Cloud Models and Platforms

Dr. Sanjay P. Ahuja, Ph.D.
Fidelity National Financial Distinguished Professor of CIS
School of Computing, UNF
A Working Definition of Cloud Computing

- Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

- This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.
Objectives of Cloud Computing

- **Elasticity**: Ability to scale virtual machines resources up or down
- **On-demand usage**: Ability to add or delete computing power (CPU, memory), and storage according to demand
- **Pay-per-use**: Pay only for what you use
- **Multitenancy**: Ability to have multiple customers access their servers in the data center in an isolated manner
5 Essential Cloud Characteristics

- On-demand self-service
- Broad network access
- Resource pooling
  - Location independence
- Rapid elasticity
- Measured service
3 Cloud Service Models

- **Cloud Software as a Service (SaaS)**
  - The **capability provided to the consumer is to use the provider's applications** running on a cloud infrastructure and accessible from various client devices through a thin client interface such as a Web browser (e.g., web-based email). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

- **Cloud Platform as a Service (PaaS)**
  - The **capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created applications using programming languages and tools supported by the provider** (e.g., Java, Python, .Net). The consumer does not manage or control the underlying cloud infrastructure, network, servers, operating systems, or storage, but the consumer has control over the deployed applications and possibly application hosting environment configurations.

- **Cloud Infrastructure as a Service (IaaS)**
  - The **capability provided to the consumer is to rent processing, storage, networks, and other fundamental computing resources** where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly select networking components (e.g., firewalls, load balancers).

- To be considered “cloud” they must be deployed on top of cloud infrastructure that has the key characteristics
Service Model Architectures

- Software as a Service (SaaS) Architectures
  - Cloud Infrastructure
    - IaaS
    - PaaS
    - SaaS

- Platform as a Service (PaaS) Architectures
  - Cloud Infrastructure
    - IaaS
    - PaaS

- Infrastructure as a Service (IaaS) Architectures
  - Cloud Infrastructure
    - IaaS
Saas, PaaS, IaaS
3 Features of Mature SaaS Applications

- SaaS is hosting applications on the Internet as a service (both consumer and enterprise)

- Features of Mature Saas applications:
  - **Scalable**
    - Handle growing amounts of work in a graceful manner
  - **Multi-tenancy**
    - One application instance may be serving hundreds of companies
    - Opposite of multi-instance where each customer is provisioned their own server running one instance
  - **Metadata driven configurability**
    - Instead of customizing the application for a customer (requiring code changes), one allows the user to configure the application through metadata
SaaS Maturity Levels

- Level 1: Ad-Hoc/Custom
- Level 2: Configurable
- Level 3: Configurable, Multi-Tenant-Efficient
- Level 4: Scalable, Configurable, Multi-Tenant-Efficient
3 Cloud Deployment Models

- **Private cloud**
  - The cloud infrastructure is operated solely for an organization. It may be managed by the organization or a third party and may exist on premise or off premise.

- **Public cloud**
  - Mega-scale cloud infrastructure is made available to the general public or a large industry group and is owned by an organization selling cloud services.

- **Hybrid cloud**
  - The cloud infrastructure is a composition of two or more clouds (private or public) that remain unique entities but are bound together by standardized or proprietary technology that enables data and application portability.
Common Cloud Characteristics

- Cloud computing often leverages:
  - Massive scale
  - Homogeneity
  - Virtualization
  - Low cost software
  - Geographic distribution
  - Advanced security technologies
Security is the Major Issue (2008)

Q: Rate the challenges/issues ascribed to the 'cloud'/on-demand model
(1=not significant, 5=very significant)

- Security: 74.6%
- Performance: 63.1%
- Availability: 63.1%
- Hard to integrate with in-house IT: 61.1%
- Not enough ability to customize: 55.8%
- Worried on-demand will cost more: 50.4%
- Bringing back in-house may be difficult: 50.0%
- Regulatory requirements prohibit cloud: 49.2%
- Not enough major suppliers yet: 44.3%

Source: IDC Enterprise Panel, August 2008  n=244
Cloud Challenges are Changing (2016)

