Session Agenda

Cloud Computing – An Overview

- Introductions – Tutor, Course & Students
- Paradigms & Distributed Computing
- origins & Motivation for Cloud
- What is Cloud Computing
- Is Cloud Computing for me?

Types of Clouds & Service Deployment

- 3-4-5 Rule of Cloud Computing
- Cloud Infrastructure & Deployment

Wrap Up

- Perceived benefits of Cloud ecosystem
- Challenges to Overcome
- Advantages & Disadvantages
- Commercial offering of Cloud services
- Reality Check

Introductions

About Myself

- Name : Arun Vadekkedhil
- Profession : Solutions Architect
- Interests : Tutoring, writing
- Contact : 9881300394

Graduated from BITS Pilani in 1995, have over 24 years of IT experience, Tutor for the past 8 years

About the Course

- Units: 5

Objective :

No	Course Objective
CO1	Students will learn the fundamental ideas behind Cloud Computing, the evolution of the paradigm, its applicability; benefits, as well as current and future challenges;
CO2	Students will learn the basic ideas and principles in data centre design and management
CO3	Students will learn about cloud components and technologies and relevant distributed file systems
CO4	Students will learn a variety of programming models and develop working experience

Evaluation Components

2 Quizzes.

2 Written Exams

1 Assignment

What is Cloud Computing

Facts about Cloud

Here are our top cloud facts for 2022:

1. By 2022, more than 90% of enterprises will rely on a hybrid cloud environment to meet their infrastructure needs. **(Source)**
2. The value of the cloud computing market is estimated to be $832.1 billion by the end of 2025 (compared to $371.4 billion in 2020). **(Source)**
3. Cloud security is a top concern for 75% of enterprises. **(Source)**
4. By 2025 there will be over 100 Zettabytes of data stored in the cloud (1 zettabyte = 1 billion terrabytes = 1 trillion gigabytes). **(Source)**
5. 50% of companies reported higher cloud usage than planned during the Covid-19 pandemic. **(Source)**
6. 61% of organizations want to optimize cloud spend, making it the top initiative for the 5th year in a row. **(Source)**
7. The public cloud sector is expected to generate $331 billion in revenue by 2022 (up from $175.8 billion in 2018). **(Source)**
8. Spending on IT infrastructure is predicted to reach $55.7 billion by 2022. IDC predicts a 10.9% growth in demand for servers, switchers, and storage solutions.
9. Platform-as-a-Service (PaaS) grew in adoption to 56% in 2021, making it the fastest growing segment in cloud platforms. **(Source)**
10. 93% of businesses have a multi-cloud strategy. As these deployments mature, cost containment and cybersecurity will be top priorities. **(Source)**

Why is it Called Cloud

Cloud computing, eponymously is named after the cloud symbol used in architecture documents.

By now, you must be aware that this has no relation to its namesake from the meteorology department, but It simply means that we are using the internet to store data on remote servers, rather than storing them locally on our hard disks.

Many types of Cloud Computing Applications Exist. Depending on the need of your IT the type of cloud solution may vary.

Cloud Computing is up to 40 times more cost-effective for an SMB compared to running its own IT system or department.

Origins

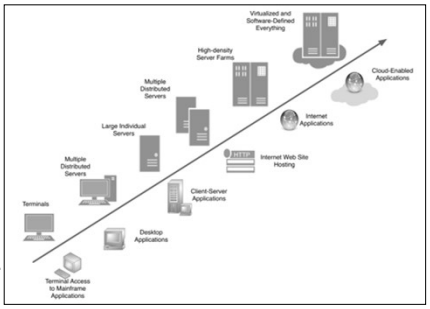
We are at a major inflexion point in computing today.

Traditional computational models are now passé.

Amazon started the concept by renting spare computing power from their retail business.

Web and mobile technologies have resulted in information explosion. This means traditional computing power is not enough to process data.

To solve this challenge, we need to use large scale distributed networks. These distributed networks evolved to the Cloud technology.

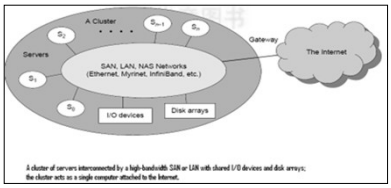


Paradigms

A *paradigm* is a standard, perspective, or set of ideas. A *paradigm* is a way of looking at something.

Types of Distributed Computing

- **Parallel Computing** : Parallel computing is a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently ("in parallel")
- **Cluster** : A cluster is a group of loosely coupled computers that work together closely, so that in some respects they can be regarded as a single computer.
- **Grids** : Grid computing is the most distributed form of parallel computing. It makes use of computers communicating over the Internet to work on a given problem.



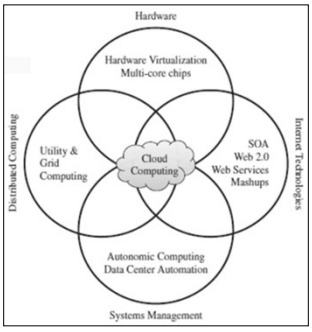
Paradigms

A *paradigm* is a standard, perspective, or set of ideas. A *paradigm* is a way of looking at something.

Cloud is the convergence of several traditional computing technologies

- Web Technologies
- Web Services
- SOA (Service Oriented Architectures)
- Distributed Computing
  - Grids
  - Clusters

We are now experiencing computing as a service provided by professional organizations, which is served through the medium of high speed internet.

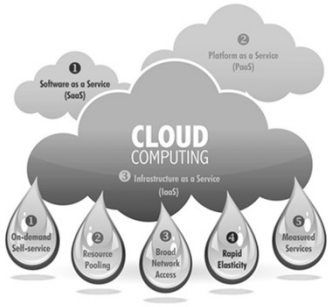


How do we define Cloud ?

innovate achieve lead

To summarize, cloud computing is the result of the mash-up of several existing disparate technologies, which were progressively modified to suit contemporary computing requirements.

- The applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards comes under Cloud computing. Cloud computing converts the technology, services, and applications that are similar to those on the Internet into a self-service utility Communicate and coordinate actions by passing messages.
- Cloud computing is based on the concept of pooling physical resources and presenting them as a virtual resource. This computing model supports a new way of provisioning resources, staging applications and for using applications.
- It's bringing computing on an internet scale.



Terms to Remember

innovate achieve lead

Cloud computing solely exists because of Virtualization technology & Abstraction

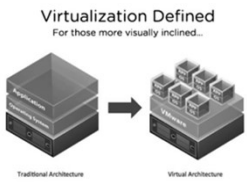
Abstraction :

The details of system implementation is hidden from users and developers.

Applications run on unspecified physical systems with unknown locations for data, with outsourced system administration of systems.

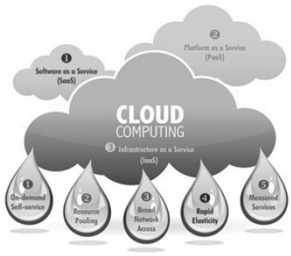
Virtualization:

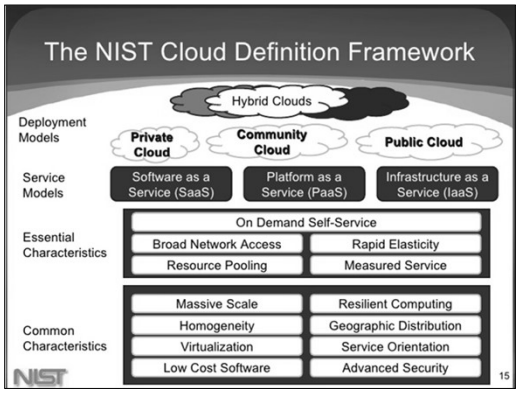
The resourced are pooled and shared among the users giving them the illusion that they are the sole owner of the resource. Also resources scales up/down in really short time and without human intervention, charged on metered basis, with multi-tenancy support.



NIST 3-4-5 Rule for Cloud

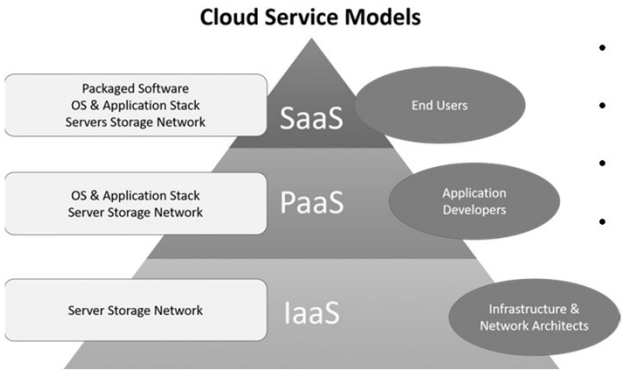
- **3** cloud service models or service types for any cloud platform
- **4** Deployment models
- **5** Essential characteristics of cloud computing infrastructure





The applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards comes under Cloud computing.

Cloud Service Models (AAS es)



- There are 3 service models
- Infrastructure as a Service
- Platform as a Service
- Software as a Service

Infrastructure as a Service

innovate achieve lead

Why IaaS

Capability

Iaas provides the following
- Servers- compute, machines
- Storage
- Network
- Operating system

Characteristics

Resources are distributed as a service
- Allows dynamic scaling (1...10....100.....)
- Has a variable costs-
- Generally includes multiple-users on a single piece of hardware. (**multi-tenancy)**

Enabler : Virtualization Technology
- ✓ Manageability and Interoperability
- ✓ Availability and Reliability
- ✓ Scalability and Elasticity

Models

IaaS can be obtained as
(1) Public or
(2) Private infrastructure or
(3) combination of both

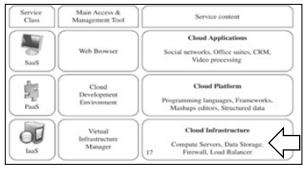
Benefit

The user instead of **purchasing-** servers, software, **data center space** or network equipment, **rent** those resources as a fully **outsourced** service on-demand model.

Infrastructure as a Service - Definition

innovate achieve lead

- **The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources**.
- The consumer can deploy and run 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 limited control of select networking components (e.g., host firewalls).
- **Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS).**
- Infrastructure services are considered to be the bottom layer of cloud computing systems .
- Ex : Amazon EC2 : Elastic Compute Cloud, Eucalyptus, GoGrid, Rackspace Cloud



BITS Pilani, Pilani Campus

Infrastructure as a Service - Applicability

Where IaaS helps

1. Where demand is very **volatile**- encountering **spikes and troughs.**
2. For new enterprise without **capital to invest in hardware or** entrepreneurs starting on a shoestring budget.
3. Where the enterprise is **growing rapidly** and **scaling hardware** would be problematic.
4. For **specific line** of business, **trial** or **temporary** infrastructural needs
5. When you need **computing power on the go**, turn to IaaS.

What to Do with IaaS

Test and development. Teams can quickly set up and dismantle test and development environments, bringing new applications to market faster.

Website hosting. Running websites using IaaS can be less expensive than traditional web hosting.

Storage, backup and recovery. Organizations avoid the capital outlay. IaaS is useful for handling unpredictable demand and steadily growing storage needs. It can also simplify planning and management of backup and recovery systems

Big data analysis. Mining data sets to locate or tease out these hidden patterns requires a huge amount of processing power, which IaaS economically provides.

Platform as a Service - Overview

Why PaaS

Capability

Paas provides the following
- Tools to build applications
- Scripting Environment
- Database Platform

Characteristics

Collaborative platform for application development using workflows.
Platform which allows creation of proprietary data or application

Enabler : **Runtime Environment Design**
- ✓ Fault Tolerant Design
- ✓ Containerization
- ✓ Avoiding DLL Hell
- ✓ Secure

Models

PaaS can be obtained as
(1) Public or
(2) Private infrastructure or
(3) combination of both

Benefit

Development tools served up on a Platter a-la carte

No need to worry about upgrading to newer platforms or worry about license costs

Platform as a Service - Definition

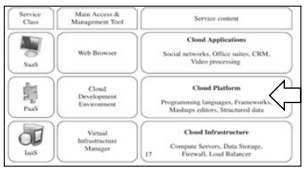
•**The capability provided to the consumer is to deploy onto the cloud infrastructure, consumer-created or acquired applications created using programming languages and tools supported by the provider**.

•The consumer **does not manage** or control the underlying **cloud infrastructure** including **network, servers, operating systems,** or **storage,** but has control over the **deployed applications** and possibly application **hosting** environment **configurations**.

•A **PaaS platform** offers an environment on which developers **create** and deploy **applications** and do not **necessarily** need to **know** how many **processors** or how much **memory** that **applications** will be **using**.

• In addition, multiple programming models and **specialized services** (e.g., data access, authentication, and payments) are offered as building blocks to new applications.

•Google AppEngine, Azure, Force.com an example of Platform as a Service



Platform as a Service - Applicability

Where PaaS helps

PaaS allows developers to frequently **change** or **upgrade** operating system features.

•It also helps development teams **collaborate** on projects.

•**Security** is provided, including data security and backup and recovery.

•Adaptability; Features can be **changed** if circumstances dictate that they should.

•**Flexibility**; customers can have control over the tools that are installed within their platforms and can create a platform that suits their specific requirements.

What to Do with PaaS

- **Application Services**
  - •Services to develop, test, deploy, host and maintain applications in the same integrated development environment.
  - •Web based user interface creation tools help to create, modify, test and deploy different UI scenarios
- **Multi Tenancy**
  - •Construct architecture where multiple concurrent users utilize the same development application
- **Collaboration**
  - •Support for development team collaboration
  - •Tools to handle billing and subscription management

Software as a Service - Overview

Why SaaS

Capability

Saas provides the following
- Hosted , Finished Product
- Subscription to Services

Characteristics

Not owned, but subscribed to from an external service provider.
Designed to be Multi Tenant
Customer Configuration instead of Application configuration
Centralized management

Enabler : **Web Service**
- ✓ Accessibility & Portability

Models

SaaS can be obtained as
(1) Managed Service
(2) Monthly Subscription

Benefit

No need to purchase Software or Licenses upfront
Centralized management helps SaaS Vendors
Lower TCO for users

Software as a Service - Definition

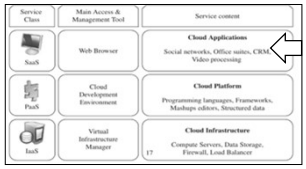
•The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure.

•The applications are accessible from various client devices through a thin client interface such as a web browser (e.g., web-based email).

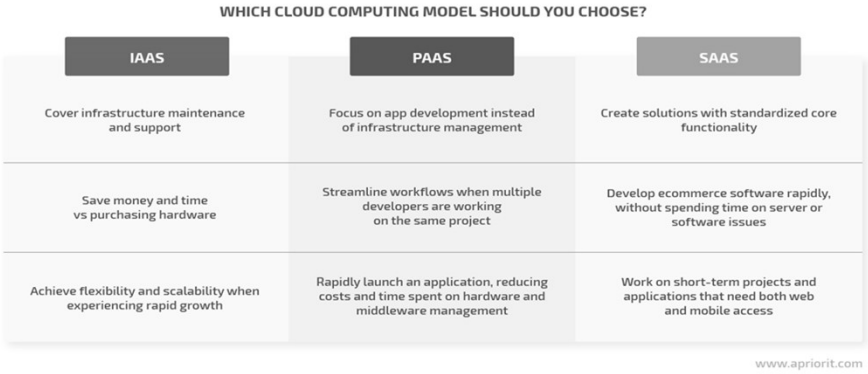
•The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

•This model of delivering applications, known as Software as a Service (SaaS), alleviates the burden of software maintenance for customers and simplifies development and testing for providers.

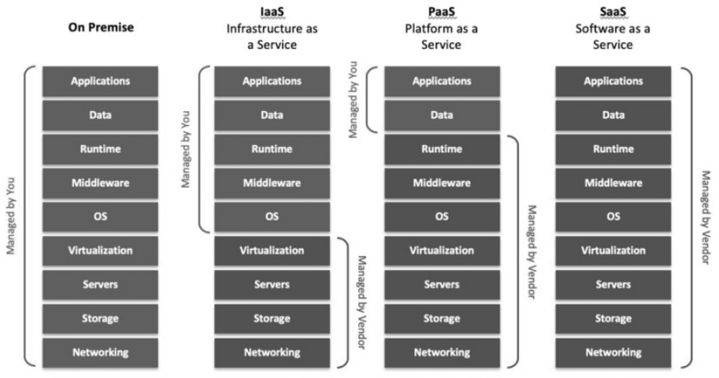
•Salesforce.com, SaaS model, offers business productivity applications (CRM) that reside completely on their servers, allowing costumers to customize and access applications on demand.



Which AAS ?



Who Manages the AAS es?

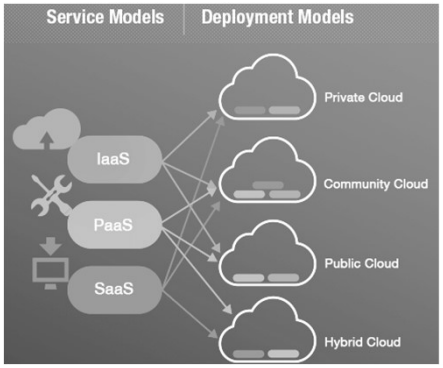


Deployment Models in Cloud

There are four primary cloud deployment models :

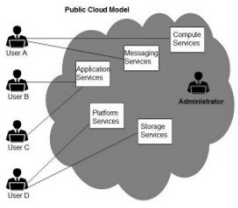
- Public Cloud
- Private Cloud
- Community Cloud
- Hybrid Cloud

Each can exhibit the previously discussed characteristics; their differences lie primarily in the scope and access of published cloud services, as they are made available to service consumers.

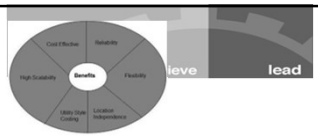


Pubic Cloud

- **Public cloud** is a cloud infrastructure **owned by a cloud service provider** that provides cloud services to the public for **commercial purposes**.
- Cloud infrastructure available for public consumption on a **pay per use** basis.
- Examples of public clouds include **Amazon Elastic Compute Cloud (EC2),** IBM's Blue Cloud, Sun Cloud, Google AppEngine and Windows Azure Services Platform.
- **Characteristics**
  - ✓ Homogeneous infrastructure
  - ✓ Common policies , Shared resources and multi-tenant
  - ✓ Leased or rented infrastructure, Economies of scale



Pubic Cloud - Advantage



Cost Effective
- Since **public cloud shares** same resources with large number of customers it turns out **inexpensive**.

Reliability
- The **public cloud** employs large number of **resources** from different locations. If any of the resources fails, public cloud can **employ** another one.

Flexibility
- The public cloud can **smoothly integrate** with **private** cloud, which gives customers a **flexible** approach.

Location Independence
- **Public cloud** services are delivered through Internet, ensuring location independence.

Utility Style Costing
- Public cloud is also based on **pay-per-use** model and resources are accessible whenever customer needs them.

High Scalability
- Cloud resources are made available on demand from a **pool of resources**, i.e., they can be scaled up or down according the requirement.

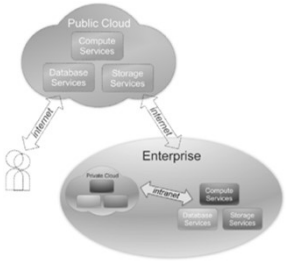
Pubic Cloud - Disadvantage

Low Security
- In **public cloud model,** data is hosted off-site and resources are shared publicly, therefore does not ensure higher level of security.

Less Customizable
- It is comparatively less customizable than private cloud.

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**.
- Also referred to as **internal cloud** or on-premise cloud, a private cloud **intentionally limits access** to its resources to **service consumers** that belong to the same organization that owns the cloud.
- Characteristics :
  - Heterogeneous infrastructure
  - Customized and tailored policies
  - Dedicated resources
  - In-house infrastructure
  - End-to-end control



Private Cloud - Advantage

High Security and Privacy

- **Private cloud** operations are not available to general public and resources are shared from distinct pool of resources. Therefore, it ensures high **security** and **privacy.**

More Control

- The **private cloud** has more control on its resources and hardware than public cloud because it is accessed only within an organization.

