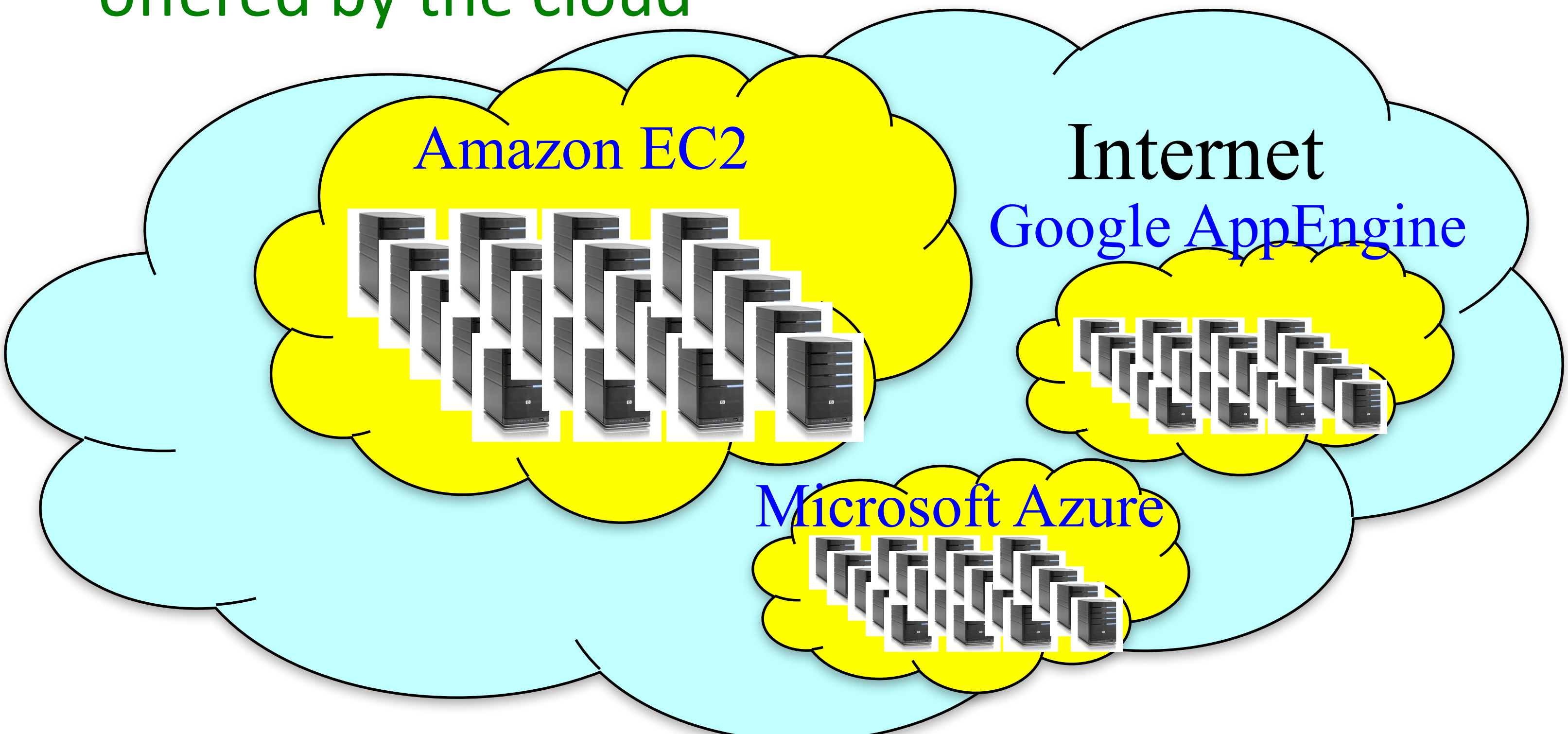
- Note Java VMs virtualize an abstract machine, not actual hardware, unlike system VMs
  - i.e. the target machine that Java byte code is being compiled for is a software specification

Virtual Machines

- Cloud Computing
  - Very easy to provision and deploy VM instances on the cloud
  - Amazon’s Elastic Compute Cloud (EC2) uses Xen virtualization
    - There are different types of VMs or instances that can be deployed:
      - Standard, High-Memory, High-CPU
    - Users can create and reboot their own VMs
    - To store data persistently, need to supplement EC2 with an additional cloud service, e.g. Amazon’s Simple Storage Service (S3)

Cloud Computing

- Computing as a utility
  - Renting resources, pay as you go, usage-based pricing
  - Many different models, and degrees of services offered by the cloud



Cloud Computing: Advantages

- Don’t have to pay up front to buy a lot of server hardware
- Maintenance of servers is simplified, uptime/failover is likely managed by cloud
- Statistical multiplexing of computation and storage on a large scale drives down costs 5-7X
  - Includes costs of electricity, network bandwidth, operations, software, and hardware

Cloud Computing: Advantages

- *Elasticity* of resources
  - an application can expand to a 1000 servers and contract as needed
  - Time scales of elasticity can be much smaller, with a lead time of just minutes
  - Illusion of infinite resources for the application. Don’t have to worry about overprovisioning or underprovisioning in the typical case.