Cloud Challenges 2016 vs. 2015

- Compliance: 26% (2016), 25% (2015)
- Managing multiple cloud services: 26% (2016), 25% (2015)
- Managing costs: 26% (2016), 24% (2015)
- Governance/Control: 23% (2016), 23% (2015)
- Performance: 15% (2016), 17% (2015)

Lack of resources/expertise replaces security as the #1 cloud challenge.
Cloud Security Advantages

- Shifting public data to a external cloud reduces the exposure of the internal sensitive data
- Dedicated Security Team
- Greater Investment in Security Infrastructure
- Cloud homogeneity makes security auditing/testing simpler
- Clouds enable automated security management and real-time detection of system tampering
- Rapid Re-Constitution of Services
- Redundancy / Disaster Recovery
Cloud Security Challenges

- Trusting vendor’s security model
- Multi-tenancy
- Data ownership issues
- QoS guarantees
- Attraction to hackers (high-value target)
- Security of virtual OSs in the cloud
- Obtaining support from cloud vendor for security related investigations
Cloud Security Challenges

- Indirect administrator accountability
- Proprietary cloud vendor implementations can’t be examined
- Loss of physical control
- Possibility for massive outages

- Encryption needs for cloud computing
  - Encrypting access to the cloud resource control interface
  - Encrypting administrative access to OS instances
  - Encrypting access to applications
  - Encrypting application data at rest
Typical use case of provisioning a virtual machine
Typical use case of provisioning a virtual machine

- The *management environment* consists of components required to effectively deliver services to consumers. The various services offered span from image management and provisioning of machines to billing, accounting, metering, and more. The cloud management system (CMS) forms the heart of the management environment along with the hardware components.

- The *managed environment* is composed of physical servers and in turn the virtual servers that are “managed-by” the management environment. The servers in the managed environment belong to a customer pool; where customers or users can create virtual servers on-demand and scale up/down as needed.

- The management environment controls and processes all incoming requests to create, destroy, manage, and monitor virtual machines and storage devices. In the context of a public cloud, the users get direct access to the VMs created in the managed environment, through the Internet. They can access the machines after they are provisioned by the management layer.
Typical use case of provisioning a virtual machine

The previous figure describes the following actions:

1. User makes a request to create a VM by logging onto the cloud portal.
2. The request is intercepted by the request manager and is forwarded to the management environment.
3. The management environment, on receiving the request, interprets it and applies to it provisioning logic to create a VM from the set of available physical servers.
4. External storage is attached to the VM from a storage area network (SAN) store during provisioning in addition to the local storage.
5. After the VM is provisioned and ready to use, the user is notified of this information and finally gains total control of the VM. The user can access this VM through the public Internet because the VM has a public IP address (e.g. through SSH).
Cloud Ecosystem

Figure. The cloud ecosystem for building private clouds. (a) Cloud consumers need flexible infrastructure on demand.
(b) Cloud management provides remote and secure interfaces for creating, controlling, and monitoring virtualized resources on an infrastructure-as-a-service cloud. (c) Virtual infrastructure (VI) management provides primitives to schedule and manage VMs across multiple physical hosts. (d) VM managers provide simple primitives (start, stop, suspend) to manage VMs on a single host.

Figure from Virtual Infrastructure Management in Private and Hybrid Clouds, Internet Computing, September 2009.
Cloud Ecosystem

- The public cloud ecosystem has evolved around providers, users, and technologies.

- The previous figure suggests one possible ecosystem for private clouds. There are 4 levels of development of ecosystem development: cloud users/consumers, cloud management, VI management, and VM managers.

- At the cloud management level, the cloud manager provides virtualized resources over an IaaS platform.

- At the virtual infrastructure (VI) management level, the manager allocates VMs over multiple server clusters. Examples: OpenNebula, VMWare vSphere. These can manage VM managers like Xen, KVM etc. These support dynamic placement and VM management on a pool of physical resources, automatic load balancing, server consolidation, and dynamic infrastructure resizing and partitioning.

- Finally, at the VM management level the VM managers handles VMs installed on individual host machines. Examples: Xen, VMWare, KVM.

- An ecosystem of cloud tools attempts to span both cloud management and VI management. Besides public clouds such as Amazon EC2, open source cloud tools for virtualization of cloud infrastructure include Eucalyptus and Globus Nimbus.