Cost and Energy Efficiency

- The **private cloud** resources are not as cost effective as resources in public clouds but they offer more efficiency than public cloud resources.

Private Cloud - Disadvantage

Restricted Area of Operation

- The private cloud is only accessible locally and is very difficult to deploy globally.

High Priced

- Purchasing new hardware in order to fulfill the demand is a costly transaction.

Limited Scalability

- The private cloud can be scaled only within capacity of internal hosted resources.

Additional Skills

- In order to maintain cloud deployment, organization requires skilled expertise.

Hybrid Cloud

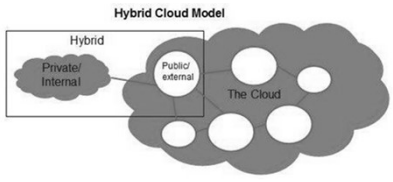
•**Hybrid clouds** are **mixtures** of these **different deployments**.

•For example, an enterprise may **rent storage** in a **public** cloud for handling peak demand.

• The **combination** of the enterprise's **private cloud** and the **rented storage** then is a hybrid cloud.

•Clouds retain their **unique identities**, but are **bound together as a unit**.

•A hybrid cloud may offer **standardized** or **proprietary** access to data and applications, as well as application portability.



Hybrid Cloud - Advantage

Scalability

- It offers features of both, the public cloud scalability and the private cloud scalability.

Flexibility

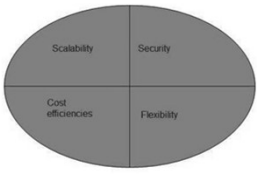
- It offers secure resources and scalable public resources.

Cost Efficiency

- Public clouds are more cost effective than private ones. Therefore, hybrid clouds can be cost saving.

Security

- The private cloud in hybrid cloud ensures higher degree of security.



Hybrid Cloud - Disadvantage

Networking Issues

- Networking becomes complex due to presence of private and public cloud.

Security Compliance

- It is necessary to ensure that cloud services are compliant with security policies of the organization.

Infrastructure Dependency

- The **hybrid cloud model** is dependent on internal IT infrastructure, therefore it is necessary to ensure redundancy across data centers.

Community Cloud

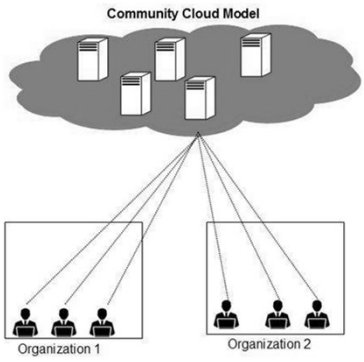
innovate achieve lead

•**Community Cloud** is a **cloud infrastructure** shared by a **community** of **multiple organizations** that generally have a **common purpose**.

•An example of a community cloud is **OpenCirrus,** which is a cloud **computing research testbed** intended to be used by **universities** and **research institutions**.

•It may be for **one organization** or for **several organizations,** but they **share common concerns** such as their mission, policies, security, regulatory compliance needs, and so on.

•A **community cloud** may be managed by the **constituent organization(s) or by a third party**.



Community Cloud - Advantage

innovate achieve lead

Cost Effective

- **Community cloud** offers same advantages as that of private cloud at low cost.

Sharing Among Organizations

- Community cloud provides an infrastructure to share cloud resources and capabilities among several organizations.

Security

- The community cloud is comparatively more secure than the public cloud but less secured than the private cloud.

Community Cloud - Disadvantage

innovate achieve lead

Logistics Issues

•Since all data is located at one place, one must be careful in storing data in community cloud because it might be accessible to others.

•It is also challenging to allocate responsibilities of governance, security and cost among organizations.

Quick Comparison

innovate achieve lead

Parameters\Type	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
Description	In public cloud, services are available for public users.	Private cloud is build up with existing private infrastructure. This type of cloud has some authentic users who can dynamically provision the resources.	Hybrid cloud is a heterogeneous distributed system, resulting from a private cloud, which incorporates different types of services and resources from public clouds.	Different types of cloud are integrated together to meet a common or particular need for some organizations.
Scalability	Very High	Limited	Very High	Limited
Reliability	Moderate	Very High	Medium to High	Very High
Security	Totally Depends on service provider	High class security	Secure	Secure
Performance	Low to medium	Good	Good	Very Good
Cost	Cheaper	High Cost	Costly	Costly
Examples	Amazon EC2, Google AppEngine	VMWare, Microsoft, KVM, Xen	IBM, HP, VMWare vCloud, Eucalyptus	SolaS Community Cloud, VMWare

Essential Characteristics

5 Essential Characteristics of Cloud Computing

Ref: The NIST Definition of Cloud Computing
http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

On-demand self-service | Ubiquitous network access | Location transparent resource pooling | Rapid elasticity | Measured service with pay per use

Source: http://aka.ms/532

Resource pooling

- **Cloud services** can support **millions of concurrent users**; for example, Skype supports 27 million concurrent users, while Facebook supported 7 million simultaneous users in 2009.
- Clearly, it is **impossible to support** this number of users if **each user needs dedicated hardware**. Therefore, cloud services need to **share resources between users** and clients in order to **reduce costs**.

✓ **Resources** are drawn from a **common pool**.
✓ **Common resources** build **economies of scale**.
✓ **Common infrastructure** runs at **high efficiency**.
✓ Appropriate **management** of **security & privacy**.

Broad Network Access

- **Ubiquitous access** to cloud applications **from desktops, laptops to mobile devices** is critical to the success of a Cloud platform.
- Thus, **connectivity** is a **critical requirement** for effective use of a **Cloud Application**.
- For example, cloud services like Amazon, Google, and Yahoo! are available world-wide via the Internet.
- They are also accessible by a wide variety of devices, such as mobile phones, iPads, and PCs.

- ✓ Users **abstracted** from the implementation
- ✓ Near **real-time delivery** (seconds or minutes)
- ✓ **Services accessed** through a **self-serve web interface**.

On Demand Self Service

On demand self-service: The compute, storage or platform resources needed by the user of a **cloud platform** are **self-provisioned** or **auto-provisioned** with **minimal configuration**.

For example is possible to log on to Amazon Elastic Compute Cloud (a popular cloud platform) and obtain resources, such as virtual servers or virtual storage, within minutes.

- ✓ Open standards and APIs
- ✓ Almost always IP, HTTP, and REST
- ✓ Available from anywhere with an internet connection

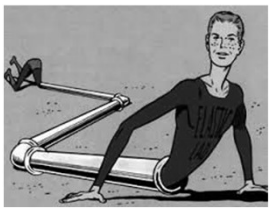
Rapid Elasticity

A **cloud platform** should be able to **rapidly increase** or **decrease** computing **resources** as needed.

Further, the time taken to provision a new server is very small, on the order of minutes.

- ✓ Resources dynamically-allocated between users.
- ✓ Additional resources dynamically-released when needed.
- ✓ Fully automated

This also increases the speed with which a new infrastructure can be deployed.



Metered by Use

One of the **compelling business** use cases for cloud computing is the ability to "**pay as you go**," where the **consumer pays** only for the **resources** that are **actually used** by his applications.

Commercial cloud services, like Salesforce.com, measure resource usage by customers, and charge proportionally to the resource usage.

- ✓ Services can be cancelled at any time
- ✓ Pay as you go approach

Cloud Advantages

innovate achieve lead

Reduced costs : Significant cost reductions are achieved due to higher efficiencies and greater utilization of cloud networks

Ease of utilization: The upfront cost involved in the purchase of hardware and software licenses is lowered a lot. Due to that one can easily make utilization of cloud services.

Quality of Service: Service level agreements with vendor assure the Quality of service

Reliability: The resource scaling and load balancing with fault tolerance capabilities emphasize the high availability of systems.

Outsourced IT management: It results into considerable reduction in IT management complexities and the associated cost.

Simplified maintenance and upgrade: Always latest features are provided to the users removing the need of constant update and up gradations.

Low Entry Barrier: Upfront infrastructure investments are not needed for moving to the cloud.

Cloud vs Hosted

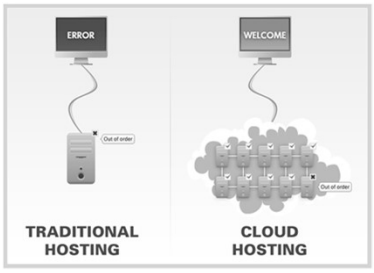
innovate achieve lead

Cloud apps are **web apps** in the sense that they can be used through **web browsers** but not all web apps are cloud apps.

For your **web app** to evolve into a **cloud app,** it should exhibit **certain properties** such as True multi-tenancy to support unique requirements & needs for individual consumers.

Support for virtualization technology, which plays a starring role for cloud era apps.

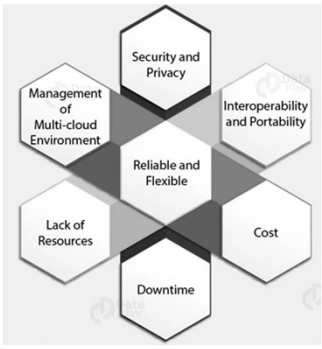
Web applications should either be built to support this or re-engineered to do so



Cloud Challenges

Security and Privacy of Cloud

- The data store in the cloud must secure and provide full **confidentiality**. The **cloud provider** should take necessary **security measures** to **secure** the data of the customers.
- Securities are also the **responsibility of the customer** as they should provide a strong password, should not share the password with anyone, and regularly change the password when we did.
- **Hacking** can lead to **data loss;** disrupt the encrypted file system and many other problems.



Cloud Challenges

Interoperability and Portability

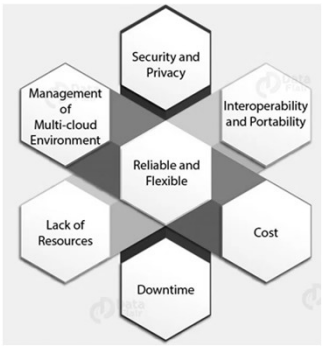
- The customer must be provided with the services of migration in and out of the cloud – no holds barred.

Reliable and Flexible

- Reliability and flexibility means that the **data provided** to the cloud should **not leak** and the host **should provide trust to the customers**.
- To eliminate this challenge the **services provided** by the third party should be **monitored** and **supervision** should be done on **performance, robustness** and business dependency.

Cost

- Cloud computing is **affordable** but tailor-made deployment based on customer's demand can be **expensive**. Use **Multitenancy to minimize costs**



Cloud Challenges

innovate achieve lead

Downtime

- Downtime is the common challenges of cloud computing as no cloud provider guarantees a platform that is free from downtime. **Apply redundancy and or DR to minimize**.

Lack of resources

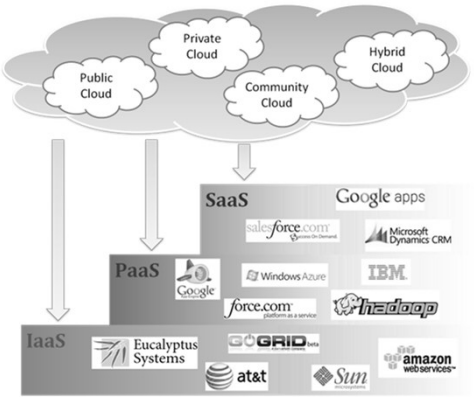
- Lack of resources and expertise is also one of the major challenges faced by the cloud industry and many companies are hoping to overcome this challenge by hiring more workers which are more experienced. **Use Automation**

Management of Multi-Cloud Environment

- Companies nowadays do not use a **single cloud** instead they are using **multiple clouds**. On an average company are using 4.8 different **public** and **private clouds** due to which their management is hindered. **Invest in a good Cloud monitoring tool**

Cloud ecosystem

innovate achieve lead



What is cloud computing in your mind

Clear or Cloudy?

Cloud computing is a new paradigm shift in the way we make use of computing resources.

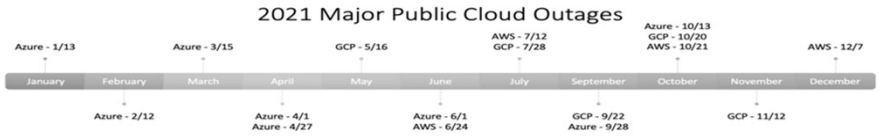
Cloud computing can provide high quality of service at perceived cost benefits

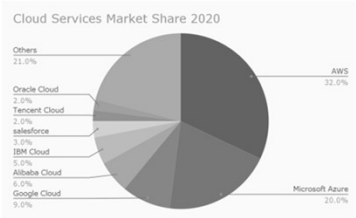
We are moving to an era of computing where we can use code to setup infrastructure on an ad-hoc basis.

Service models and deployment models provide services that can be used to

- Rent fundamental computing resources
- Deploy and develop customer-created applications on clouds
- Access provider's applications over network (wired or wireless)

Cloud Failures





Date	Descritpion
12th Dec	One of the mission-critical AWS cloud units us-east-1 was hit with an outage that took down services like Disney+, Netflix, Slack, Ticketmaster, stock trading app Robinhood, and the crypto exchange Coinbase. Key internal tools like Flex and AtoZ apps used by Amazon warehouse and delivery workers were affected as well.
12th Nov	Google Cloud went down in mid-November and with it took services like Home Depot, Snap, and Spotify. What caused the outage? A glitch in a network configuration. Yet another scenario showing that betting on a single provider to manage all your apps is pretty risky
20th Oct	Facebook and its subsidiaries – Messenger, Instagram, WhatsApp, Mapillary, and Oculus – became unavailable for 6 to 7 hours and the world went crazy. Many desperate users flocked to Twitter, Discord, Signal, and Telegram and this resulted in disruptions on these apps' servers.

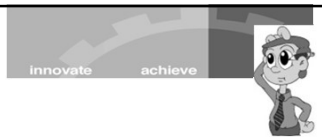
https://www.crn.com/slide-shows/cloud/the-10-biggest-cloud-outages-of-2021-so-far-

Agenda

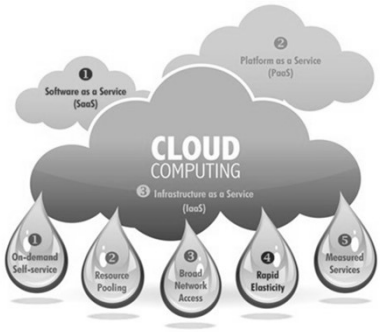
- Cloud Recap
  - What is NIST 3-4-5 Rule
  - Advantages of Cloud
  - Disadvantages
- Introduction to Virtualization
  - What is Virtualization
  - Use & demerits of Virtualization
  - Introducing the Hypervisor
  - Purpose, Design Goals & Types of Hypervisor
- Virtualization
  - Types of Virtualization
  - X86 Hardware Virtualization
  - NFV - VNF

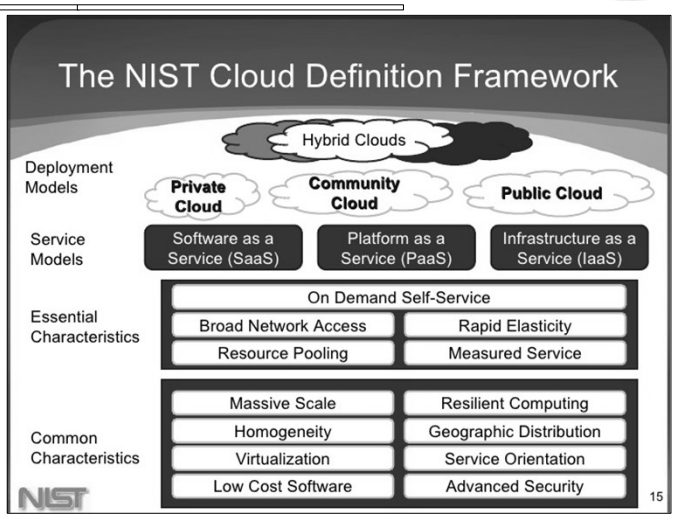
50

NIST Definitions



- **3** cloud service models or service types for any cloud platform
- **4** Deployment models
- **5** Essential characteristics of cloud computing infrastructure





Motivations & Origins

Motivation

1 machine → 1 OS → several applications

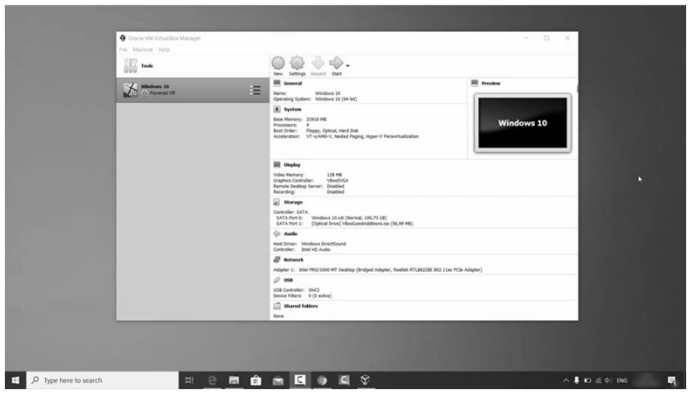
Applications can affect each other

Big disadvantage: machine utilization is very low, most of the times it is below than 25%

Origins

- Server virtualization has existed for several decades
- IBM pioneered more than 30 years ago with the capability to "multitask"
- The inception was in specialized, proprietary, high-end server and mainframe systems. By 1980/90 servers virtualization adoption reduced
- Inexpensive x86 hardware platforms
- Windows/Linux adopted as server

Video – Virtualization



Learning Objectives

- Introduce Oracle Virtual Box, a hosted hypervisor.
- Demonstrate what a host system is what a guest VM is and what is the role of the hypervisor.
- Students will use the same as home work and install virtual box and a choice of their own OS after class.

What is Virtualization?

innovate achieve lead

Virtualization Defined

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.

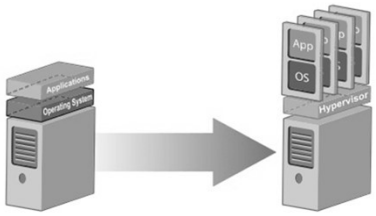
Virtualization allows multiple operating system instances to run concurrently on a single computer

Instead of purchasing and maintaining an entire computer for one application, each application can be given its own operating system, and all those operating systems can reside on a single piece of hardware.

Virtualization allows an operator to control a guest operating system's use of CPU, memory, storage, and other resources, so each guest receives only the resources that it needs.

