Lecture 2: Cloud Overview

Outline

- Cloud computing
- Cloud environment
- Cloud essentials
- Cloud characteristics and properties
- Cloud deployment models
- Cloud as a service
- Service models
DEFINITION 3.1: Cloud computing provides a model for enabling on-demand network access to a shared pool of computing resources, rapidly and with minimal management efforts.

EXAMPLE 3.1: Example of service that are provided networks, servers, storage, applications, and services
**LECTURE 2: CLOUD OVERVIEW**

**WHAT DOES IT TAKE TO MANAGE A CLOUD INFRASTRUCTURE?**

<table>
<thead>
<tr>
<th>Software &amp; Hardware Stack</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATIONS</td>
<td></td>
</tr>
<tr>
<td>MIDDLEWARE</td>
<td></td>
</tr>
<tr>
<td>DATABASE</td>
<td></td>
</tr>
<tr>
<td>OPERATING SYSTEM</td>
<td></td>
</tr>
<tr>
<td>VIRTUAL MACHINE</td>
<td></td>
</tr>
<tr>
<td>SERVERS</td>
<td></td>
</tr>
<tr>
<td>STORAGE</td>
<td></td>
</tr>
</tbody>
</table>

**INSTALLING AND MAINTAINING NEW APPLICATIONS IS A NIGHTMARE**
- Keep running updated
- maintaining
- customizing

**TIME CONSUMING**
- Installing required OS
- creating virtual machines

**SOMETIMES SOME PHYSICAL WORK IS NEED** (UPGRADING HARDWARE PART)

**Security**
LECTURE 2: CLOUD OVERVIEW

DEFINING CLOUD ENVIRONMENT

- Traditional environment
Disadvantage of the traditional system

- Over or under-provisioning
- Underestimating provisioning
- Overestimating system utilization

Solution: Low utilization

Dynamic provisioning
LECTURE 2: CLOUD OVERVIEW
DEFINING CLOUD ENVIRONMENT

• New environment
  a) Create cloud with predefined resources, users, quotas, and access rights

VM MANAGEMENT TOOL

Admin

b) Request resource

SELF-SERVICE PORTAL

User
LECTURE 2: CLOUD OVERVIEW
CLOUD ESSENTIALS

• Cloud platform (or cloud computing) is a utility computing

• All the services provided by the cloud are controlled and monitored by the cloud provider via a pay-per-use business model.

• The ideal cloud computing platform is:
  - Efficient regarding resource management
  - scalable
  - elastic
  - self-managing
  - highly available
  - Accessible
What makes a cloud a cloud:

- Self-service access.
- Elasticity.
- Metered use of resources.
- Pooled Resources
LECTURE 2: CLOUD OVERVIEW

CLOUDS: DEPLOYMENT MODELS

• Public cloud

DEFINITION 10.1:
A public cloud is an open infrastructure to the general public where anyone can access the services provided by the cloud.

• Private cloud

DEFINITION 10.2:
A private cloud is provisioned for privilege and exclusive infrastructure for use by a single organization comprising multiple consumers.

• Community cloud

DEFINITION 10.3:
A community cloud is similar to private cloud the only difference is in the fact that the exclusive users from the community share the same concerns.

• Hybrid cloud

DEFINITION 10.4:
A hybrid cloud is a composition of two or more distinct cloud infrastructure. (e.g. merging between private, public, and community clouds)
LECTURE 2: CLOUD OVERVIEW
CLOUD AS A SERVICE

IaaS
INFRASTRUCTURE AS A SERVICE

PaaS
PLATFORM AS A SERVICE

SaaS
SOFTWARE AS A SERVICE

FILE
WEB
EMAIL
CACHING
STREAMING
ERP
SECURITY
DECISION SUPPORT
NETWORKING
APPLICATION DEVELOPMENT
COLLABORATIVE
LECTURE 2: CLOUD OVERVIEW
CLOUD CORE STACK

APPLICATION
- ANALYTICAL
- TRANSACTIONAL
- INTERACTIVE
- BROWSING

APPLICATION CAPABILITY COMPONENTS
- WEB SERVER
- APP SERVER
- REPORTING
- ESB
- CACHE
- DATABASE
- MSQQ

COMPUTING FRAMEWORKS
- TRANSACTIONAL / DISPATCHING
- ANALYTICAL / SCHEDULING

Resources
- SERVER
- STORAGE
- NETWORK

Virtualization Resources

Physical Resources
LECTURE 2: CLOUD OVERVIEW

CLOUD AS A SERVICE

• hardware & software stack evolution:

- **Owned, operated and managed by the organization**
- **Run, operated and managed by the provider**

[Diagram showing the evolution of hardware and software stack in cloud services]

*Source: Based on the model developed by NIST (2011)*
LECTURE 2: CLOUD OVERVIEW

CLOUD SERVICE LAYERS

SOURCE: HTTPS://WWW.KATESCOMMENT.COM/IAAS-PAAS-SAAS-DEFINITION/
# LECTURE 2: CLOUD OVERVIEW

**SERVICE MODEL**

<table>
<thead>
<tr>
<th>IaaS</th>
<th>PaaS</th>
<th>SaaS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>WHO?</strong></td>
<td>System Administrator</td>
<td>Developers</td>
</tr>
<tr>
<td></td>
<td>System Manager</td>
<td>Engineers</td>
</tr>
<tr>
<td><strong>WHAT?</strong></td>
<td>OS</td>
<td>Development</td>
</tr>
<tr>
<td></td>
<td>virtual machines</td>
<td>Integration</td>
</tr>
<tr>
<td></td>
<td>Networking</td>
<td>Application test</td>
</tr>
<tr>
<td></td>
<td>CPU</td>
<td>Service</td>
</tr>
<tr>
<td></td>
<td>Memory</td>
<td></td>
</tr>
<tr>
<td>Create platforms for</td>
<td>Create or deploy</td>
<td>To complete</td>
</tr>
<tr>
<td>service and application:</td>
<td>applications and services</td>
<td>specific tasks</td>
</tr>
<tr>
<td>test, development,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>integration, and deployment</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

15
LECTURE 2: CLOUD OVERVIEW
DELIVERY MODEL

IaaS

PaaS

SaaS

System Admin

Software engineers / Developers

Users

CLIENT

PROVIDER
Software As A Service
LECTURE 2: CLOUD OVERVIEW
SOFTWARE AS A SERVICE (SAAS)

• Characteristics:
  • On-demand service (No difference between the license fee and the hosting fee.)
  • Software delivery model:
    • No hardware deployment needed
    • Accessible anywhere, anytime
    • The application is delivered over a web browser or other thin client.
    • The application is configurable, but not customizable.
Lecture 2: Cloud Overview
Software as a Service (SaaS)

- SaaS Architecture (Maturity Model) / maturity levels:
  - Level I: Ad Hoc/Custom
  - Level II: Configurable
  - Level III: Configurable, Multi-Tenant-Efficient
  - Level IV: Scalable, Configurable, Multi-Tenant-Efficient
LECTURE 2: CLOUD OVERVIEW
SOFTWARE AS A SERVICE (SAAS)

• SaaS Architecture (Maturity Model) / maturity levels:

  • Level I: Ad Hoc/Custom
LECTURE 2: CLOUD OVERVIEW
SOFTWARE AS A SERVICE (SAAS)

• SaaS Architecture (Maturity Model) / maturity levels:
  • Level II: Configurable

  Tenant 1  Tenant 2  Tenant 3
  Instance   Instance   Instance
• SaaS Architecture (Maturity Model) / maturity levels:
  • Level III: Configurable, Multi-Tenant-Efficient
• SaaS Architecture (Maturity Model) / maturity levels:

• Level IV: Scalable, Configurable, Multi-Tenant-Efficient
• Metadata Services:
  • providing the ability to the users to create their own primary configurations and customizations of the applications to meet their needs.

• Security services:
  • The nature of SaaS makes security both a paramount concern for customers, and a high priority for application architects.
LECTURE 2: CLOUD OVERVIEW
SAAS EXAMPLES

SaaS For Enterprises

SaaS For Consumers
Platform As A Service
• Characteristics:
  • PaaS is a computing platform that abstract the infrastructure, OS, and middleware to drive developer productivity.
  • Offers an environment for developers
    • Create and deploy application
  • Different programming languages
  • Different DB
  • Web based console to create application (with dashboard)
• Types of PaaS:

  • Close PaaS: provides a fixed set of services you can use. You cannot install your own services.