- To access these cloud tools, one can use the Amazon EC2WS interface among others.
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizeable computing capacity—literally, servers in Amazon’s data centers—that you use to build and host your software systems. You can access the components and features that EC2 provides using a web-based GUI, command line tools, and APIs.

With EC2, you use and pay for only the capacity that you need. This eliminates the need to make large and expensive hardware purchases, reduces the need to forecast traffic, and enables you to automatically scale your IT resources to deal with changes in requirements or spikes in popularity related to your application or service.


Amazon Cloud EC2: AMI

- An Amazon Machine Image (AMI) is a template that contains a software configuration (operating system, application server, and applications). From an AMI, you launch instances, which are running copies of the AMI. You can launch multiple instances of an AMI, as shown in the following figure.

- Your instances keep running until you stop or you terminate them, or until they fail. If an instance fails, you can launch a new one from the AMI.

- You can use a single AMI or multiple AMIs depending on your needs. From a single AMI, you can launch different types of instances.
Amazon Cloud EC2: AMI

- An *instance type* is essentially a hardware archetype. As illustrated in the following figure, you select a particular instance type based on the amount of memory and computing power you need for the application or software that you plan to run on the instance.

- Amazon publishes many AMIs that contain common software configurations for public use. In addition, members of the AWS developer community have published their own custom AMIs.

- For example, if your application is a website or web service, your AMI could be preconfigured with a web server, the associated static content, and the code for all dynamic pages. Alternatively, you could configure your AMI to install all required software components and content itself by running a bootstrap script as soon as the instance starts. As a result, after launching the AMI, your web server will start and your application can begin accepting requests.
Amazon Cloud EC2: Regions and Availability Zones

- Amazon has data centers in different areas of the world (for example, North America, Europe, and Asia). Correspondingly, Amazon EC2 is available to use in different Regions. By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements. Prices for Amazon EC2 usage vary by Region.

- Each Region contains multiple distinct locations called Availability Zones (illustrated in the following diagram). Each Availability Zone is engineered to be isolated from failures in other Availability zones and to provide inexpensive, low-latency network connectivity to other zones in the same Region. By launching instances in separate Availability Zones, you can protect your applications from the failure of a single location.
Amazon Cloud EC2: Storage

- To store data, Amazon EC2 offers the following storage options:
  1. Amazon Elastic Block Store (Amazon EBS)
  2. Amazon EC2 Instance Store
  3. Amazon Simple Storage Service (Amazon S3)

**Amazon EBS**
- Amazon EBS volumes are the recommended storage option for the majority of use cases. Amazon EBS provides the instances with persistent, block-level storage. Amazon EBS volumes are essentially hard disks that you can attach to a running instance.

- Amazon EBS is particularly suited for applications that require a database, file system, or access to raw block-level storage.
Amazon Cloud EC2: Storage

- To keep a back-up copy, you can create a *snapshot* of the volume. As illustrated in the following figure, snapshots are stored in Amazon S3.

- You can create a new Amazon EBS volume from a snapshot, and attach it to another instance, as illustrated in the following figure.
Amazon Cloud EC2: Storage

- You can also detach a volume from an instance and attach it to a different one, as illustrated in the following figure.

- **Instance Store**

  All instance types, with the exception of Micro instances, offer *instance store*. This is storage that doesn't persist if the instance is stopped or terminated. Instance store is an option for inexpensive temporary storage. You can use instance store volumes if you don't require data persistence.

- **Amazon S3**

  Amazon S3 is storage for the Internet. It provides a simple web service interface that enables you to store and retrieve any amount of data from anywhere on the web.
Amazon Cloud EC2: Storage

The following figure shows the relationship between these storage options and your instance.
Amazon Cloud S3

Amazon S3 Functionality
1. Write, read, and delete objects containing from 1 byte to 5 terabytes of data each.
2. The number of objects you can store is unlimited.
3. Each object is stored in a bucket and retrieved via a unique, developer-assigned key.
4. A bucket can be stored in one of several Regions. You can choose a Region to optimize for latency, minimize costs, or address regulatory requirements.
5. Objects stored in a Region never leave the Region unless you transfer them out. For example, objects stored in the EU (Ireland) Region never leave the EU.
6. Authentication mechanisms are provided to ensure that data is kept secure from unauthorized access. Objects can be made private or public, and rights can be granted to specific users.
7. Options for secure data upload/download and encryption of data at rest are provided for additional data protection.
8. Uses standards-based REST and SOAP interfaces designed to work with any Internet-development toolkit.
Amazon Cloud S3: Use Cases

Content Storage and Distribution
- Amazon S3 can store a variety of content ranging from web applications to media files. A user can offload an entire storage infrastructure onto the cloud.