Key Terms:

- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created

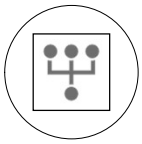


What is Virtualization?

innovate achieve lead

Virtualization Objectives

ABSTRACTION – TO SIMPLIFY THE USE OF THE UNDERLYING RESOURCE (E.G., BY REMOVING DETAILS OF THE RESOURCE'S STRUCTURE)



REPLICATION – TO CREATE MULTIPLE INSTANCES OF THE RESOURCE (E.G., TO SIMPLIFY MANAGEMENT OR ALLOCATION)

ISOLATION – TO SEPARATE THE USES WHICH CLIENTS MAKE OF THE UNDERLYING RESOURCES (E.G., TO IMPROVE SECURITY)

Key Terms:
- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created

What is Virtualization?

innovate achieve lead

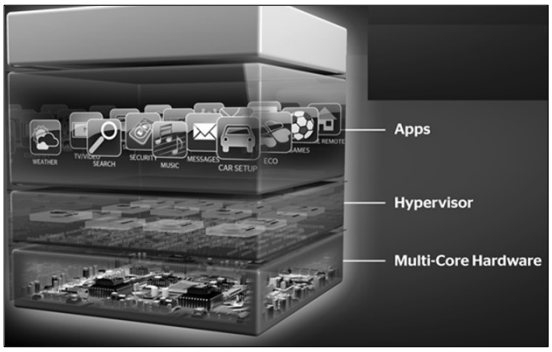
Need of Virtualization

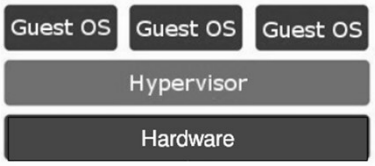
- Cloud can exist without Virtualization, although it will be difficult and inefficient.
- Cloud makes notion of "Pay for what you use", "infinite availability- use as much you want".
- These notions are practical only if we have
  - lot of flexibility
  - efficiency in the back-end.
- This efficiency is readily available in Virtualized Environments and Machines

Key Terms:
- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created

Virtualization Architecture

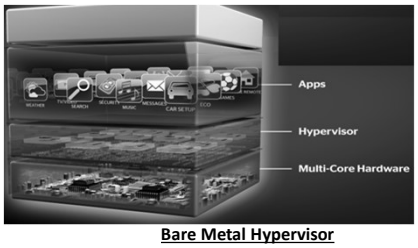
- OS assumes complete control of the underlying hardware.
- Virtualization architecture provides this illusion through a hypervisor/VMM.
- Hypervisor/VMM is a software layer which:
- Allows multiple Guest OS (Virtual Machines) to run simultaneously on a single physical host
- Provides hardware abstraction to the running Guest OSs and efficiently multiplexes underlying hardware resources



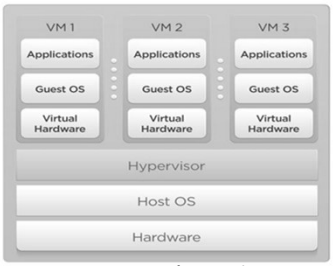


Hypervisor

- A **hypervisor** or **virtual machine monitor** (**VMM**) is computer software, firmware, or hardware. VMM creates and runs virtual machines.
- A computer on which a hypervisor runs one or more virtual machines is called a *host machine*,
- Each virtual machine is called a *guest machine*
- The hypervisor presents the guest systems with a virtual operating platform and manages the execution of the guest operating systems.
- Multiple instances of a variety of operating systems may share the virtualized hardware resources:



Bare Metal Hypervisor



Hosted Hypervisor

Hypervisor Goals

The main function of the software layer for virtualization is to **virtualize the physical hardware** of the host machine to virtual resources to be used by guest VM's

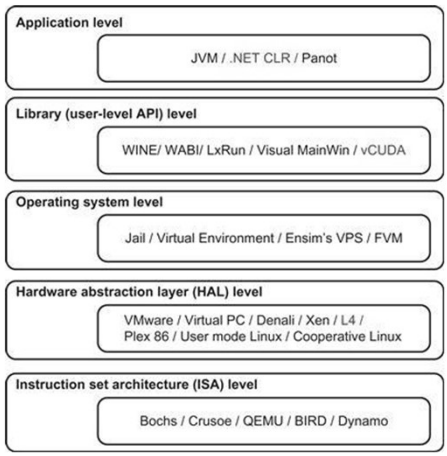
The virtualization software creates the abstraction of VM's by interposing a virtual layer at various levels of the computer system.

At the ISA level, virtualization is performed by emulating a given ISA on the ISA of the host machine.

For example, MIPS binary code can run on an x86-based host machine with the help of ISA emulation.

With this approach, it is possible to run a large amount of legacy binary code written for various processors on any given new hardware host machine.

Instruction set emulation leads to virtual ISAs created on any hardware machine.

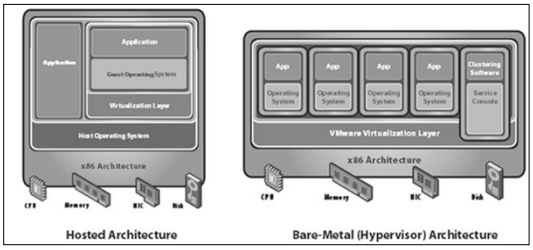


Hypervisor - Samples

- **BOCHS** :
  - Bochs is a portable IA-32 and x86-64 IBM PC compatible emulator and debugger mostly written in C++ and distributed as free software under the GNU Lesser General Public License.
  - It supports emulation of the processor, memory, disks, display, Ethernet, BIOS and common hardware peripherals of PCs.
- **BSD Jail** :
  - The jail mechanism is an implementation of FreeBSD's OS-level virtualisation that allows system administrators to partition a FreeBSD-derived computer system into several independent mini-systems called jails, all sharing the same kernel, with very little overhead.

Hypervisor Types

- **Hosted**: A **hosted** architecture installs and runs the virtualization layer as an application on top of an operating system and supports the broadest range of hardware configurations. (VMware Player, ACE)
- **Bare Metal** : The architecture installs the virtualization layer directly on a clean x86-based system. Since it has direct access to the hardware resources rather than going through an operating system, a hypervisor is more efficient than a hosted architecture and delivers greater scalability, robustness and performance. (ESX Server)
- **Hybrid**: The architecture installs the VM layer directly on the hardware like a bare metal, but also leverages the features of the host OS. Xen and Microsoft's Hyper-V are examples of hybrid hypervisors



Design Goals

- **Reliability**
  - Minimal code base
  - Strictly layered design
  - Not extensible
- **Isolation**
  - Security isolation
  - Fault isolation
  - Resource isolation
- **Scalability**
  - Scale to large number of cores
  - Large memory systems

Hypervisor Architecture

Monolithic hypervisor

- Simpler than a modern kernel, but still complex
- Contains its own drivers model

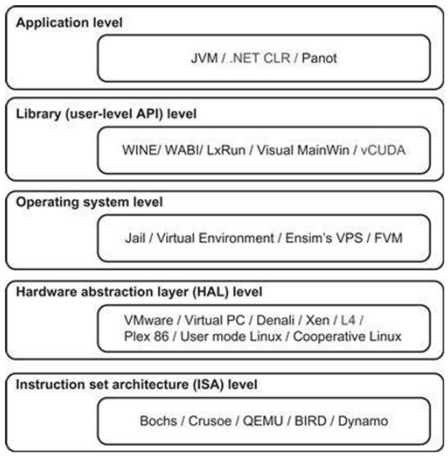
Microkernel hypervisor

- Simple partitioning functionality
- Increase reliability and minimize lowest level of the TCB
- No third-party code
- Drivers run within guests

BASIS FOR COMPARISON	MICROKERNEL	MONOLITHIC KERNEL
Basic	In microkernel user services and kernel, services are kept in separate address space.	In monolithic kernel, both user services and kernel services are kept in the same address space.
Size	Microkernel are smaller in size.	Monolithic kernel is larger than microkernel.
Execution	Slow execution.	Fast execution.
Extendible	The microkernel is easily extendible.	The monolithic kernel is hard to extend.
Security	If a service crashes, it does effect on working of microkernel.	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel, more code is required.	To write a monolithic kernel, less code is required.
Example	QNX, Symbian, L4Linux, Singularity, K42, Mac OS X, Integrity, PikeOS, HURD, Minix, and Coyotos.	Linux, BSDs (FreeBSD, OpenBSD, NetBSD), Microsoft Windows (95,98,Me), Solaris, OS-9, AIX, HP-UX, DOS, OpenVMS, XTS-400 etc.

Comparison

Level of Implementation	Higher Performance	Application Flexibility	Implementation Complexity	Application Isolation
ISA	X	XXXXX	XXX	XXX
Hardware-level virtualization	XXXXX	XXX	XXXXX	XXXX
OS-level virtualization	XXXXX	XX	XXX	XX
Runtime library support	XXX	XX	XX	XX
User application level	XX	XX	XXXXX	XXXXX



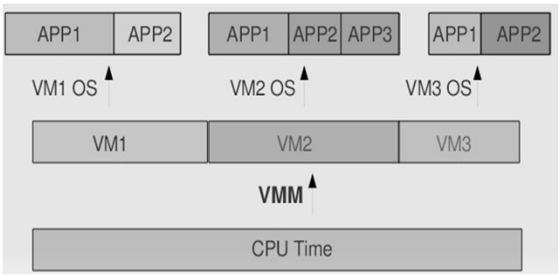
The number of X's in the table cells **reflects the advantage points** of each implementation level. **Five X's implies the best case** and **one X implies the worst case**.
Overall, **hardware and OS support** will yield the **highest performance**. However, the hardware and application levels are also the most expensive to implement. User isolation is the most difficult to achieve. ISA implementation offers the best application flexibility.

Resource Sharing in VM - CPU

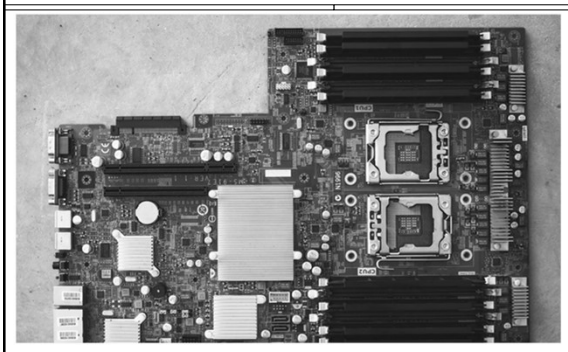
VMM or Hypervisor provides a virtual view of CPU to VMs.

In multi processing, CPU is allotted to the different processes in form of time slices by the OS.

Similarly VMM or Hypervisor allots CPU to different VMs.



Resource Sharing in VM - CPU



A CPU Socket is a physical connector on the motherboard to which a single physical CPU is connected.

A CPU (central processing unit, microprocessor chip, or processor) is a computer component. It is the electronic circuitry with transistors that is connected to a socket.

A CPU core is the part of a processor(CPU) containing the L1 cache. The CPU core performs computational tasks independently without interacting with other cores and external components of a "big" processor that are shared among cores. Basically, a core can be considered as a small processor built into the main processor that is connected to a socket. Applications should support parallel computations to use multicore processors rationally.

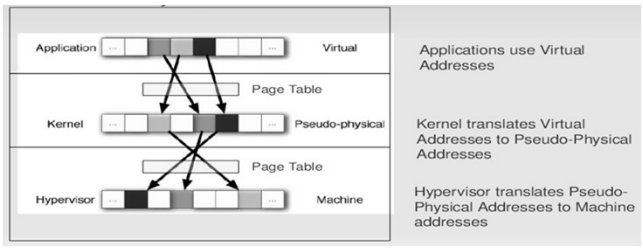
Hyper-threading is a technology developed by Intel engineers to bring parallel computation to processors that have one processor core. The debut of hyper-threading was in 2002 when the Pentium 4 HT processor was released and positioned for desktop computers. An operating system detects a single-core processor with hyper-threading as a processor with two logical cores (not physical cores). Similarly, a four-core processor with hyper-threading appears to an OS as a processor with 8 cores.

A vCPU is a virtual processor that is configured as a virtual device in the virtual hardware settings of a VM. A virtual processor can be configured to use multiple CPU cores. A vCPU is connected to a virtual socket.

Resource Sharing in VM - Memory

In Multiprogramming there is a single level of indirection maintained by Kernel.

In case of Virtual Machines there is one more level of indirection maintained by VMM



Memory sharing relies on the observation that several virtual machines might be running instances of the same guest operating system.

These virtual machines might have the same applications or components loaded, or contain common data.

In such cases, a host uses a proprietary Transparent Page Sharing (TPS) technique to eliminate redundant copies of memory pages.

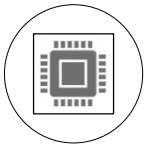
With memory sharing, a workload running on a virtual machine often consumes less memory than it might when running on physical machines.

As a result, higher levels of overcommitment can be supported efficiently.

The amount of memory saved by memory sharing depends on whether the workload consists of nearly identical machines which might free up more memory.

A more diverse workload might result in a lower percentage of memory savings.

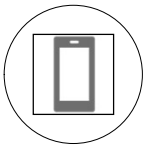
Resource Sharing in VM - IO



Device needs to use Physical Memory location.

In a virtualized environment, the kernel is running in a hypervisor-provided virtual address space

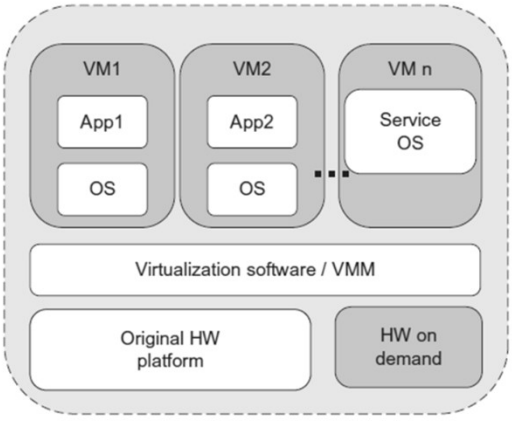
Allowing the guest kernel to convey an arbitrary location to device for writing is a serious security hole



Each device defines its own protocol for talking to drivers

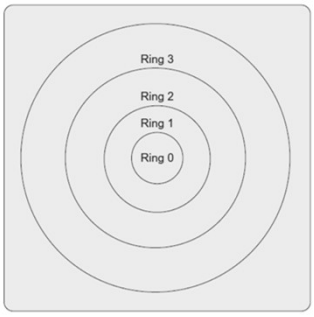
Hypervisor Techniques

- At a very high level, all three types of hypervisors described earlier operate in a similar manner.
- In each case, the guests continue execution until they try to access a shared physical resource of the hardware (such as an I/O device), or an interrupt is received.
- When this happens, the hypervisor regains control and mediates access to the hardware, or handles the interrupt.



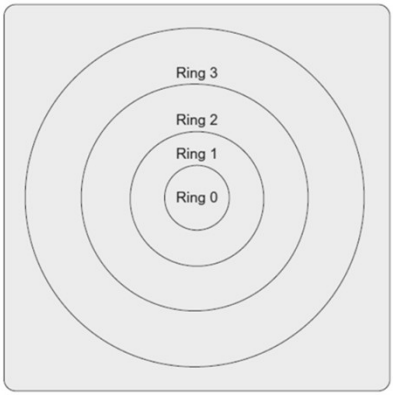
Hypervisor Techniques

- To accomplish this functionality, hypervisors rely on a feature of modern processors known as the privilege level or protection ring.
- The basic idea behind privilege levels is that all instructions that modify the physical hardware configuration are permitted at the highest level,
- At lower levels, only restricted sets of instructions can be executed.
- There are four rings, numbered from 0 to 3.
- Programs executing in Ring 0 have the highest privileges, and are allowed to execute any instructions or access any physical resources such as memory pages or I/O devices.
- Guests are typically made to execute in ring 3. This is accomplished by setting the Current Privilege Level (CPL) register of the processor to 3 before starting execution of the guest.



Hypervisor Techniques

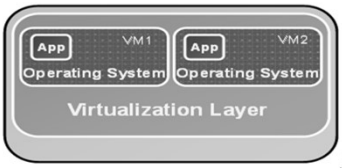
- If the guest tries to access a protected resource, such as an I/O device, an interrupt takes place, and the hypervisor regains control.
- The hypervisor then emulates the I/O operation for the guest.
- The exact details depend upon the particular hypervisor (e.g., Xen or Hyper-V).
- Note that in order to emulate the I/O operation, it is necessary for the hypervisor to have **maintained the state of the guest and its virtual resources**



Benefits of Virtualization

- Single OS image per machine
- Software and hardware tightly coupled
- Running multiple applications on same machine often creates conflict
- Underutilized resources
- Inflexible and costly infrastructure

- Hardware-independence of operating system and applications
- Virtual machines can be provisioned to any system
- Can manage OS and application as a single unit by encapsulating them into virtual machines



Virtualization Summary

•Virtualization allows multiple operating system instances to run concurrently on a single computer. It is a means of separating hardware from a single operating system.

•Each "guest" OS is managed by a Virtual Machine Monitor (VMM), also known as a hypervisor.

•Because the virtualization system sits between the guest and the hardware, it can control the guests' use of CPU, memory, and storage, even allowing a guest OS to migrate from one machine to another.

•Instead of purchasing and maintaining an entire computer for one application, each application can be given its own operating system, and all those operating systems can reside on a single piece of hardware.

•Virtualization allows an operator to control a guest operating system's use of CPU, memory, storage, and other resources, so each guest receives only the resources that it needs.

Key Terms to Remember

innovate achieve lead

Key Terms:

VM : Virtual Machine

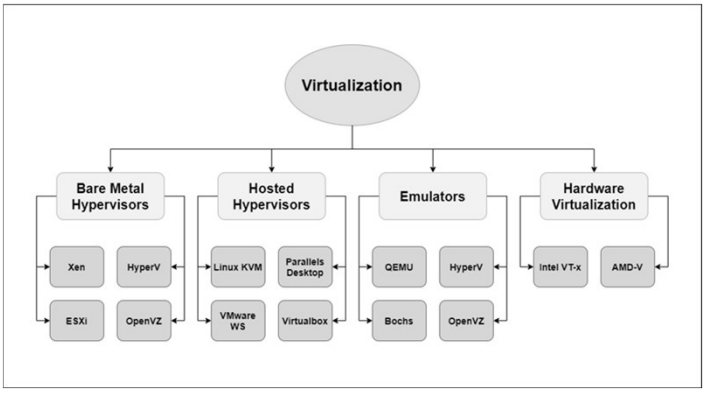
VMM: Virtual Machine Monitor

Hypervisor : VMM

Multiplexed: Many or several

Host: System where the VMM resides

Guest : Virtual Machines created



Agenda

innovate achieve lead

- Virtualization Recap
- Virtualization Approaches
  - Motivations
  - Full Virtualization
  - Para Virtualization
  - Hardware Assisted Virtualization
  - Compare & Contrast architectures
- X86 Hardware Virtualization
  - Motivation & Challenges
  - X86 Hardware Virtualization
  - NFV - VNF

What is Virtualization?

Virtualization Defined

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.

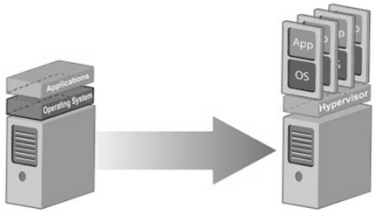
Virtualization allows multiple operating system instances to run concurrently on a single computer

Instead of purchasing and maintaining an entire computer for one application, each application can be given its own operating system, and all those operating systems can reside on a single piece of hardware.

Virtualization allows an operator to control a guest operating system's use of CPU, memory, storage, and other resources, so each guest receives only the resources that it needs.

Key Terms:

- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created



What is Hypervisor

Hypervisor Demystified

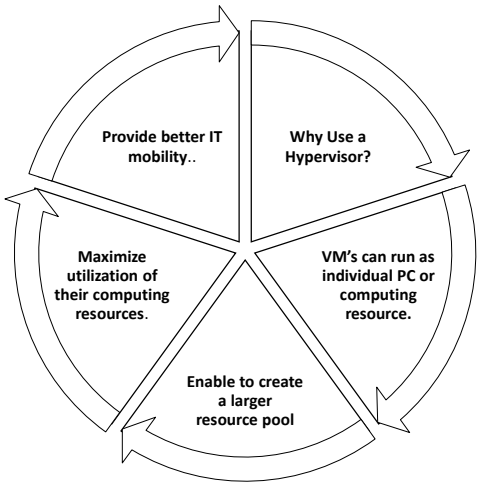
A hypervisor is a form of virtualization software used in Cloud hosting to divide and allocate the resources on various pieces of hardware.

A hypervisor is a crucial piece of software that makes virtualization possible. It creates a virtualization layer that separates the actual hardware components - processors, RAM, and other physical resources - from the virtual machines and the operating systems they run.

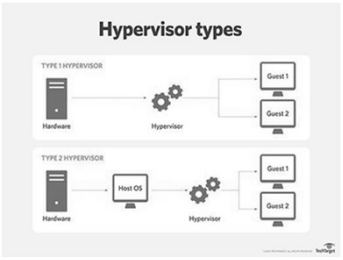
Hypervisors emulate available resources so that guest machines can use them. No matter what operating system boots up on a virtual machine, it will think that actual physical hardware is at its disposal..