Cloud Computing: Disadvantages

- How much do you trust the cloud to keep your proprietary/customer/personal data sufficiently private?
  - The cloud might actually do a better job of managing data than you could.
  - But what if data is passed to many 3rd parties?
- Long-term costs of renting may outweigh costs of building your own server farm
  - Computational and storage costs
  - Data transfer can be costly within cloud and externally

Cloud Computing: Disadvantages

- May be a performance penalty
  - Processes are running within VMs – layers of virtualization slow down execution
  - Data transfer may be a bottleneck
  - Cloud computing may not be suitable to high performance computing (HPC) apps
    - That are highly parallelizable but require tight time synchronization among many threads/processes
    - Poor visibility into the cloud

Cloud Computing

- Should your company move to the cloud?
  - Makes sense for a startup, low upfront costs
  - For mid-size companies, maybe outsource some operations
  - Large companies might be better off managing their own private cloud
    - But Netflix moved to Amazon!

Data Centers

Facebook, Prineville, OR

Google, Dalles, OR

Amazon, Dallas/Fort Worth, TX

Physical Organization

Racks and Chassis

A data center is generally organized in rows of "racks"

Fig. 1. Physical organization of a data center.

Racks and Chassis

- Each rack contains modular assets such as servers, switches, storage ‘‘bricks”, or specialized appliances
  - 78 in. high, 23–25 in. wide and 26–30 in. deep
- Servers are typically housed in self-contained chassis
- A chassis can hold up to 16 1U servers
- Rack unit (U or RU)
  - The size of a piece of rack-mounted equipment is frequently described as a number in "U". E.g., 1U, 2U …
  - A rack unit, U or RU is a unit of measure that describes the height of equipment designed to mount in a rack
  - One rack unit is 1.75 inches (4.445 cm) high.

Racks and Chassis

- Six chasses can fit in a singe rack
- Each chassis has its own power supply, fans, backplane interconnect, and management infrastructure

Cloud Computing Models

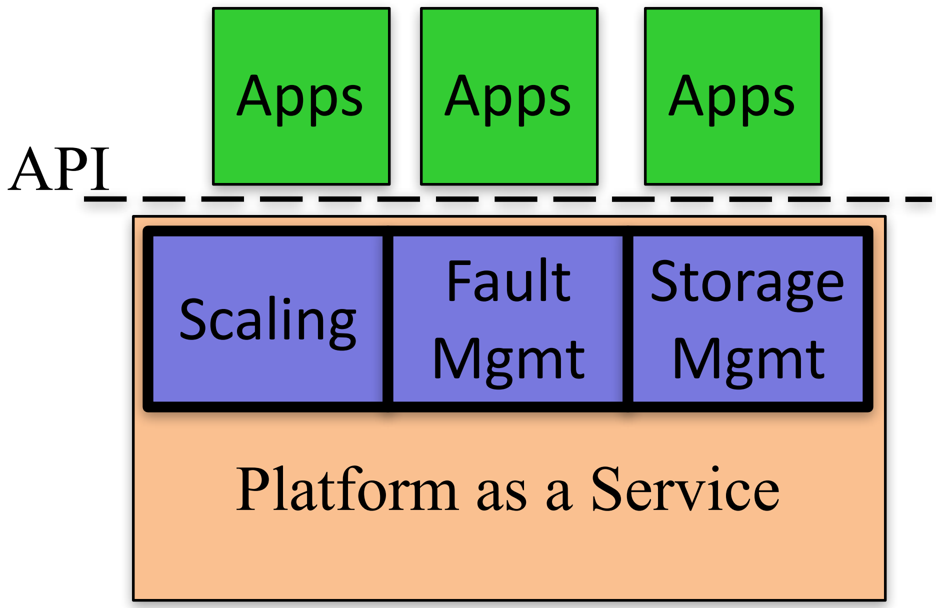
- How is cloud computing achieved?
  - Virtualization over commodity hardware
  - Each recent rack server for Google has 2TB of disk, 16 GB RAM, multi-core, 10 Gbps Ethernet, GHz CPUs
    - Run Google-optimized Ubuntu Linux

Cloud Computing Models

- Software As A Service (SaaS)
  - Software and associated data are centrally hosted on the cloud and licensed to the clients
  - Accessed by user using a thin client via web browsers
  - Common delivery model for business applications: office and messaging software, payroll processing software, DBMS software, management software, CAD software, development software, …
  - Implementation hidden as a large server farm/data center

- Platform As A Service (PaaS)
  - Provides computing platform, typically including operating system, programming language execution environment, database, and web server, as well as software tools and libraries
  - Example: Google AppEngine
  - Microsoft's Azure is somewhere between AppEngine and EC2, but closer to PaaS