Storage for Data Analysis
- Whether a user is storing pharmaceutical data for analysis, financial data for computation and pricing, or photo images for resizing, Amazon S3 can be used to store the original content. The user can then send this content to Amazon EC2 for computation, resizing, or other large scale analytics – without incurring any data transfer charges for moving the data between the services.

  E.g. For example, you could stream terabytes of data off of a genomic sequencer as it is being created, store the final data set as a single object and then analyze any subset of the data in EC2 using a ranged GET.

Backup, Archiving and Disaster Recovery
- The Amazon S3 solution offers a scalable and secure solution for backing up and archiving critical data.
Amazon Cloud: Databases

If the application running on EC2 needs a database, the common ways to implement a database for the application are:

1. Use Amazon Relational Database Service (Amazon RDS) to get a managed relational database in the cloud

2. Launch an instance of a database AMI, and use that EC2 instance as the database

Amazon RDS offers the advantage of handling database management tasks, such as patching the software, backing up and storing the backups.
Each instance is launched into the Amazon EC2 network space and assigned a public IP address. If an instance fails and a replacement instance is launched, the replacement will have a different public IP address than the original.

Security groups are used to control access to user instances. These are analogous to an inbound network firewall that allows a user to specify the protocols, ports, and source IP ranges that are allowed to reach user instances.

A user can create multiple security groups and assign different rules to each group. Each instance can be assigned to one or more security groups, and the rules determine which traffic is allowed in to the instance. A security group can be configured so that only specific IP addresses or specific security groups have access to the instance.
The following figure shows a basic three-tier web-hosting architecture running on Amazon EC2 instances. Each layer has a different security group (indicated by the dotted line around each set of instances). The security group for the web servers only allows access from hosts over TCP on ports 80 and 443 (HTTP and HTTPS) and from instances in the App Servers security group on port 22 (SSH) for direct host management.

The security group for the app servers allows access from the Web Servers security group for web requests, and from the corporate subnet over TCP on port 22 (SSH) for direct host management. The user’s support engineers could log directly into the application servers from the corporate network, and then access the other instances from the application server boxes.

The DB Servers security group permits only the App Servers security group to access the database servers.
Amazon Cloud: Networking and Security

- Only permit web layer access to app layer
- Only permit app layer access to DB layer

Port 80 (HTTP) and 443 (HTTPS) of web layer open to Internet

Only port 22 (SSH) of app layer open to developers in corporate network

All other traffic denied
Amazon Cloud: Monitoring, Auto Scaling, and Load Balancing

- AWS provides several features that enable the following:

  - Monitor basic statistics for instances and Amazon EBS volumes.
  - Automatically scale EC2 capacity up or down according to conditions defined by the user.
  - Automatically distribute incoming application traffic across multiple EC2 instances. It detects unhealthy instances and reroutes traffic to healthy instances until the unhealthy instances have been restored. Elastic Load Balancing automatically scales its request handling capacity in response to incoming traffic.
  - Elastic Load Balancing provides several different interfaces that can be used to manage a user’s load balancers. Users can create, access, and manage their load balancers using the AWS Management Console, the command line interface (CLI), or the Query API. Users need to install the command line interface and the Query API before they can be used.
Amazon Cloud: Identity and Access Management (IAM)

- Amazon EC2 integrates with AWS Identity and Access Management (IAM), a service that lets the user organization do the following:
  - Create users and groups under user organization's AWS account
  - Share an organization’s AWS account resources between the users in the account
  - Assign unique security credentials to each user
  - Granularly control users access to services and resources
  - Get a single AWS bill for all users under the AWS account
  - For example, you can use IAM with Amazon EC2 to control which users under an AWS account can create AMIs or launch instances.