From a VM's standpoint, there is no difference between the physical and virtualized environment. Guest machines do not know that the hypervisor created them in a virtual environment or that they share available computing power.

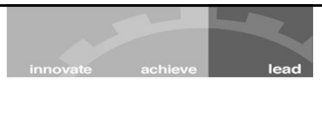
Why use Hypervisor



There are two types of hypervisors, according to their place in the server virtualization structure:
•**Type 1 Hypervisors,** also known as bare-metal or native.
•**Type 2 Hypervisors,** also known as hosted hypervisors.
The sections below explain both types in greater detail.



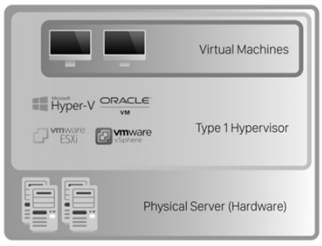
Type 1 Hypervisor



A Type 1 hypervisor is a layer of software installed directly on top of a physical server and its underlying hardware. Since no other software runs between the hardware and the hypervisor, it is also called the bare-metal hypervisor.

This hypervisor type provides excellent performance and stability since it does not run inside Windows or any other operating system. Instead, it is a simple operating system designed to run virtual machines. The physical machine the hypervisor runs on serves virtualization purposes only.

Type 1 hypervisors are mainly found in enterprise environments.



Type 1 Hypervisor – Pros

- **VM Mobility** - Type 1 hypervisors enable moving virtual machines between physical servers, manually or automatically. This move is based on the resource needs of a VM at a given moment and happens without any impact on the end-users. In case of a hardware failure, management software moves virtual machines to a working server as soon as an issue arises. The detection and restoration procedure takes place automatically and seamlessly.
- **Security** - The type 1 hypervisor has direct access to hardware without an additional OS layer. This direct connection significantly decreases the attack surface for potential malicious actors.
- **Resource Over-Allocation** - With type 1 hypervisors, you can assign more resources to your virtual machines than you have. For example, if you have 128GB of RAM on your server and eight virtual machines, you can assign 24GB of RAM to each. This totals 192GB of RAM, but VMs themselves will not consume all 24GB from the physical server. The VMs detect they have 24GB when they only use the amount of RAM they need to perform particular tasks.

Type 1 Hypervisor – Cons

Cons

•**Limited functionality** - Type 1 hypervisors are relatively simple and do not offer many features. The functionalities include basic operations such as changing the date and time, IP address, password, etc.

•**Complicated management** - To create virtual instances, you need a management console set up on another machine. Using the console, you can connect to the hypervisor on the server and manage your virtual environment.

•**Price** - Depending on what functionalities you need, the license cost for management consoles varies substantially.

Type 2 Hypervisor

Type 2 hypervisors run inside the physical host machine's operating system, which is why they are called **hosted hypervisors.** Unlike bare-metal hypervisors that run directly on the hardware, hosted hypervisors have one software layer in between. The system with a hosted hypervisor contains:

- A physical machine.
- An operating system installed on the hardware (Windows, Linux, macOS).
- A type 2 hypervisor software within that operating system.
- Guest virtual machine instances.

Type 2 hypervisors are typically found in environments with a small number of servers.
What makes them convenient is that they do not need a management console on another system to set up and manage virtual machines. Everything is performed on the server with the hypervisor installed, and virtual machines launch in a standard OS window.
Hosted hypervisors also act as management consoles for virtual machines. Any task can be performed using the built-in functionalities. Below is one example of a type 2 hypervisor interface (VirtualBox by Oracle):

Type 2 Hypervisor - PROs

Pros

- **Easy to manage** - There is no need to install separate software on another machine to create and maintain your virtual environment. Install and run a type 2 hypervisor as any other application within your OS. Create snapshots or clone your virtual machines, import or export appliances, etc.
- **Convenient for testing** - Type 2 hypervisors are convenient for testing new software and research projects. It is possible to use one physical machine to run multiple instances with different operating systems to test how an application behaves in each environment or to create a specific network environment. You only need to ensure that there are enough physical resources to keep the host and virtual machines running.
- **Allows access to additional productivity tools** - The users of type 2 hypervisors can use the tools available on other operating systems alongside their primary OS. For example, Windows users can access Linux applications by creating a Linux virtual machine.

Type 2 Hypervisor - CONs

innovate achieve lead

Cons

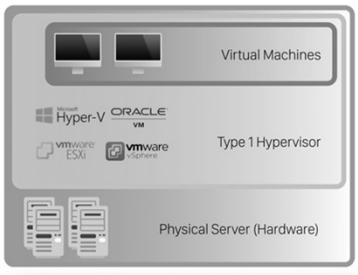
•**Less flexible resource management** - Allocating resources with this type of hypervisor is more difficult than with type 1. Bare-metal hypervisors can dynamically allocate available resources depending on the current needs of a particular VM. A type 2 hypervisor occupies whatever the user allocates to a virtual machine. When a user assigns 8GB of RAM to a VM, that amount will be taken up even if the VM is using only a fraction of it. If the host machine has 32GB of RAM and the user creates three VMs with 8GB each, they are left with 8GB of RAM to keep the physical machine running. Creating another VM with 8GB of ram would bring down the system.

•**Performance** - The host OS creates additional pressure on physical hardware, which may result in VMs having latency issues.

•**Security** - Type 2 hypervisors run on top of an operating system. This fact introduces a potential vulnerability since attackers may use potential vulnerabilities of the OS to gain access to virtual machines

Type 1 & 2 Hypervisor – At a Glance

innovate achieve lead



← Type 1 or Bare Metal is installed directly on hardware

Type 2 is installed on top of an existing OS →

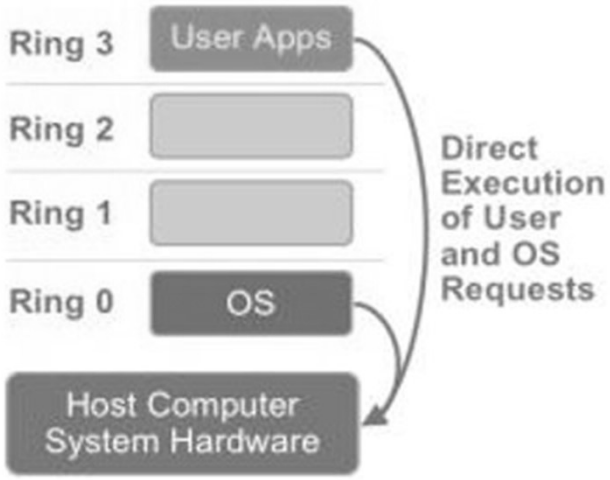
← Mostly used by Enterprise CSP

Mostly used by Medium & Medium Small scale enterprises →

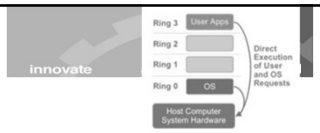
← Need to have a separate VM Management console to monitor the VM

Has an in-built console →

x86 Architecture



X86 Architecture



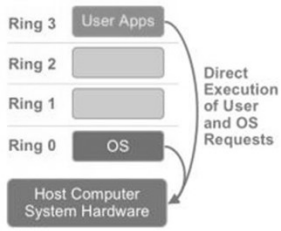
The x86 architecture uses a system of privilege rings to control access to sensitive areas of the computer's memory and resources. This system is known as Ring Architecture, and there are four privilege levels, or rings, numbered 0 through 3.

1. **Ring 0**: Also known as the kernel mode, ring 0 is the most privileged level, and it has full access to all of the computer's resources, including memory and I/O devices. This is where the operating system kernel runs.
2. **Ring 1**: This ring is used for device drivers and other low-level system components. Ring 1 has access to more resources than ring 2 and ring 3, but it still has limited access compared to ring 0.
3. **Ring 2**: Ring 2 is not used in modern x86 systems, but it was used in the past for system extensions, such as device drivers and system libraries. Ring 2 has even less access to the computer's resources than ring 1.
4. **Ring 3**: Also known as user mode, ring 3 is the least privileged level, and it is where user applications run. Ring 3 has limited access to the computer's resources, and any access to sensitive areas of the system must be performed through system calls that are handled by the operating system kernel running in ring 0.

Each process running on an x86 system runs in a **specific ring**, and the ring level **determines the level of access the process has to the computer's resources**. This system of privilege rings provides a layer of security by ensuring that user applications cannot interfere with the operation of the operating system and other system components.

Challenges to Virtualization

- X86 operating systems are designed to run directly on the bare-metal hardware, so they naturally assume they fully 'own' the computer hardware.
- As shown, the x86 architecture offers four levels of privilege known as Ring 0, 1, 2 and 3 to operating systems and applications to manage access to the computer hardware.
- While user level applications typically run in Ring 3, the operating system needs to have direct access to the memory and hardware and must execute its privileged instructions in Ring 0.
- Virtualizing the x86 architecture requires placing a virtualization layer under the operating system (which expects to be in the most privileged Ring 0) to create and manage the virtual machines that deliver shared resources



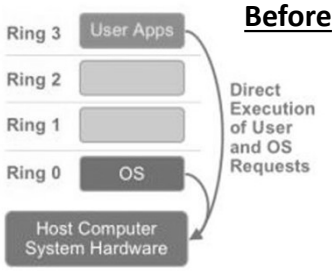
Further complicating the situation, some sensitive instructions can't effectively be virtualized as they have different semantics when they are not executed in Ring 0.

The difficulty in trapping and translating these sensitive and privileged instruction **requests at runtime was the challenge that originally made x86 architecture virtualization look impossible**

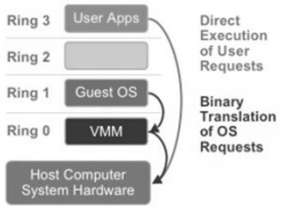
VMware's Solution

- VMware resolved the challenge in 1998, developing binary translation techniques that allow the VMM to run in Ring 0 for isolation and performance,
- The operating system was moved to a user level ring with greater privilege than applications in Ring 3 but less privilege than the virtual machine monitor in Ring 0.
- While VMware's full virtualization approach using binary translation is the de facto standard today the industry as a whole has not yet agreed on open standards to define and manage virtualization.
- Each company developing virtualization solutions is free to interpret the technical challenges and develop solutions with varying strengths and weaknesses.

Before



After



Virtualization Evolution

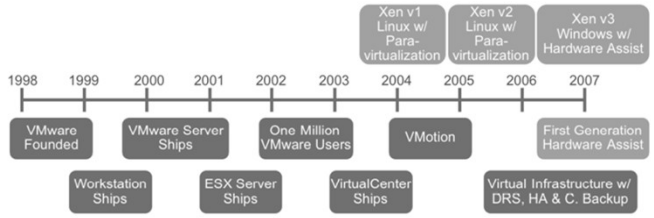


Figure 1 – Summary timeline of x86 virtualization technologies

Currently, the **most commonly used virtualization feature is hardware-assisted virtualization**. Hardware-assisted virtualization is a feature built into modern CPUs that provides improved performance and security for virtualization compared to traditional software-based virtualization technologies.
Hardware-assisted virtualization is supported by Intel's VT-x and AMD's AMD-V technology and is used by many popular virtualization platforms, such as VMware, Hyper-V, and Oracle VirtualBox. This technology allows the **virtualization software** to run **virtual machines** with **near-native performance**, as the **virtualization software** is able to directly access and use the CPU's hardware-assisted virtualization features

In 1998, VMware figured out how to virtualize the x86 platform, once thought to be impossible, and created the market for x86 virtualization.

The solution was a combination of binary translation and direct execution on the processor that allowed multiple guest OSes to run in full isolation on the same computer with readily affordable virtualization overhead

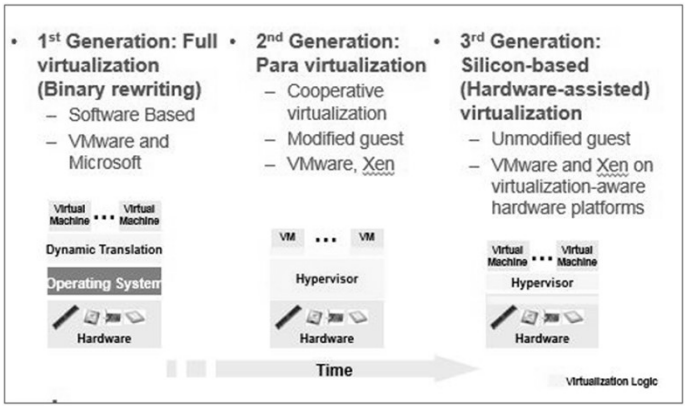
Introduction

Evolution

1st Generation: Full virtualization (Binary rewriting)
- Software Based
- VMware and Microsoft

2nd Generation: Para virtualization
- Cooperative virtualization
- Modified guest
- VMware, Xen

3rd Generation: Silicon-based (Hardware-assisted) virtualization
- Unmodified guest
- VMware and Xen on virtualization-aware hardware platforms



Emulation

Emulation is the process where the virtualizing software mimics that portion of hardware, which is provided to the guest operating system in the virtual machine. The presented emulated hardware is independent of the underlying physical hardware.

Emulation provides VM portability and wide range of hardware compatibility, which means the possibility of executing any virtual machine on any hardware, as the guest operating system interacts only with the emulated hardware.

In an emulated environment, both the application and guest operating system in virtual machines run in the user mode of base operating system. In simple terms, the behavior of the hardware is produced by a software program.

Emulation process involves only those hardware components so that user or virtual machines does not understand the underlying environment.

Emulation

Only CPU & memory are sufficient for basic level of emulation. Typically, emulation is implemented using interpretation. The emulator component takes each and every instruction of user mode and translates to equivalent instruction suitable according to the underlying hardware. This process is also termed as interpretation.

This means that the guest OS remains completely unaware of the virtualization. Also, in interpretation, each and every instruction issued by a VM is trapped in the VMM and interpreted for execution in the hardware. Goes without saying that computationally it is a very expensive method. However, in some cases,, it is needed to use an interpretation technique. However, due to the huge disadvantage of performance, emulation using interpretation is hardly used in virtualization.

Binary Translation / Full Virtualization

innovate achieve lead

In its basic form known as “full virtualization” the hypervisor provides a fully emulated machine in which an operating system can run. VMWare is a good example.

The biggest advantage to this approach is its flexibility: one could run a RISC-based OS as a guest on an Intel-based host.

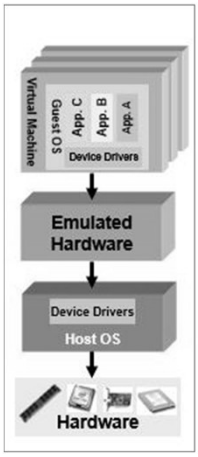
While this is an obvious approach, there are significant performance problems in trying to emulate a complete set of hardware in software.

1st Generation offering of x86/x64 server virtualization .

Dynamic binary translation.

The emulation layer talks to an operating system which talks to the computer hardware.

The guest OS doesn't see that it is used in an emulated environment.



Binary Translation / Full Virtualization - Pros

innovate achieve lead

The emulation layer Isolates VMs from the host OS and from each other.

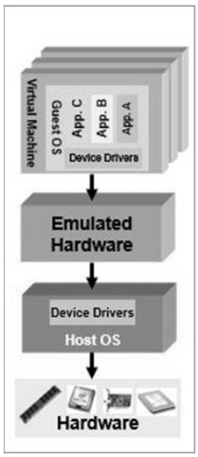
Controls individual VM access to system resources, preventing an unstable VM from impacting system performance.

Total VM portability.

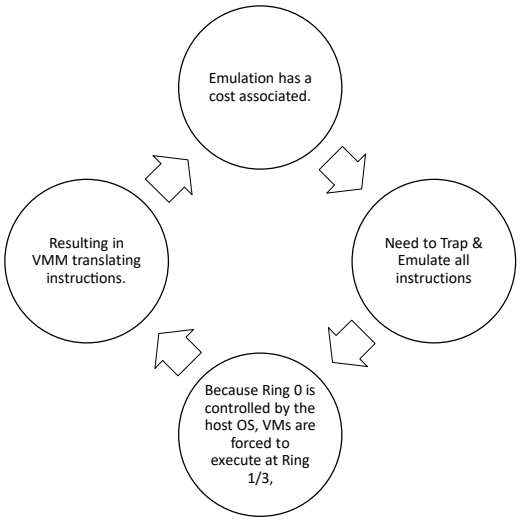
By emulating a consistent set of system hardware, VMs have the ability to transparently move between hosts with dissimilar hardware without any problems.

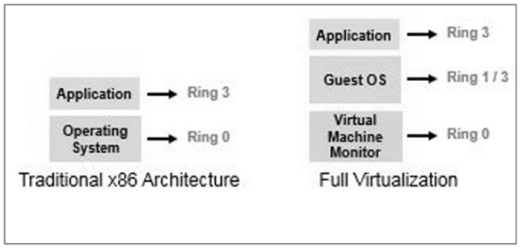
It is possible to run an operating system that was developed for another architecture on your own architecture.

A VM running on a Dell server can be relocated to a Hewlett-Packard server.



Binary Translation / Full Virtualization - Cons





Para Virtualization

Paravirtualization," found in the XenSource, open source Xen product, attempts to reconcile these two approaches. Instead of emulating hardware, paravirtualization uses slightly altered versions of the operating system which allows access to the hardware resources directly as managed by the hypervisor

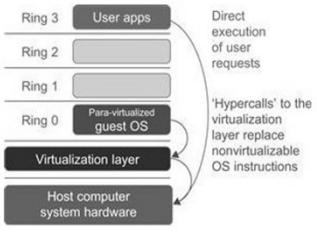
The Guest OS is modified and thus run kernel level operations at Ring 1 (or 3)

the guest is fully aware of how to process privileged instructions

thus, privileged instruction translation by the VMM is no longer necessary

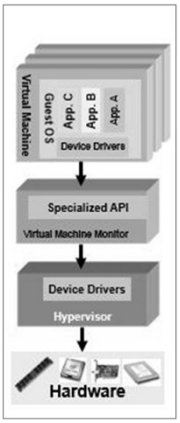
The guest operating system uses a specialized API to talk to the VMM and, in this way, execute the privileged instructions

The VMM is responsible for handling the virtualization requests and putting them to the hardware



Para Virtualization

- Today, VM guest operating systems are para virtualized using two different approaches:
- **Recompiling the OS kernel**
  - Para virtualization drivers and APIs must reside in the guest operating system kernel.
  - Modified operating system that includes this specific API, requiring a compiling operating systems to be virtualization aware.
  - Some vendors (such as Novell) have embraced para virtualization and have provided para virtualized OS builds, while other vendors (such as Microsoft) have not.
- **Installing para virtualized drivers**
  - In some operating systems it is not possible to use complete para virtualization, as it requires a specialized version of the operating system
  - To ensure good performance in such environments, para virtualization can be applied for individual devices
  - For example, the instructions generated by network boards or graphical interface cards can be modified before they leave the virtualized machine by using para virtualized drivers



H/W Assisted Virtualization

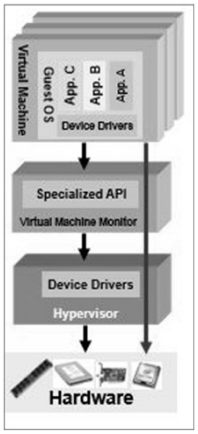
This technique attempts to simplify virtualization because full or paravirtualization is complicated.

Intel and AMD add an additional mode called privilege mode level (some people call it Ring-1) to x86 processors.

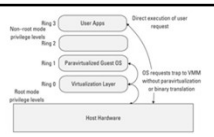
Therefore, operating systems can still run at Ring 0 and the hypervisor can run at Ring -1.

All the privileged and sensitive instructions are trapped in the hypervisor automatically.