  • Open PaaS: provides support for you to develop your own automated service deployments.
LECTURE 2: CLOUD OVERVIEW
PAAS EXAMPLES
Infrastructure
As A Service
LECTURE 2: CLOUD OVERVIEW
INFRASTRUCTURE AS A SERVICE (IAAS)

• Characteristics:
  • Dynamic provisioning
    • provides virtual machine
    • cannot provide elastic computing by itself (needs PaaS)
  • Multi-tenancy
    • single instance of the software runs on a server and serving multiple clients.
LECTURE 2: CLOUD OVERVIEW
INFRASTRUCTURE AS A SERVICE (IAAS)

• Two important stacks:
  • Virtualization resources
  • Physical resources Server
VIRTUALIZATION

DEFINITION 30.1:
Virtualization is the abstraction of logical resources away from the underlying physical resources.

- Virtualization

Traditional Stack

Virtualized Stack
**LECTURE 2: CLOUD OVERVIEW**

**IAAS: VIRTUALIZATION**

**DEFINITION 30.1:** Virtualization
Virtualization is the abstraction of logical resources away from the underlying physical resources.

- Virtualization

![Diagram comparing Traditional Stack and Virtualized Stack](image)

- Traditional Stack
- Virtualized Stack
LECTURE 2: CLOUD OVERVIEW
IAAS: VIRTUALIZATION

• Types of virtualization:
  
• Emulation
  
  • VM emulate/simulate complete hardware
  
  • Unmodified guest OS can be run on a different PC

• Full/native virtualization
  
  • VM simulates stick minimum hardware to allow run an unmodified guest OS in isolation
Types of virtualization:

- Para-virtualization
  - VM does not simulate the hardware
  - Special API is used by the modified guest OS
  - Hypercalls are trapped by the Hypervisor and serviced

- OS-level virtualization
  - OS allows to run multiple secure virtual servers
  - Guest OS is the same as host OS (However it appress isolated)

- Application level virtualization
  - Application is given a copy of its own component and they are shared
DEFINITION 30.1: Hypervisor
Hypervisor or Virtual Machine Monitor (VMM) is another technology at the heart of system virtualization which is responsible for allocating resources to each virtual machine.

EXAMPLE 30.1: Example of Hypervisor
Kernel-based Virtual Machine (KVM) is an open source virtualization platform

• Characteristics: (Provides)
  • Policy-based automation
  • Virtual hard disk
  • Life cycle management
  • Life migration
  • Real time resource allocation
• Hypervisors:
  • Native/Bare-Metal Hypervisor
    • Runs directly on the hardware
  • Hosted Hypervisor
    • Runs on host operating system
• Protection rings

• CPU protection rings

• Level 0: operating system kernel

• Level 1 & level 2: operating system services

• Level 3: Applications run by the user or users.
• Traditional OS
  • When application invoke a system call
    • CPU will trap to interrupt handler vector in OS.
    • CPU will switch to kernel mode (Ring0) and execute OS instructions.
  • When hardware event:
    • Hardware will interrupt CPU execution, and jump to interrupt handler in OS.
• VMM & Guest OS:
  • System Call
    • CPU will trap to interrupt handler vector of VMM
    • VMM jump back into guest OS.
  • Hardware interrupt
    • Hardware make CPU trap to interrupt handler of VMM.
    • VMM jump to corresponding interrupt handler of guest OS.
  • Privilege Instruction
    • Running privilege instructions in guest OS will be trapped to VMM for instruction emulation.
    • After emulation, VMM jump back to guest OS.
LECTURE 2: CLOUD OVERVIEW
IAAS EXAMPLES
• Clouds provide services at three main levels:
  
  • Infrastructure as a service
  
  • Platform as a service
  
  • Software as a service
  
  • Give a new programming models enabling an easy way for development of large-scale applications.
CONCLUSION

CLOUD

build  buy  deploy

IaaS  SaaS  PaaS

Source: based on the model provided by NIST

Essential Characteristics

- BROAD NETWORK ACCESS
- RAPID ELASTICITY
- MEASURED SERVICE
- ON-DEMAND SELF-SERVICE

RESOURCE POOLING

Service Models

- SOFTWARE AS A SERVICE
- PLATFORM AS A SERVICE
- INFRASTRUCTURE AS A SERVICE

Deployment Models

- Private
- Public
- Community
- Hybrid

44