PaaS


Apps
Apps
Apps
API
Scaling
Fault Mgmt
Storage Mgmt
Platform as a Service

Google AppEngine

- Makes it easy to deploy server-side Web apps using Google's scalable infrastructure
  - Free to get started
  - Compare: difficult to get LAMP stack (Linux, Apache, MySQL, Python) up and running to build a Web app
- Focused only on running Web apps
  - Not a VM provider
  - Not a grid solution – don't run general compute jobs

Google AppEngine

- Simplifies:
  - code development
  - deployment
  - monitoring
  - failover
  - scaling: Automatic scaling to hundreds of requests per second and beyond
    - Launches more application instances as necessary
    - Users charged only on cycles used over some free limit

Google AppEngine "Stack"

1. Scalable Serving Infrastructure
2. Python or Java Run-Time environment
3. Software Development Kit (SDK) with Eclipse plug-in
4. Web-based Admin Console
5. Scalable DataStore – BigTable

- Use GQL to access database, similar but not identical to SQL, not a relational DB
- BigTable is built on top of Google File System (GFS)

AppEngine APIs

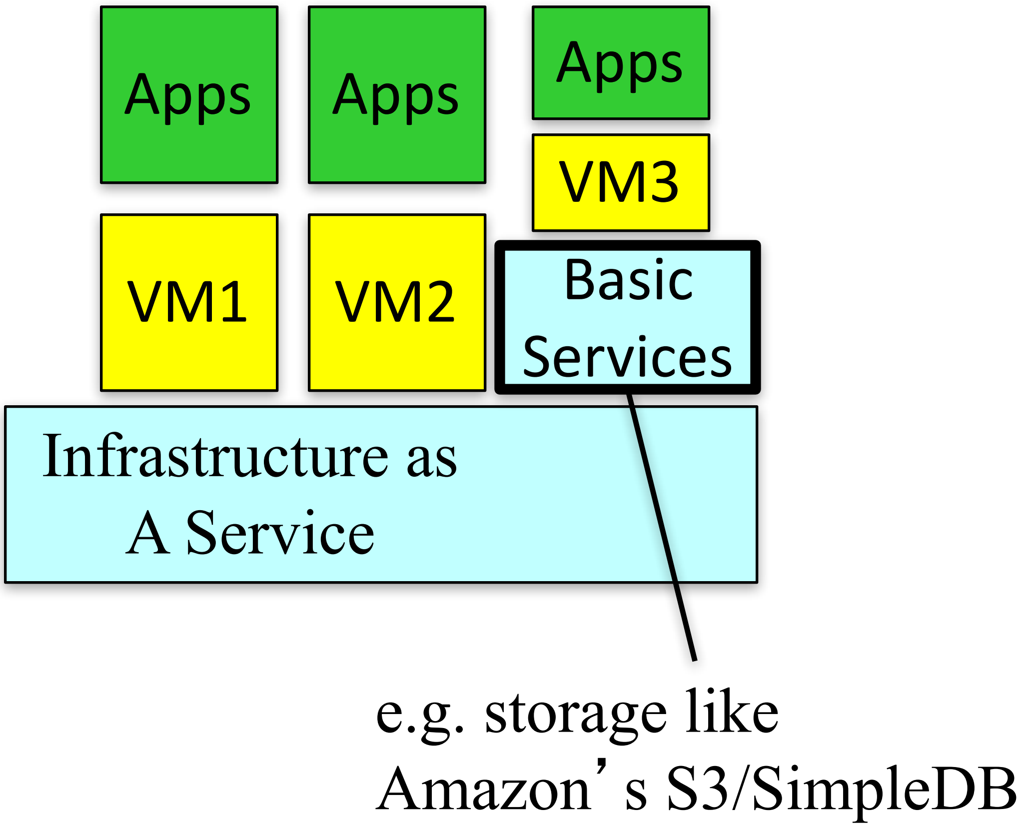
- Current APIs allow
  - storing and retrieving data from a BigTable non-relational database;
  - making HTTP requests;
  - sending e-mail;
  - manipulating images;
  - Google authentication
  - frameworks
  - and caching.

AppEngine Limitations

- App lock-in
  - Coding environment not a simple VM
  - Coding in Python and Java to APIs
- Data lock-in
  - GQL, not SQL
  - Most existing Web applications can't run on App Engine without modification, because they require a relational database.

- Infrastructure As A Service (IaaS)
  - CPU, memory, storage, networking resources, IP addresses
  - Users get complete control of the computing resources
    - Very easy to provision and deploy VM instances on the cloud
  - Amazon's Elastic Compute Cloud (EC2), Rackspace
  - Amazon's Elastic Compute Cloud (EC2) uses Xen virtualization
    - There are different types of VMs or instances that can be deployed:
      - Standard, High-Memory, High-CPU
    - Users can create and reboot their own VMs
    - To store data persistently, need to supplement EC2 with an additional cloud service, e.g. Amazon's Simple Storage Service (S3)

IaaS


Apps
Apps
Apps
VM3
VM1
VM2
Basic Services
Infrastructure as A Service
e.g. storage like Amazon's S3/SimpleDB

- Other cloud computing models:
  - Desktop as a Service (DaaS), Backend as a Service (BaaS), Information Technology Management as a Service (IMaaS)