This technique removes the difficulty of implementing binary translation of full virtualization. It also lets the operating run without modification unlike para virtualization



H/W Assisted Virtualization



Hardware-assisted virtualization. This term refers to a scenario in which the hardware provides architectural support for building a virtual machine manager able to run a guest operating system in complete isolation.

This technique was originally introduced in the IBM System/370. At present, examples of hardware-assisted virtualization are the extensions to the x86-64 bit architecture introduced with *Intel VT* (formerly known as *Vanderpool*) and *AMD V* (formerly known as *Pacifica*).

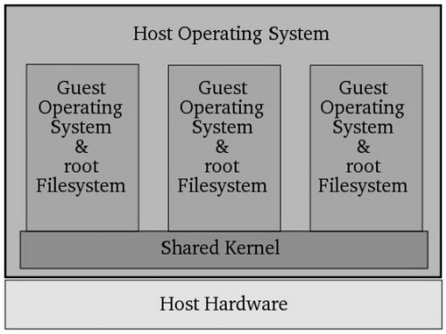
These extensions, which differ between the two vendors, are meant to reduce the performance penalties experienced by emulating x86 hardware with hypervisors. Before the introduction of hardware-assisted virtualization, software emulation of x86 hardware was significantly costly from the performance point of view.

The reason for this is that by design the x86 architecture did not meet the formal requirements introduced by Popek and Goldberg, and early products were using binary translation to trap some sensitive instructions and provide an emulated version. Products such as VMware Virtual Platform, introduced in 1999 by VMware, which pioneered the field of x86 virtualization, were based on this technique. After 2006, Intel and AMD introduced processor extensions, and a wide range of virtualization solutions took advantage of them: Kernel-based Virtual Machine (KVM), VirtualBox, Xen, VMware, Hyper-V, Sun xVM, Parallels, and others.

SKI Virtualization

Instead of using a hypervisor, it runs a separate version of the Linux kernel and sees the associated virtual machine as a user-space process on the physical host. This makes it easy to run multiple virtual machines on a single host. A device driver is used for communication between the main Linux kernel and the virtual machine.
Processor support is required for virtualization (Intel VT or AMD – v). A slightly modified QEMU process is used as the display and execution containers for the virtual machines. In many ways, kernel-level virtualization is a specialized form of server virtualization.

Examples: User – Mode Linux(UML) and Kernel Virtual Machine(KVM) , Docker, LXC



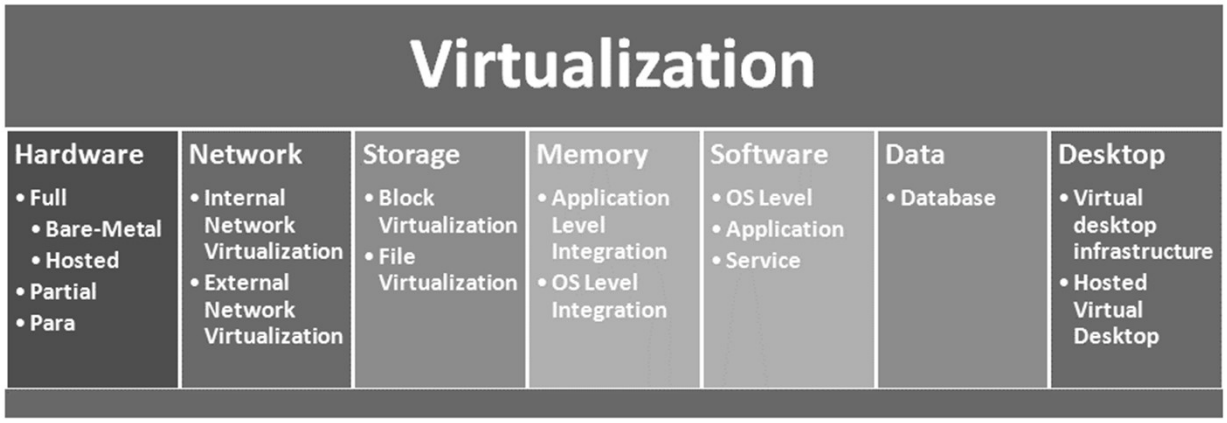
Virtualization Comparison

innovate achieve lead

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic

Virtualization

innovate achieve lead



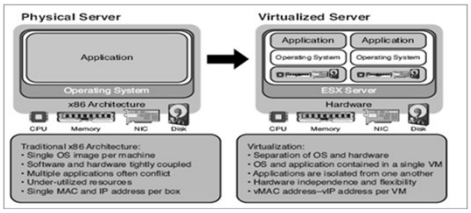
Server Virtualization

Abstracts the physical machine on which the software and operating system is running on and provides an illusion that the software is running on a virtual machine.

Enables Infrastructure as a service model.

A single physical machine can be used to create several VMs that can run several operating systems independently and simultaneously. VMs are stored as files, so restoring a failed system can be as simple as copying its file onto a new machine.

The hypervisor software enables the creation of a virtual machine (VM) that emulates a physical computer by creating a separate OS environment that is logically isolated from the host server.



Server Virtualization - Beneits

Partitioning

Run multiple operating systems on one physical machine.

Divide the physical system resources among virtual machines.

One VM does not know the presence of the other.

Management

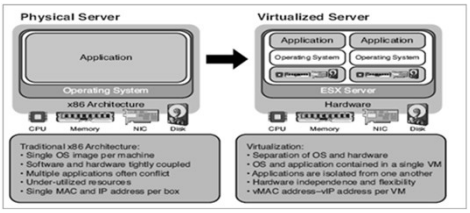
Failure of one VM does not affect other VMs.

Management agents can be run on each VM separately to determine the individual performance of the VM and the applications that are running on the VM.

Encapsulation

The entire VM state can be saved in a file.

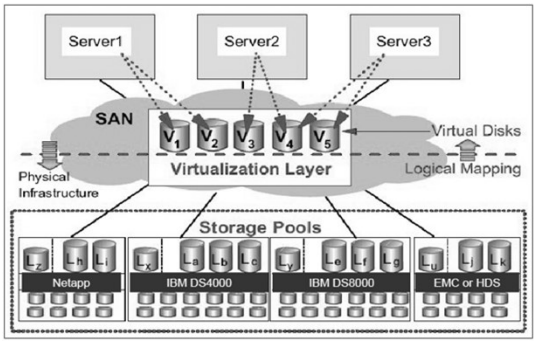
Moving and copying VM information is as easy as copying files.



Server virtualization is a key driving force in reducing the number of physical servers and hence the physical space, cooling, cabling, and capital expenses in any data center consolidation projects

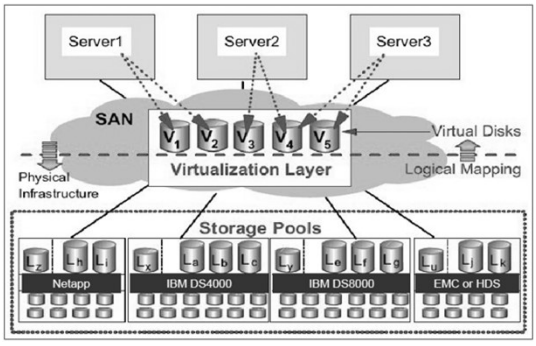
Storage Virtualization

- Storage virtualization refers to providing a logical, abstracted view of physical storage devices.
- It provides a way for many users or applications to access storage without being concerned with where or how that storage is physically located or managed.
- It enables physical storage in an environment to be shared across multiple application servers, and physical devices behind the virtualization layer to be viewed and managed as if they were one large storage pool with no physical boundaries.
- The storage virtualization hides the fact there are separate storage devices in an organization by making all the devices appear as one device.
- Virtualization hides the complex process of where the data needs to be stored and bringing it back and presenting it to the user when it is required.



Storage Virtualization - Benefits

- Typically, the benefits are :-
- **Resource optimization** : Storage virtualization enables you to obtain the storage space on an as-needed basis without any wastage, and it allows organizations to use existing storage assets more efficiently without the need to purchase additional assets.
- **Cost of operation**: Storage virtualization enables adding storage resources without regard to the application, and storage resources can be easily added to the pool by a drag-and-drop method using a management console by the operations people.
- **Increased availability:** Storage virtualization provisions the new storage resources in a minimal amount of time, improving the overall availability of resources
- **Improved performance**

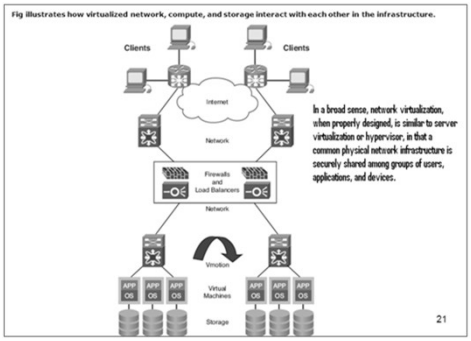


Examples of Storage Virtualization:

- SAN ➔ Storage area Network
- VDA ➔ Virtual Disk Array

Network Virtualization

- Network virtualization might be the most ambiguous virtualization of all virtualization types. Several types of network virtualization exist, as briefly described here:
- A VLAN is a simple example of network virtualization. VLANs allow logical segmentation of a LAN into several broadcast domains. VLANs are defined on a switch on a port-by-port basis. That is, you might choose to make ports 1–10 part of VLAN 1 and ports 11–20 part of VLAN 2. There's no need for ports in the same VLAN to be contiguous. Because this is a logical segmentation and not physical, workstations connected to the ports do not have to be located together, and users on different floors in a building or different buildings can be connected together to form a LAN.
- Virtual Routing and Forwarding (VRF), commonly used in Multi-Protocol Label Switching (MPLS) networks, **allows multiple instances of a routing table to coexist within the same router at the same time.**



Memory Virtualization

Beyond CPU virtualization, the next critical component is memory virtualization.

This involves sharing the physical system memory and dynamically allocating it to virtual machines.

Virtual machine memory virtualization is very similar to the virtual memory support provided by modern operating systems.

Applications see a contiguous address space that is not necessarily tied to the underlying physical memory in the system.

The operating system keeps mappings of virtual page numbers to physical page numbers stored in page tables. All modern x86 CPUs include a memory management unit (MMU) and a translation lookaside buffer (TLB) to optimize virtual memory performance

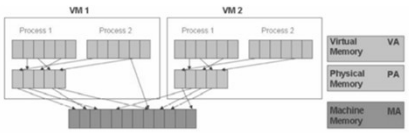


Figure 8 – Memory Virtualization

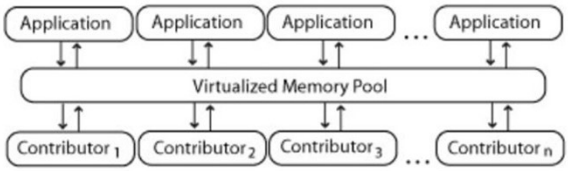
Memory Virtualization

innovate achieve lead

It introduces a way to decouple memory from the server to provide a shared, distributed or networked function. It enhances performance by providing greater memory capacity without any addition to the mai n memory. That's why a portion of the disk drive serves as an extension of the main memory.

Application level integration – Applications running on connected computers directly connect to the memory pool through an API or the file system.

Operating System Level Integration – The operating system first connects to the memory pool, and makes that pooled memory available to applications.



Device Virtualization

innovate achieve lead

It provides the work convenience and security.

As one can access remotely, you are able to work from any location and on any PC. It provides a lot of flexibility for employees to work from home or on the go.

It also protects confidential data from being lost or stolen by keeping it safe on central servers.

This involves managing the routing of I/O requests between virtual devices and the shared physical hardware.

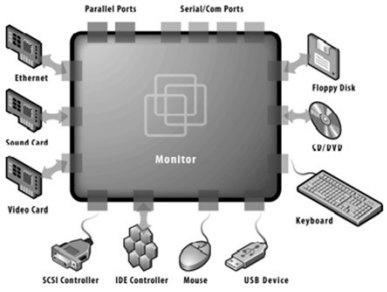


Figure 9 – Device and I/O virtualization

BITS Pilani, Pilani Campus

Virtualization Advantages

- Instant provisioning - fast scalability
- Live Migration is possible
- Load balancing and consolidation in a Data Center is possible.
- Low downtime for maintenance
- Virtual hardware supports legacy operating systems efficiently
- Security and fault isolation

Virtualization Summary

	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen I lypervisor VMI-Linux is Hypervisor agnostic

Virtualization Advantages

Security: by compartmentalizing environments with different security requirements in different virtual machines one can select the guest operating system and tools that are more appropriate for each environment. For example, we may want to run the Apache web server on top of a Linux guest operating system and a backend MS SQL server on top of a guest Windows XP operating system, all in the same physical platform. A security attack on one virtual machine does not compromise the others because of their isolation.

Virtualization Advantages

Reliability and availability: A software failure in a virtual machine does not affect other virtual machines.

Cost: It is possible to achieve cost reductions by consolidation smaller servers into more powerful servers. Cost reductions stem from hardware cost reductions (economies of scale seen in faster servers), operations cost reductions in terms of personnel, floor space, and software licenses. VMware cites overall cost reductions ranging from 29 to 64%

Virtualization Advantages

Adaptability to Workload Variations: Changes in workload intensity levels can be easily taken care of by shifting resources and priority allocations among virtual machines. Autonomic computing-based resource allocation techniques, such as the ones in can be used to dynamically move processors from one virtual machine to another.

Load Balancing: Since the software state of an entire virtual machine is completely encapsulated by the VMM, it is relatively easy to migrate virtual machines to other platforms in order to improve performance through better load balancing

Virtualization Advantages

Legacy Applications: Even if an organization decides to migrate to a different operating system, it is possible to continue to run legacy applications on the old OS running as a guest OS within a VM. This reduces the migration cost.

Points to Note

- **Software licensing**

  One of the most significant virtualization-related issues to be aware of is software licensing. Virtualization makes it easy to create new servers, but each VM requires its own separate software license. Organizations using expensive licensed applications could end up paying large amounts in license fees if they do not control their server sprawl.
- **IT training**

  IT staff used to dealing with physical systems will need a certain amount of training in virtualization. Such training is essential to enable the staff to debug and troubleshoot issues in the virtual environment, to secure and manage VMs, and to effectively plan for capacity.
- **Hardware investment**

  Server virtualization is most effective when powerful physical machines are used to host several VMs. This means that organizations that have existing not-so-powerful hardware might still need to make upfront investments in acquiring new physical servers to harvest the benefits of virtualization.

Application of Virtualization

•Today, virtualization can apply to a range of system layers, including hardware-level virtualization, operating system-level virtualization, and high-level language virtual machines.

•**Maximize resources** — Virtualization can reduce the number of physical systems you need to acquire, and you can get more value out of the servers. Most traditionally built systems are underutilized. Virtualization allows maximum use of the hardware investment.

•

•**Multiple systems** — With virtualization, you can also run multiple types of applications and even run different OS for those applications on the same physical hardware.

•**IT budget integration** — When you use virtualization, management, administration and all the attendant requirements of managing your own infrastructure remain a direct cost of your IT operation.

Technology Trends

- Virtualization is Key to Exploiting Trends
- Allows most efficient use of the compute resources
  - Few apps take advantage of 16+ CPUs and huge memory as well as virtualization
  - Virtualization layer worries about NUMA, not apps
- Maximize performance per watt across all servers
  - Run VMs on minimal # of servers, shutting off the others
  - Automated, live migration critical:
    - Provide performance guarantees for dynamic workloads
    - Balance load to minimize number of active servers
- Stateless, Run-anywhere Capabilities
  - Shared network and storage allows flexible mappings
  - Enables additional availability guarantees

Agenda

- Virtualization Recap
- Infrastructure as a Service
  - What is IaaS
  - Introduce AWS
  - AWS Reference Model
  - AWS Compute
  - AWS Storage
  - AWS Network
  - AWS Case Study - Abof

What is Virtualization?

innovate achieve lead

Virtualization Defined

Virtualization is a computer architecture technology by which multiple virtual machines (VMs) are multiplexed in the same hardware machine.

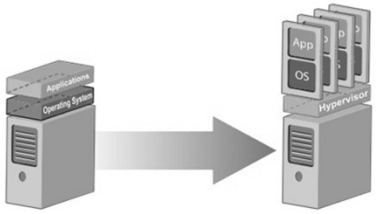
Virtualization allows multiple operating system instances to run concurrently on a single computer

Instead of purchasing and maintaining an entire computer for one application, each application can be given its own operating system, and all those operating systems can reside on a single piece of hardware.

Virtualization allows an operator to control a guest operating system's use of CPU, memory, storage, and other resources, so each guest receives only the resources that it needs.

Key Terms:

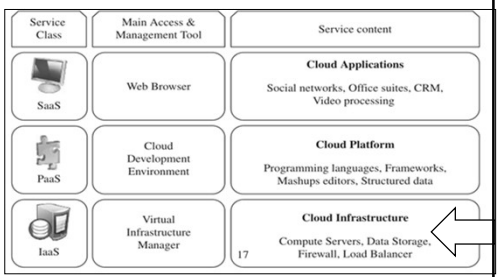
- VM ➔ Virtual Machine
- VMM ➔ Virtual Machine Monitor
- Hypervisor ➔ VMM
- Multiplexed ➔ Many or several
- Host ➔ System where the VMM resides
- Guest ➔ Virtual Machines created



What is IaaS?

innovate achieve lead

- **The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources**.
- 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 limited control of select networking components (e.g., host firewalls).
- Offering virtualized resources (computation, storage, and communication) on demand is known as Infrastructure as a Service (IaaS).
- Infrastructure services are considered to be the bottom layer of cloud computing systems .
- Ex : Amazon EC2 : Elastic Compute Cloud, Eucalyptus, GoGrid, Rackspace Cloud



Infrastructure as a Service

Why IaaS

Capability

Iaas provides the following
- Servers- compute, machines
- Storage
- Network
- Operating system

Characteristics

Resources are distributed as a service
- Allows dynamic scaling (1...10....100.....)
- Has a variable costs-
- Generally includes multiple-users on a single piece of hardware. (**multi-tenancy)**

Enabler : Virtualization Technology
- ✓ Manageability and Interoperability
- ✓ Availability and Reliability
- ✓ Scalability and Elasticity

Models

IaaS can be obtained as
(1) Public or
(2) Private infrastructure or
(3) combination of both

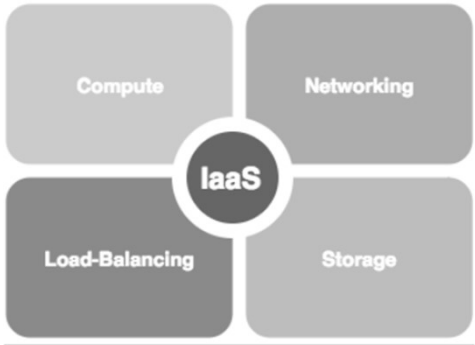
Benefit

The user instead of **purchasing-** servers, software, **data center space** or network equipment, **rent** those resources as a fully **outsourced** service on-demand model.

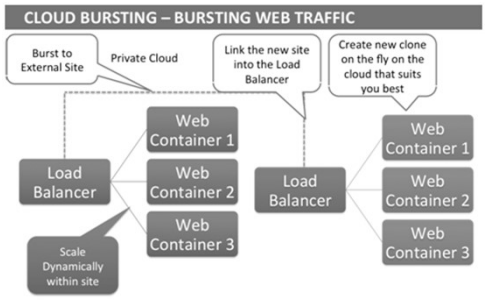
IaaS - Motivations

- No upfront cost
- Multi-tier auto provisioning within hours
- Scalability, services/resources are provided on demand
- Flexible/Secure direct connectivity
- SLA with 99.9% uptime

CORE



IaaS – Key Terms



•**Cloudbursting**: The process of off-loading tasks to the cloud during times when the most compute resources are needed.

•**Resource pooling**: **Pooling** is a resource management term that refers to the grouping together of resources (compute(cpu), network(bandwidth), storage) for the purposes of ***maximizing advantage*** and/or ***minimizing risk*** to the users.

•**Multi-tenant computing**: Multi-tenancy is an architecture in which a single instance of a software application serves multiple customers. Each customer is called a tenant. Tenants may be given the ability to customize some parts of the application, such as color of the user interface (UI) or business rules, but they cannot customize the application's code.

•**Hypervisor**: Software which enables virtualization.

Pros & Cons of IaaS

IaaS helps

1. Where demand is very **volatile**- encountering **spikes and troughs.**
2. For new enterprise without **capital to invest in hardware or** entrepreneurs starting on a shoestring budget.
3. Where the enterprise is growing rapidly and scaling hardware would be problematic.
4. For specific line of business, trial or temporary infrastructural needs
5. When you need computing power on the go, turn to IaaS.

IaaS Negates

- Where regulatory **compliance** makes the offshoring or outsourcing of data storage and processing difficult
- Where the **highest levels of performance** are required, and on premise or dedicated hosted infrastructure has the capacity to meet the organization's needs

Introducing Amazon Web Service

What is AWS

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

Global Infrastructure: AWS serves over one million active customers in more than 190 countries, and it continues to expand its global infrastructure

Security: All AWS customers benefit from data center and network architectures built to satisfy the requirements of the most security-sensitive organizations.

- Application building blocks
- Stable APIs
- Proven Amazon infrastructure
- Focus on innovation and creativity
- Long-term investment

AWS Understanding Service Offering

- AWS operates state-of-the-art, highly available data centers. Although rare, failures can occur that affect the availability of instances that are in the same location.
- If you host all of your instances in a single location that is affected by a failure, none of your instances would be available.
- Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of AWS Regions, Availability Zones, Local Zones, AWS Outposts, and Wavelength Zones.

Availability Zones located in AWS Regions consist of one or more discrete data centers, each of which has redundant power, networking, and connectivity, and is housed in separate facilities. Each AZ has multiple internet connections and power connections to multiple grids.

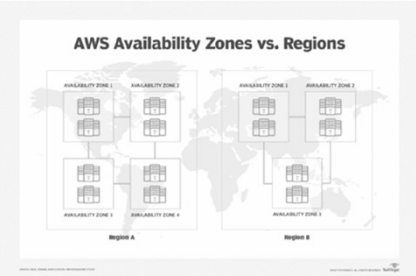
AWS Regions

- AWS provides a **highly available technology infrastructure platform** with multiple locations worldwide. These locations are composed of **regions** and **Availability Zones**. Each **region** is a **separate geographic area**.
- It is important to remember that each AWS Region is completely independent. Any Amazon service you initiate (for example, creating database instances or listing available database instances) runs only in your current default AWS Region.
- The default AWS Region can be changed in the console, or by setting the AWS_DEFAULT_REGION environment variable. Or it can be overridden by using the --region parameter with the AWS Command Line Interface (AWS CLI).
- Each AWS Region is designed to be isolated from the other AWS Regions. This design achieves the greatest possible fault

Availability Zones located in AWS Regions consist of one or more discrete data centers, each of which has redundant power, networking, and connectivity, and is housed in separate facilities. Each AZ has multiple internet connections and power connections to multiple grids.

AWS Availability Zones

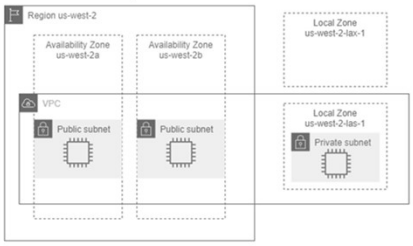
- Each Region has multiple, isolated locations known as **Availability Zones**. Each **Availability Zone** is also **isolated,** but the **Availability Zones** in a region are connected **through low-latency links**.
- **Availability Zones** are **physically separated** within a typical metropolitan region and are located in lower-risk flood plains (specific flood zone categorization varies by region). In addition to using a **discrete uninterruptable power supply** (UPS) and on-site backup generators, they are each fed via **different grids** from **independent utilities** (when available) to reduce single points of failure further.
- **Availability Zones** are all **redundantly connected** to multiple tier-1 transit providers. By placing resources in **separate Availability Zones**, you can protect your website or application from a **service disruption** impacting a single location.
- The code for Availability Zone is its Region code followed by a letter identifier. For example, us-east-1a.



- If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in **another Availability Zone can handle requests.**

AWS Local Zones

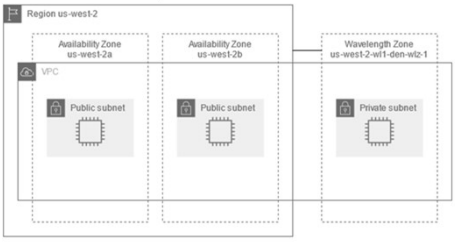
A **Local Zone** is an extension of an **AWS Region** in geographic proximity to your users.
Local Zones have their own connections to the internet and support AWS Direct Connect, so that resources created in a Local Zone can serve local users with low-latency communications.
The code for a Local Zone is its Region code followed by an identifier that indicates its physical location. For example, us-west-2-lax-1 in Los Angeles.
The **VPC** spans the **Availability Zones** and one of the Local Zones. Each **zone** in the **VPC** has one **subnet**, and each **subnet has an instance**.
When you launch an instance, you can specify a subnet that is in a Local Zone. You also allocate an IP address from a network border group. A network border group is a unique set of Availability Zones, Local Zones, or Wavelength Zones from which AWS advertises IP addresses, for example, us-west-2-lax-1a.



- Some **AWS resources** might not be available in all **Regions**. Make sure that you can create the resources that you need in the desired Regions or Local Zones before launching an instance in a specific Local Zone
- Before you can **specify a Local Zone** for a **resource** or service, you must **opt in** to Local Zones.

AWS Wavelength Zones

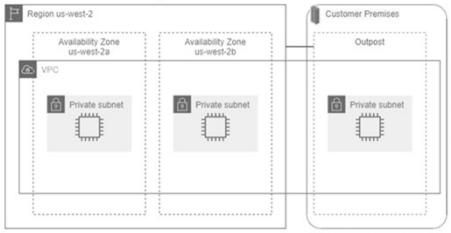
- **Wavelength Zones** are **AWS infrastructure deployments** that embed AWS compute and storage services within **telecommunications providers' data centers** at the edge **of the 5G network**, so application traffic can reach application servers running in Wavelength Zones without leaving the mobile providers' network.
- This **prevents the latency** that would result from multiple hops to the internet and enables customers to take **full advantage of 5G networks**. Wavelength Zones extend **AWS to the 5G edge**, delivering a consistent developer experience across **multiple 5G networks around the world**. Wavelength Zones also allow developers to **build** the next generation of **ultra-low latency applications** using the same **familiar AWS services**, APIs, tools, and functionality they already use today.
- Processing at the network edge can help avoid transmitting large volumes of data over the network provider's infrastructure, and offload processing from mobile device hardware.



This enables new classes of compute-intensive, latency sensitive applications latency. For example, a fleet of autonomous cars interacting with road sensors to prevent crashes, smart industrial robots assessing and reacting to plant conditions in a dangerous manufacturing environment, or retailers serving personalized promotions to shoppers' mobile phones in real time as they pass product displays.

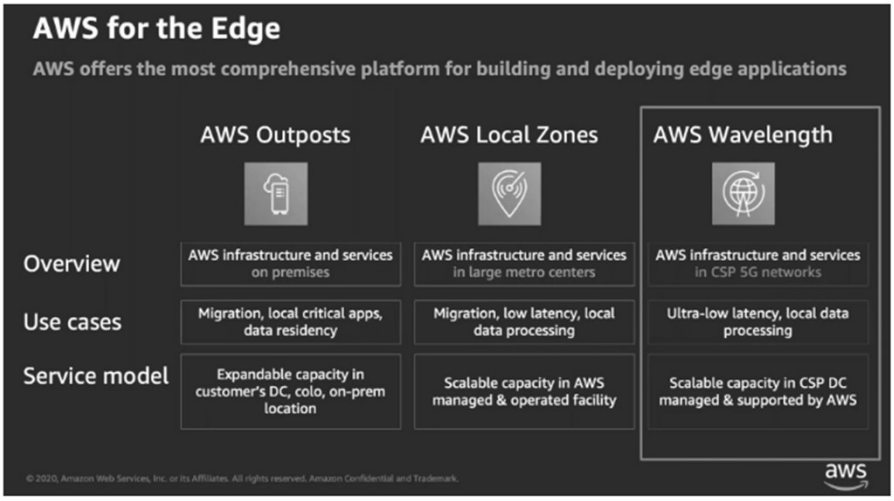
AWS Outposts

- **AWS Outposts** is a fully **managed service** that extends **AWS infrastructure**, services, APIs, and tools to **customer premises**. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.
- **AWS operates, monitors, and manages this capacity** as part of an **AWS Region**. You can create subnets on your Outpost and specify them when you create AWS resources. Instances in Outpost subnets communicate with other instances in the AWS Region using private IP addresses, all within the same VPC.
- The following diagram illustrates the AWS Region us-west-2, two of its Availability Zones, and an Outpost. The VPC spans the Availability Zones and the Outpost. The Outpost is in an on-premises customer data center. Each zone in the VPC has one subnet, and each subnet has an instance.



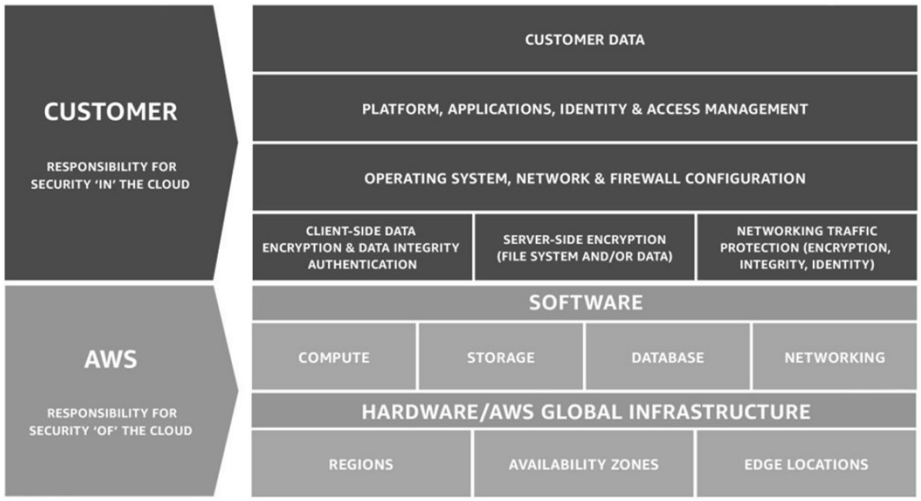
- This enables new classes of compute-intensive, latency sensitive applications latency. For example, a fleet of autonomous cars interacting with road sensors to prevent crashes, smart industrial robots assessing and reacting to plant conditions in a dangerous manufacturing environment, or retailers serving personalized promotions to shoppers' mobile phones in real time as they pass product displays.

AWS for the Edge



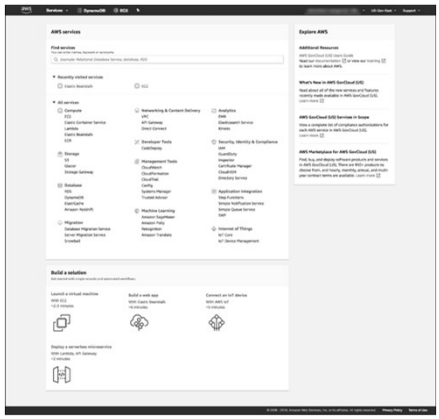
AWS edge computing infrastructure and service platforms. Source: AWS

AWS & Customer



Using AWS - Connecting

- **AWS Management Console**: is a web application for managing AWS Cloud services. The console provides an intuitive user interface for performing many tasks. Each service has its own console, which can be accessed from the AWS Management Console. The console also provides information about the account and billing.
- **AWS Command Line Interface (CLI)** is a unified tool used to manage AWS Cloud services. With just one tool to download and configure, you can control multiple services from the command line and automate them through scripts.
- **The AWS Software Development Kits (SDKs)** provide an application programming interface (API) that interacts with the web services that fundamentally make up the AWS platform. The SDKs provide support for many different programming languages and platforms to allow you to work with your preferred language.



Using AWS - Video

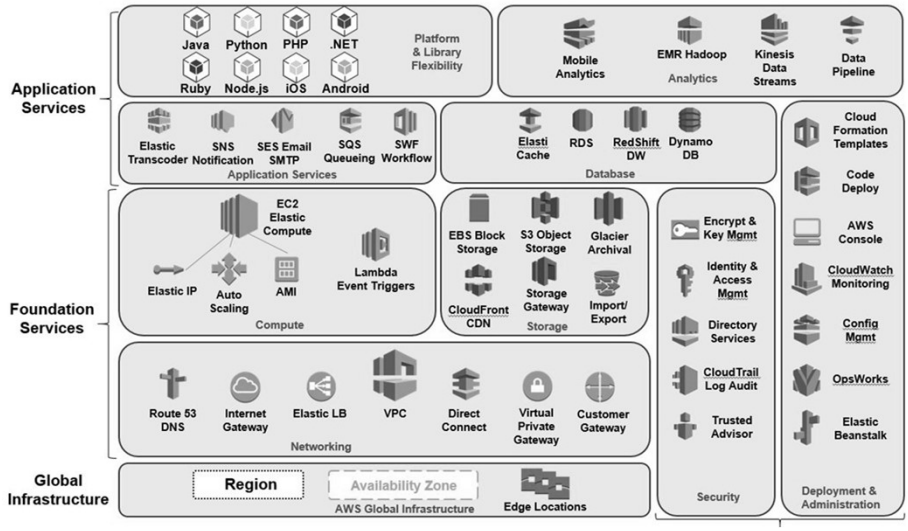
AWS Management Console: is a web application for managing AWS Cloud services.

The console provides an intuitive user interface for performing many tasks.

Each service has its own console, which can be accessed from the AWS Management Console.

The console also provides information about the account and billing.

AWS Reference Model



AWS Elastic Compute Cloud EC2

Amazon EC2

- Virtual Compute Cloud
- Root-level System Access
- Elastic Capacity
- Management API
- Scale in Minutes
- Multiple Instance Sizes
- Network Security Model

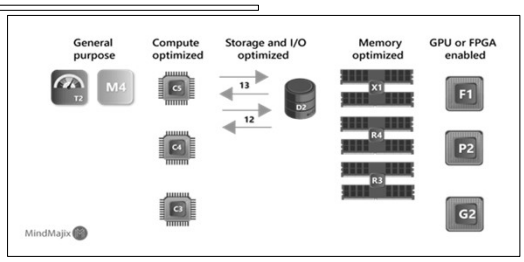
EC2 Introduction

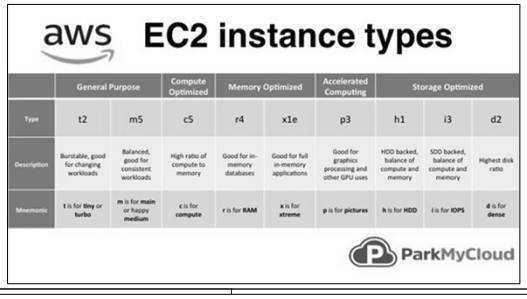
- Amazon EC2 is **AWS primary web service** that provides **resizable compute capacity** in the **cloud**.
- **Compute** refers to the amount of **computational power required to fulfill your workload**.
- Amazon EC2 allows you to acquire compute through the **launching** of **virtual servers** called **instances**.
- When you launch an **instance**, you can make use of the compute as you wish, just as you would with an on-premises server.
- **Users pay** for the **computing power** of the instance. Charged per hour while the instance is running. When you **stop the instance**, you are **no longer charged**.

- Virtual Compute Cloud
- Root-level System Access
- Elastic Capacity
- Management API
- Scale in Minutes
- Multiple Instance Sizes
- Network Security Model

EC2 - Instance Type & AMI

- There are two concepts that are key to launching instances on AWS:
- (1) **Instance Type** :The amount of virtual hardware dedicated to the instance and
- (2) **AMI**: The software loaded on the instance.
  - AMI ➔ Amazon Machine Image
- Note
  - ***Instance Type is similar to the processor***
  - ***AMI is similar to the OS***

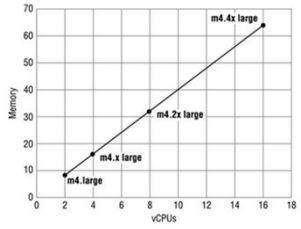




EC2 - Instance Type

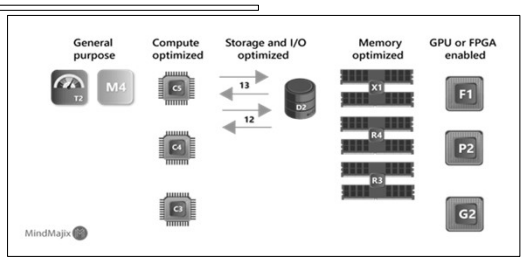
- The **instance type** defines the **virtual hardware** supporting an Amazon EC2 instance.
- There are dozens of instance types available, varying in the following dimensions:
  - Virtual CPUs (vCPUs)
  - Memory
  - Storage (size and type)
- Network performance **Instance types** are **grouped into families** based on the **ratio of these** values to **each other**.
- For instance, the **m4 family** provides a balance of **compute, memory, and network** resources, and it is a good choice for many applications.
- Within each family there are several choices that scale up linearly in size.
- Note that the ratio of vCPUs to memory is constant as the sizes scale linearly.

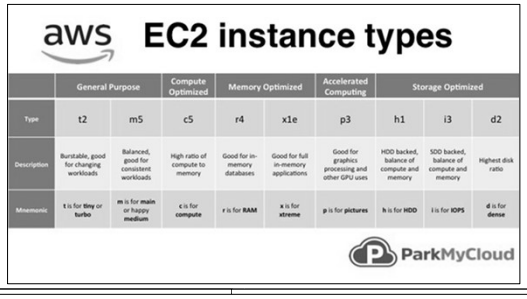
Instance Family	Instance Type(s)
General Purpose (M3)	M3.medium, M3.large, M3.xlarge, M3.2xlarge
Compute Optimized (C3)	C3.large, C3.xlarge, C3.2xlarge, C3.4xlarge, C3.8xlarge
Memory Optimized (R3)	R3.large, R3.xlarge, R3.2xlarge, R3.4xlarge, R3.8xlarge
Storage Optimized (I2, HS1)	I2.xlarge, I2.2xlarge, I2.4xlarge, I2.8xlarge, HS1.8xlarge
GPU (G2)	G2.2xlarge
Micro (T1, M1)	T1.micro, M1.small



EC2 - Instance Type

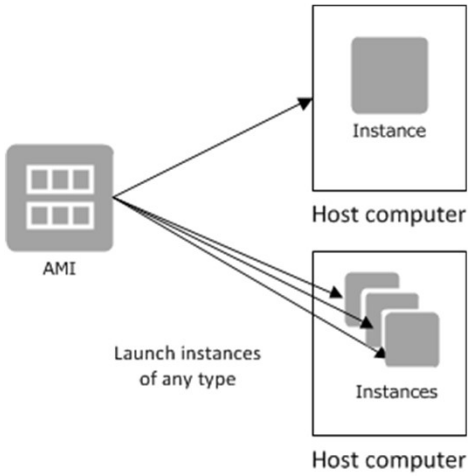
- Another variable to consider when choosing an instance type is **network performance**.
- For most instance types, AWS publishes a relative measure of **network performance**: *low*, *moderate*, or *high*.
- Some instance types specify a network performance of **10 Gbps**.
- The network performance increases within a family as the instance type grows.
- For workloads which require low latency, AWS provides enhanced networking support.





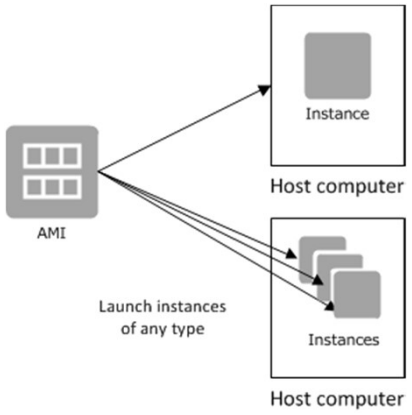
EC2 - Amazon Machine Image (AMI)

- The **Amazon Machine Image** (AMI) defines the **initial software** that will be on an instance when it is launched.
- An **AMI defines** every aspect of the **software state at instance launch**, including:
  - The Operating System (OS) and its configuration
  - The initial state of any patches
  - Application or system software
- All AMIs are based on x86 OSs, either Linux or Windows.



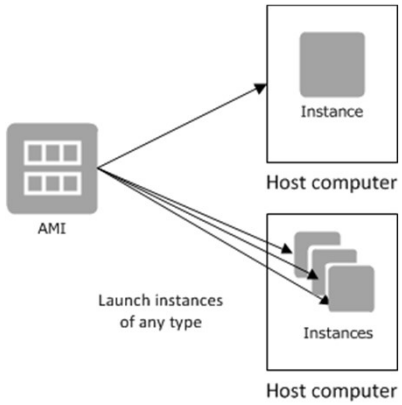
EC2 - AMI Types

- **Published by AWS**— AWS publishes AMIs with versions of many different OSs, both Linux and Windows.
- These include multiple distributions of Linux (including Ubuntu, Red Hat, and Amazon's own distribution) and Windows 2008 and Windows 2012.
- Launching an instance based on one of these AMIs will result in the default OS settings, similar to installing an OS from the standard OS ISO image. As with any OS installation, you should immediately apply all appropriate patches upon launch.



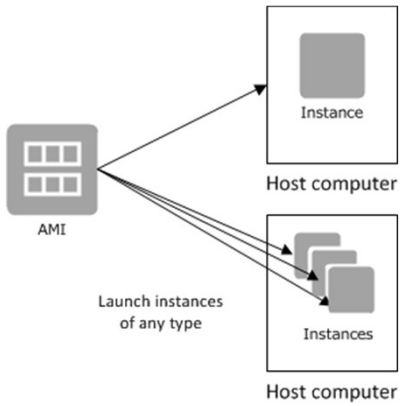
EC2 - AMI Types

- **The AWS Marketplace**— AWS Marketplace is an **online store** that helps **customers find, buy, and immediately start using the software and services that run on Amazon EC2**.
- Many AWS partners have made their software available in the AWS Marketplace.
- This provides two benefits:
  - The customer **does not need to install** the software, and
  - The **license agreement is appropriate** for the cloud.
- Instances launched from an AWS Marketplace AMI incur the standard hourly cost of the instance type plus an additional per-hour charge for the additional software (some open-source AWS Marketplace packages have no additional software charge).



EC2 - AMI Types

- **Generated from Existing Instances**— An AMI can be created from an **existing Amazon EC2 instance**.
- This is a very **common source** of AMIs. Customers **launch an instance** from a published AMI, and then the **instance is configured to meet all the customer's corporate standards** for updates, management, security, and so on.
- **An AMI is then generated** from the configured instance and used to **generate all instances** of that OS.
- In this way, all new instances **follow the corporate standard** and it is more difficult for individual projects to launch non-conforming instances.



EC2 - AMI Types

- **Uploaded Virtual Servers—** Using AWS VM Import/ Export service, customers can create images from various virtualization formats, including raw, VHD, VMDK, and OVA. The current list of supported OSs (Linux and Windows) can be found in the AWS documentation. It is incumbent on the customers to remain compliant.

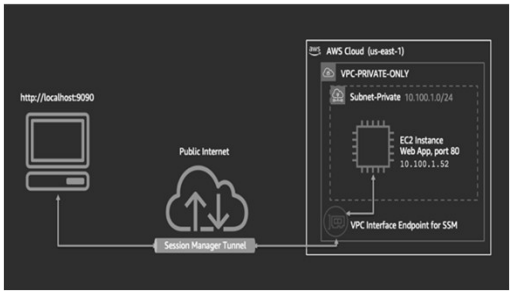
- **VMDK➔** Virtual Machine Disk： is a file format that describes containers for virtual hard disk drives to be used in virtual machines like VMware Workstation or VirtualBox.
- **VHD ➔** Virtual Hard Disk: is a file format which represents a **virtual hard disk** drive (HDD). It may contain what is found on a physical HDD, such as disk partitions and a file system, which in turn can contain files and folders. It is typically used as the hard disk of a virtual machine.
- **OVAF➔** Open Virtual Appliance/Application Format: is merely a single **file** distribution of the same **file** package, stored in the TAR format

Creating an EC2 Instance - Video

- Amazon **EC2 uses public-key cryptography** to encrypt and decrypt login information.
- **Public-key cryptography** uses a **public key to encrypt** a piece of data and an **associated private key** to decrypt the data.
- These two keys together are called a **key pair**.
- **Key pairs** can be created through the **AWS Management Console**, CLI, or API, or customers can upload their own key pairs.
- **AWS stores the public key**, and the **private key** is kept by the **customer**.
- The **private key** is essential to acquiring **secure access** to an **instance** for the **first time**.

EC2 – Accessing over Web

- There are several ways that an instance may be addressed over the web upon creation:
- **Public Domain Name System (DNS) Name**— When you launch an instance, AWS creates a DNS name that can be used to access the instance. This DNS name is generated automatically and cannot be specified by the customer.
- **Public IP**— A launched instance may also have a public IP address assigned. This IP address is assigned from the addresses reserved by AWS and cannot be specified. This IP address is unique on the Internet, persists only while the instance is running, and cannot be transferred to another instance.
- **Elastic IP**— An elastic IP address is an static address unique on the Internet that you reserve independently and associate with an Amazon EC2 instance.

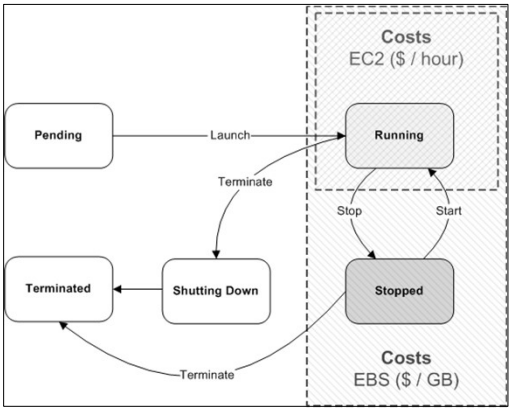


EC2 – Lifecycle

- Amazon EC2 has several features and services that facilitate the management of Amazon EC2 instances over their entire lifecycle.
- Launching
- Bootstrapping: The process of providing code to be run on an instance at launch is called bootstrapping.

Managing Instances

- When the number of instances in your account starts to climb, it can become difficult to keep track of them.
- Tags can help you manage not just your Amazon EC2 instances, but also many of your AWS Cloud services.
- Tags are key/ value pairs you can associate with your instance or other service.
- Tags can be used to identify attributes of an instance like project, environment (dev, test, and so on), billable department, and so forth.
- You can apply up to 10 tags per instance.



Monitoring Instances

AWS offers a service called Amazon CloudWatch that provides monitoring and alerting for Amazon EC2 instances, and other AWS infrastructure.

EC2 – Tenancy Options

- There are several tenancy options for Amazon EC2 instances that can help customers achieve security and compliance goals.
- **Shared Tenancy** Shared tenancy is the default tenancy model for all Amazon EC2 instances, regardless of instance type, pricing model, and so forth. **Shared tenancy means that a single host machine may house instances from different customers.** As AWS does not use overprovisioning and fully isolates instances from other instances on the same host, this is a secure tenancy model.
- **Dedicated Instances** Dedicated Instances run on hardware that's dedicated to a single customer. As a customer runs more Dedicated Instances, more underlying hardware may be dedicated to their account. Other instances in the account (those not designated as dedicated) will run on shared tenancy and will be isolated at the hardware level from the Dedicated
- **Dedicated Host** An Amazon EC2 Dedicated Host is a physical server with Amazon EC2 instance capacity fully dedicated to a single customer's use. Dedicated Hosts can help you address licensing requirements and reduce costs by allowing you to use your existing server-bound software licenses.

EC2 – Placement Groups

- Placement Groups A placement group is a **logical grouping** of instances within a **single Availability Zone**. Placement groups enable **applications to participate** in a low-latency, **10 Gbps network**. To fully use this network performance for your placement group, choose an **instance type** that supports **enhanced networking** and 10 Gbps network performance.
- **Instance Stores** An instance store (sometimes referred to as **ephemeral storage**) provides **temporary block-level storage** for your instance. This storage is located on disks that are physically attached to the host computer.
- The size and type of **instance stores** available with an Amazon EC2 instance depend on the **instance type**. Can range from **no instance store** to **24 2 TB** instance store
- **Instance stores** are included in the cost of an **Amazon EC2 instance,** so they are a very cost-effective solution for appropriate workloads. The key aspect of instance stores is that they are temporary.
- Data in the instance store is lost when:
  - The underlying disk drive fails.
  - The instance stops (the data will persist if an instance reboots).
  - The instance terminates.

AWS VPC

innovate achieve lead

WHAT

- A ***virtual private cloud*** (VPC) is a virtual network **dedicated to your AWS account**. It is **logically isolated** from other virtual networks in the AWS cloud.
- You can **launch your AWS resources,** such as Amazon EC2 instances, into **your VPC**. You can provision your own **logically isolated section of AWS,** similar to designing and implementing a separate **independent network** that would operate in an on-premises data center.
- You can configure your VPC; you can select its **IP address range, create subnets,** and **configure route tables,** network **gateways,** and **security settings**. A ***subnet*** is a **range of IP addresses** in your VPC. You can **launch AWS resources into a subnet that you select**.
- Use a **public subnet** for **resources that must be connected to the Internet,** and a **private subnet** for resources that **won't be connected to the Internet**.Within a **region,** you can create **multiple Amazon VPCs,** and each **Amazon VPC** is **logically isolated** even if it shares its IP address space..

uses

- Build virtual networks on the cloud
- No need for any VPN, hardware or physical DC
- Define bespoke network space like:
  - VPC with a single public subnet only
  - VPC with public and private subnets
  - VPC with public and private subnets and AWS Site-to-Site VPN access
  - VPC with a private subnet only and AWS Site-to-Site VPN access

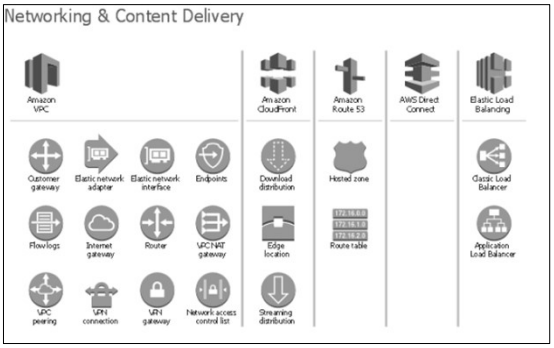
AWS VPC - Components

innovate achieve lead

WHAT

- An Amazon VPC consists of the following components:
  - Subnets
  - Route tables
  - Dynamic Host Configuration Protocol (DHCP) option sets
  - Security groups
  - Network Access Control Lists (ACLs)
- An Amazon VPC has the following optional components:
  - Internet Gateways (IGWs)
  - Elastic IP (EIP) addresses
  - Elastic Network Interfaces (ENIs)
  - Endpoints
  - Peering
  - Network Address Translation (NATs) instances and
  - NAT gateways Virtual Private Gateway (VPG), Customer Gateways (CGWs), and Virtual Private Networks (VPNs)

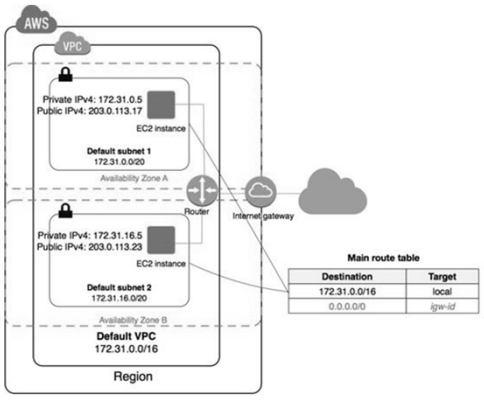
SERVICES



BITS Pilani, Pilani Campus

AWS VPC - Functioning

- **You control** how the **instances** that you launch into a **VPC access resources outside the VPC**.
- Your **default VPC includes** an **Internet gateway,** and each default **subnet is a public subnet**.
- Each instance that you launch into a **default subnet** has a **private IPv4 address and a public IPv4 address**.
- These instances can communicate with the Internet through the Internet gateway.
- **By default, each instance** that you launch into a **non-default subnet has a private IPv4 address, but no public IPv4 address**, unless you specifically assign one at launch, or you modify the subnet's public IP address attribute.
- These instances can communicate with each other, but can't access the Internet.

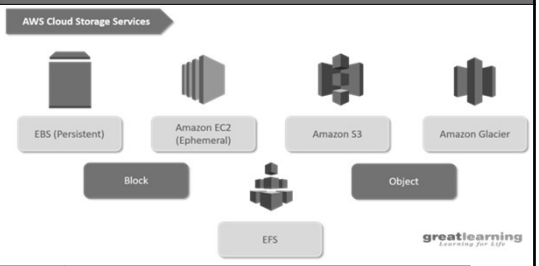


Creating VPC- Video

Amazon VPC comprises a variety of objects that will be familiar to customers with existing networks:

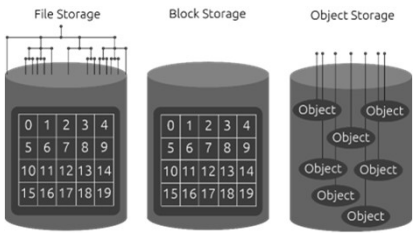
- **A Virtual Private Cloud:** A logically isolated virtual network in the AWS cloud. You define a VPC's IP address space from ranges you select.
- **Subnet:** A segment of a VPC's IP address range where you can place groups of isolated resources.
- **Internet Gateway:** The Amazon VPC side of a connection to the public Internet.
- **NAT Gateway:** A highly available, managed Network Address Translation (NAT) service for your resources in a private subnet to access the Internet.
- **Virtual private gateway:** The Amazon VPC side of a VPN connection.
- **Peering Connection:** A peering connection enables you to route traffic via private IP addresses between two peered VPCs.
- **VPC Endpoints:** Enables private connectivity to services hosted in AWS, from within your VPC without using an Internet Gateway, VPN, Network Address Translation (NAT) devices, or firewall proxies.
- **Egress-only Internet Gateway:** A stateful gateway to provide egress only access for IPv6 traffic from the VPC to the Internet.

AWS Storage



Storage Types

- **Block storage** : Operates at a lower level— the raw storage device level— and manages data as a set of numbered, fixed-size blocks.
- **File storage :** Operates at a higher level— the operating system level— and manages data as a named hierarchy of files and folders.
- *Block and file storage are often accessed over a network in the form of a Storage Area Network (SAN) for block storage, using protocols such as iSCSI or Fibre Channel, or as a Network Attached Storage (NAS) file server or "filer" for file storage.*



	Amazon Simple Storage Service (Amazon S3)	A service that provides scalable and highly durable object storage in the cloud.
	Amazon Glacier	A service that provides low-cost highly durable archive storage in the cloud.
	Amazon Elastic File System (Amazon EFS)	A service that provides scalable network file storage for Amazon EC2 instances.
	Amazon Elastic Block Store (Amazon EBS)	A service that provides block storage volumes for Amazon EC2 instances.
	Amazon EC2 Instance Storage	Temporary block storage volumes for Amazon EC2 instances.
	AWS Storage Gateway	An on-premises storage appliance that integrates with cloud storage.
	AWS Snowball	A service that transports large amounts of data to and from the cloud.
	Amazon CloudFront	A service that provides a global content delivery network (CDN).

AWS S3

- Amazon S3 is easy-to-use object storage with a simple web service interface that you can use to store and retrieve any amount of data from anywhere on the web.
- Amazon S3 also allows you to pay only for the storage you actually use, which eliminates the capacity planning and capacity constraints associated with traditional storage.
- Amazon S3 can be used alone or in conjunction with other AWS services, and it offers a very high level of integration with many other AWS cloud services.

uses

- Backup and archive for on-premises or cloud data
- Content, media, and software storage and distribution
- Big data analytics
- Static website hosting
- Cloud-native mobile and Internet application hosting
- Disaster recovery

SIEMENS Ingenuity for life

Nasdaq

GE Healthcare

RYANAIR

nielsen

AWS S3

- ✓ Amazon S3 is **cloud object storage**. Instead of being **closely associated with a server**, Amazon S3 storage is **independent of a server** and is **accessed** over the **Internet**.
- ✓ Instead of managing data as blocks or files using SCSI, CIFS, or NFS protocols, data is managed as objects using an Application Program Interface **(API) built on standard HTTP verbs**. Each Amazon **S3 object** contains both **data and metadata**.
- ✓ **Objects reside** in containers called **buckets**, and each object is **identified by a unique user-specified key (filename).**
- ✓ Buckets are a simple flat folder with no file system hierarchy.
- ✓ That is, you can **have multiple buckets**, but you **can't have a sub-bucket** within a bucket. Each **bucket** can hold an **unlimited number of objects**.

- However, keep in mind that **Amazon S3** is not a **traditional file system** and differs in significant ways.
- In **Amazon S3**, you **GET** an object or **PUT** an **object**, operating on the **whole object** at once, instead of **incrementally updating** portions of the object as you would with a file.
- Instead of a **file system**, **Amazon S3** is **highly-durable** and **highly-scalable object storage** that is optimized for reads and is built with an intentionally minimalistic feature set.
- It provides a **simple and robust abstraction** for file storage that frees you from many underlying details that you normally do have to deal with in traditional storage.
- Amazon S3 objects are **automatically replicated** on **multiple devices** in **multiple facilities** within a **region**.

AWS S3

Objects:

Opaque data to be stored (1 byte … 5 Gigabytes)

Authentication and access controls

Buckets:

Object container – any number of objects

100 buckets per account

Keys:

Unique object identifier within bucket

Up to 1024 bytes long

Flat object storage model

Standards-Based Interfaces:

REST and SOAP

URL-Addressability – every object has a URL

Service:

ListAllMyBuckets

Buckets:

CreateBucket DeleteBucket

ListBucket GetBucketAccessControlPolicy

SetBucketAccessControlPolicy

GetBucketLoggingStatus SetBucketLoggingStatus

Objects:

PutObject PutObjectInline

GetObject GetObjectExtended

DeleteObject GetObjectAccessControlPolicy

SetObjectAccessControlPolicy

AWS EBS

- Amazon EBS provides persistent block-level storage volumes for use with Amazon EC2 instances.
- Each Amazon EBS volume is automatically replicated within its Availability Zone to protect you from component failure, offering high availability and durability.
- Amazon EBS volumes are available in a variety of types that differ in performance characteristics and price.
- Multiple Amazon EBS volumes can be attached to a single Amazon EC2 instance, although a volume can only be attached to a single instance at a time.

uses

- Boot Volumes
- SQL & NoSQL Database
- Big Data workloads
- Data Warehouses
- Logging & Telemetry
- Transaction Processing

Kellogg's

Viasat

EQUIFAX

Bristol Myers Squibb

AWS EBS Types

General-Purpose SSD General-purpose SSD volumes offer cost-effective storage that is ideal for a broad range of workloads. They deliver strong performance at a moderate price point that is suitable for a wide range of workloads.

A general-purpose SSD volume can range in size from 1 GB to 16 TB and provides a baseline performance of three IOPS per gigabyte provisioned, capping at 10,000 IOPS.

They are suited for a wide range of workloads where the very highest disk performance is not critical, such as:

- System boot volumes
- Small- to medium-sized databases
- Development and test environments

AWS EBS Types

Provisioned IOPS SSD Provisioned IOPS SSD volumes are designed to meet the needs of I/ O-intensive workloads, particularly database workloads that are sensitive to storage performance and consistency in random access I/ O throughput. While they are the most expensive Amazon EBS volume type per gigabyte, they provide the highest performance of any Amazon EBS volume type in a predictable manner.

A Provisioned IOPS SSD volume can range in size from 4 GB to 16 TB. Provisioned IOPS SSD volumes provide predictable, high performance and are well suited for:

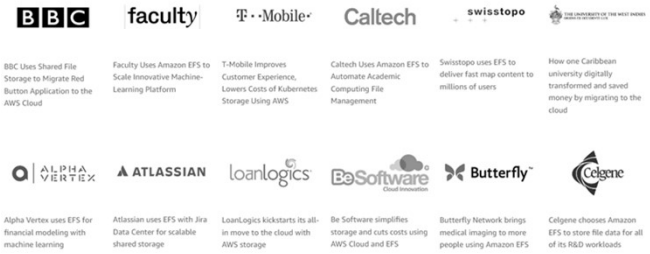
- Critical business applications that require sustained IOPS performance
- Large database workloads

AWS EFS

- EFS(Elastic file system) is a file-level storage service that basically provides a shared elastic file system with virtually unlimited scalability support
- EFS is highly available storage that can be utilized by many servers at the same time. AWS EFS is a fully managed service by amazon and it offers scalability on the fly.
- This means that the user need not worry about their increasing or decreasing workload. If the workload suddenly becomes higher then the storage will automatically scale itself and if the workload decreases then the storage will itself scale down.
- This scalability feature of EFS also provides cost benefits as you need not pay anything for the part of storage that you don't use, you only pay for what you use(Utility-based computing).

uses

- Lift-and-shift application support
- Analytics for big data
- Web server support
- Application development and testing



AWS EFS vs AWS EBS vs AWS S3

Category	S3	EBS	EFS
Storage Type	Object Storage	Block Storage	File Storage
Pricing	Pay as you Use	Pay for provisioned capacity	Pay as you Use
Storage Size	Unlimited Storage	Limited storage	Unlimited Storage
Scalability	Unlimited Scalability	Increase/decrease size manaually	Unlimited Scalability
Durability	Stored redundantly across multiple Azs	Stored redundantly in a Single AZ	Stored redundantly across multiple Azs
Availability	Max is 99.99% with S3 Standard	99.99%	No SLAs
Security	Supports Data at Rest and Data in Transit encryption	Supports Data at Rest and Data in Transit encryption	Supports Data at Rest and Data in Transit encryption
Back up and Restore	Use Versioning or cross-region replication	Automated Backups and Snapshots	EFS to EFS replication
Performance	Slower than EBS and EFS	Faster than S3 and EFS	Faster than S3, Slower than EBS
Accessibility	Publicly and Privately accessible	Accessible only via the attached EC2 instance	Accessible simulatenously from multiple EC2 and on-premises instance
Interface	Web Interface	File System Interface	Web and File System Interface
Use cases	Media, Entertainment, Big data analytics, backups and archives, web serving and content management	Boot volumes, transactional and NoSQL databases, data warehousing ETL	Media, Entertainment, Big data analytics, backups and archives, web serving and content management, home directories

Amazon Glacier

- **Amazon Glacier** is an **extremely low-cost storage** service that provides durable, secure, and flexible **storage for data archiving and online backup**.
- To keep costs low, Amazon Glacier is designed for infrequently accessed data where a **retrieval time** of **three to five hours** is acceptable.
- Amazon Glacier can **store** an **unlimited amount of virtually any kind of data**, in any format.
- In most cases, the data stored in Amazon Glacier consists of large **TAR** (Tape Archive) or ZIP files.
- In **Amazon Glacier**, data is stored in **archives**. An **archive** can contain up to **40TB** of data, and you can have an **unlimited number of archives**.
- Each **archive** is assigned a unique archive ID at the time of creation.
- (Unlike an Amazon S3 object key, you cannot specify a user-friendly archive name.) All archives are automatically **encrypted, and archives are immutable**— after an archive is created, it cannot be modified.

uses

- Digital Storage.
- Scientific Data Storage.
- Healthcare information Archiving.
- Regulatory and Compliance Archiving.
- Magnetic Tape Replacement.

ROCK & ROLL HALL OF FAME
Rock & Roll Hall of Fame preserves rock music history and modernizes on AWS.
Read the case study »

QUBE
Qube Cinema cuts costs by 80% with archival on Amazon S3 Glacier.
Read the case study »

REUTERS
Reuters builds easily accessible large-scale news archives on Amazon S3 Glacier.
Read the blog »

BandLab
BandLab decreases costs and improves availability using Amazon S3 Glacier.
Read the case study »

joyn
Joyn readies exclusive content for audiences with Amazon S3 Intelligent-Tiering and Amazon S3 Glacier.
Read the blog »

innovate achieve lead

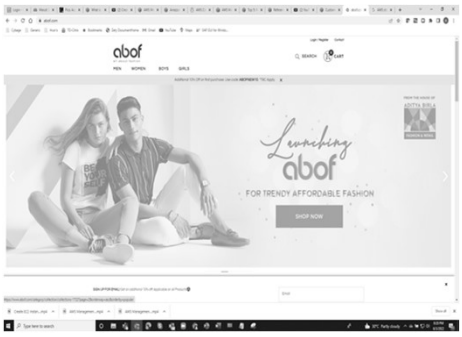
BITS Pilani

Pilani|Dubai|Goa|Hyderabad

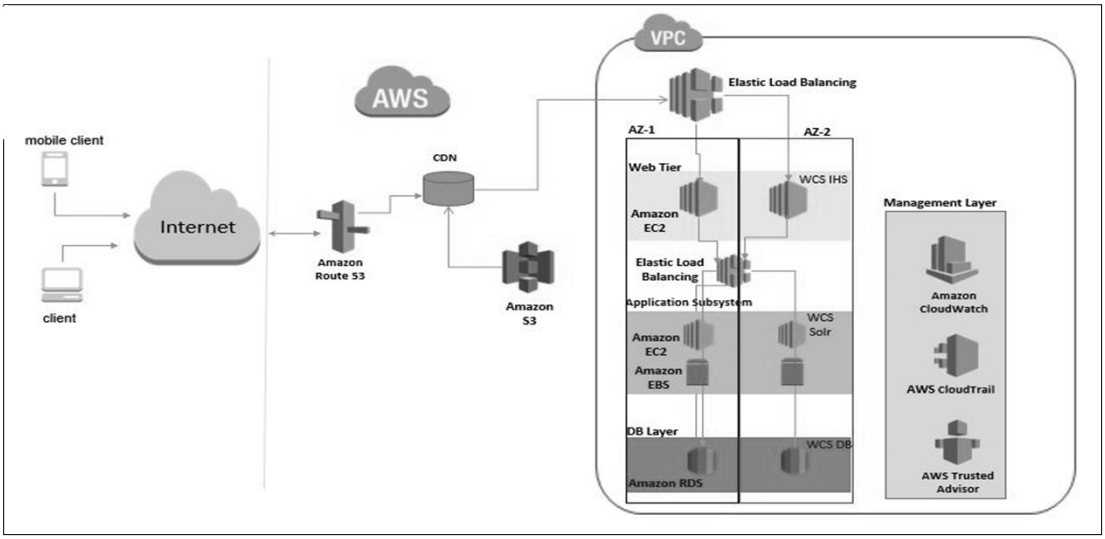
AWS Case Study

ABOF

- Running its infrastructure in AWS has enabled abof to achieve 99.999 percent availability and an average page loading time of 1.5 seconds.
- Agility needed to thrive in the competitive online fashion industry in India.
- The business is using
  - Amazon EC2 to run IBM WebSphere Commerce Suite and an IBM DB2 database,
  - Amazon S3 to store website images and video,
  - Amazon RDS to run a MySQL database supporting an in-house developed logistics system, and Amazon
  - CloudFront to deliver content to users across India and internationally.



ABOF



Agenda

- AWS Recap
- OpenStack Introduction
  - What is OpenStack
  - OS Reference Model
  - Introducing Network Function Virtualization (NFV)
  - OS Case Study

What is AWS

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Millions of customers—including the fastest-growing startups, largest enterprises, and leading government agencies—are using AWS to lower costs, become more agile, and innovate faster.

Global Infrastructure: AWS serves over one million active customers in more than 190 countries, and it continues to expand its global infrastructure

Security: All AWS customers benefit from data center and network architectures built to satisfy the requirements of the most security-sensitive organizations.

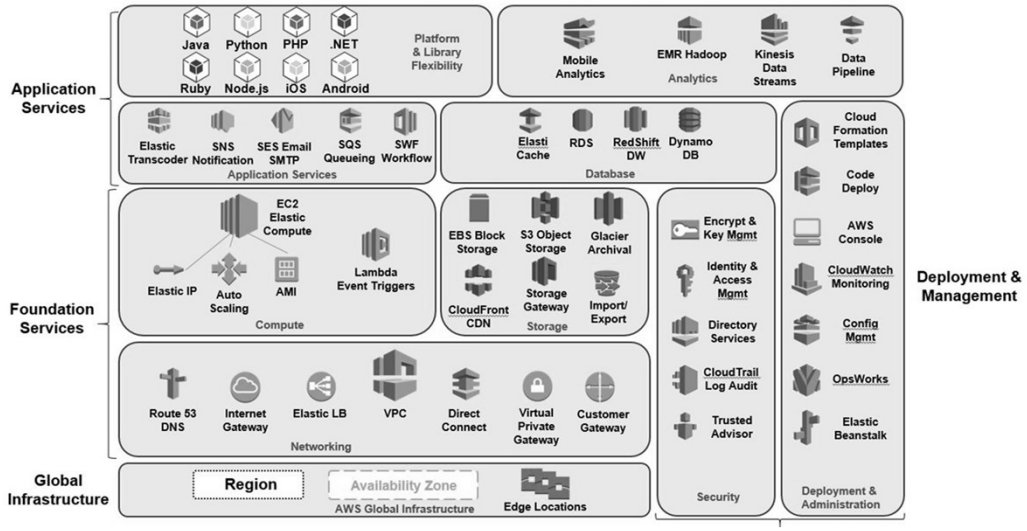
- Application building blocks
- Stable APIs
- Proven Amazon infrastructure
- Focus on innovation and creativity
- Long-term investment

AWS Regions & Availability Zones

- AWS provides a **highly available technology infrastructure platform** with multiple locations worldwide. These locations are composed of **regions** and **Availability Zones**. Each **region** is a **separate geographic area**.
- Each **region** has **multiple, isolated locations** known as **Availability Zones**. AWS enables the **placement of resources** and data in multiple **locations**. Resources **aren't replicated** across regions **unless organizations** choose to do so.
- Additionally, for faster delivery of content, AWS has **EDGE locations** concentrated in major cities. (Used with CloudFront & Route53)

Availability Zones located in AWS Regions consist of one or more discrete data centers, each of which has redundant power, networking, and connectivity, and is housed in separate facilities. Each AZ has multiple internet connections and power connections to multiple grids.

AWS Reference Model



What is OpenStack?

innovate achieve lead

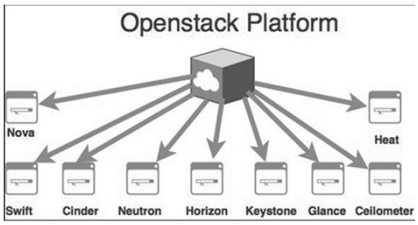
OpenStack is a **cloud operating system** that controls **large pools of compute, storage, and networking resources**.

Managed through **a dashboard** that gives administrators control while empowering their users to provision resources through a web interface.

OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds.

Backed by some of the biggest companies in software development and hosting, as well as thousands of individual community members, many think that OpenStack is the future of cloud computing.

Managed by **OpenStack Foundation a non profit organization**.

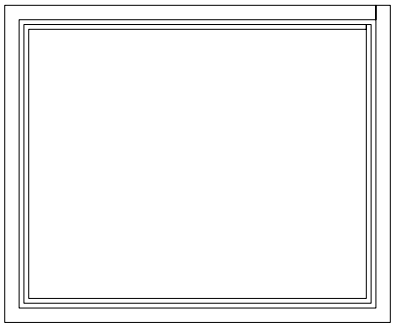


History of Open stack

innovate achieve lead

OpenStack was created during the first months of 2010. Rackspace wanted to rewrite the infrastructure code running its Cloud servers offering, and considered open sourcing the existing Cloud files code. At the same time, Anso Labs (contracting for NASA) had published beta code for Nova, a Python-based “cloud computing fabric controller”.

Both efforts converged and formed the base for OpenStack. The first Design Summit was held in Austin, TX on July 13-14, 2010, and the project was officially announced at OSCON in Portland, OR, on July 21st, 2010.



Mission Statement

The Four Opens

Open Source

We do **not** produce "open core" software.

We are committed to creating truly open source software that is usable and scalable. Truly open source software is not feature or performance limited. There will be no "Enterprise Edition".

We use the Apache License, 2.0.

- OSI approved
- GPLv3 compatible
- DFSG compatible

Open Design

We are committed to an open design process. Every development cycle the OpenStack community holds face-to-face events to gather requirements and write specifications for the upcoming release. Those events, which are **open to anyone**, include users, developers, and upstream projects. We gather requirements, define priorities and flesh out technical design to guide development for the next development cycle.

The community controls the design process. You can help make this software meet your needs.

Open Development

We maintain a publicly available source code repository through the entire development process. We do public code reviews. We have public roadmaps. This makes participation simpler, allows users to follow the development process and participate in QA at an early stage.

Open Community

One of our core goals is to maintain a healthy, vibrant developer and user community. Most decisions are made using a lazy consensus model. All processes are documented, open and transparent.

The technical governance of the project is provided by the community itself, with contributors electing team leads and members of the Technical Committee.

All project meetings are held in public IRC channels and recorded. Additional technical communication is through public mailing lists and is archived.

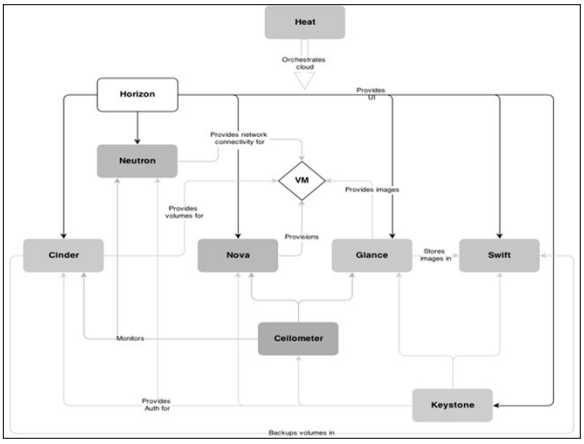
What you get with OS

OpenStack is an ***IaaS* cloud computing project** that is free open-source software.

Providing infrastructure means that OpenStack makes it easy for users to **quickly add new instance, upon which other cloud components can run**. Typically, the infrastructure then runs a "platform" upon which a developer can create software applications that are delivered to the end users.

The software platform consists of **interrelated components that control diverse, multi-vendor hardware pools of processing, storage, and networking resources throughout a data center**.

Users either manage it through a web-based dashboard, through command-line tools, or through a RESTful Application Programming Interface (API). OpenStack.org released it under the terms of the Apache License.



OpenStack Projects

innovate achieve lead

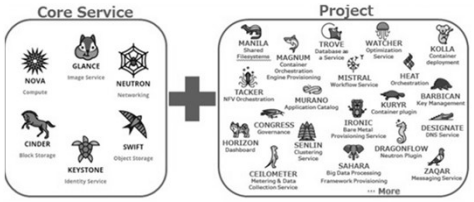
- The OpenStack platform is actually composed of multiple components, called **projects**.
- Each project is managed by a **technical committee** and the **OpenStack Foundation decides** which **projects are ready to be included in the OpenStack core**.
- These **projects work together** to provide the **services required** to deliver the Cloud.
- OpenStack has the following set of resources available to setup the cloud infrastructure:
  - Compute Resources
  - Network Resources
  - Block Storage
  - Identity Management
  - Image Resources
  - Object Storage
  - Dashboard
  - Shared Services
  - New projects are being added with each release and as the OpenStack community calls for them. New projects underway include metering, application orchestration, and database-as-a-service.

Edition	Release name	Release date	Component
1	Austin	21-10-2010	Nova, Swift
2	Bexar	03-02-2011	Nova, Glance, Swift
5	Essex	05-04-2012	Nova, Glance, Swift, Horizon, Keystone
6	Folsom	27-09-2012	Nova, Glance, Swift, Horizon, Keystone, Quantum, Cinder
7	Havana	17-10-2013	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer
8	Icehouse	17-04-2014	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove
9	Juno	16-10-2014	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove, Sahara
14	Newton	06-10-2016	Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove, Sahara, Ironic, Zaqar, Manila, Designate, Barbican, Searchlight, Magnum, aodh, cloudkitty, congress, freezer, mistral, monasca-api, monasca-log-api, murano, panko, senlin, solum, tacker, vitrage, watcher

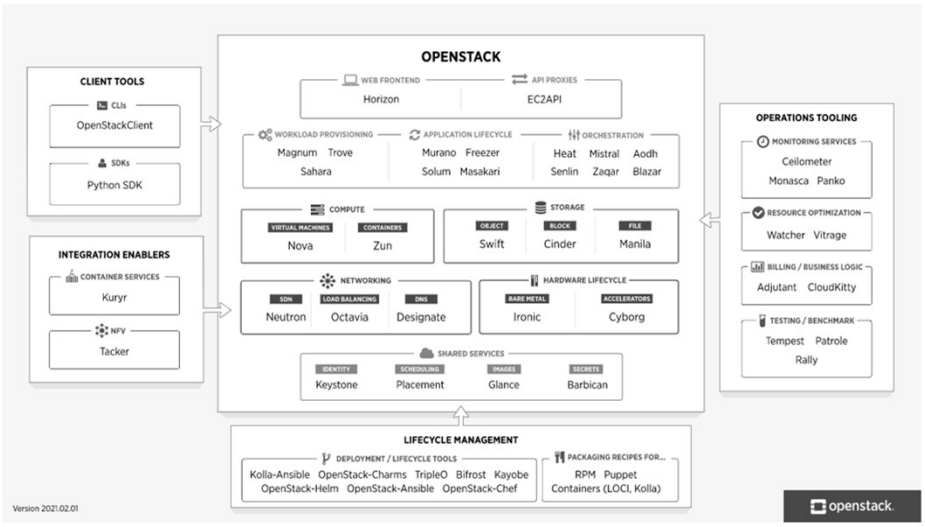
OpenStack Projects

innovate achieve lead

Service	Project name	Description
Dashboard	Horizon	Provides a web-based self-service portal to interact with underlying OpenStack services, such as launching an instance, assigning IP addresses and configuring access controls.
Compute	Nova	Manages the lifecycle of compute instances in an OpenStack environment. Responsibilities include spawning, scheduling and decomissioning of virtual machines on demand.
Networking	Neutron	Enables network connectivity as a service for other OpenStack services, such as OpenStack Compute. Provides an API for users to define networks and the attachments into them. Has a pluggable architecture that supports many popular networking vendors and technologies.
Storage		
Object Storage	Swift	Stores and retrieves arbitrary unstructured data objects via a RESTful, HTTP based API. It is highly fault tolerant with its data replication and scale out architecture. Its implementation is not like a file server with mountable directories.
Block Storage	Cinder	Provides persistent block storage to running instances. Its pluggable driver architecture facilitates the creation and management of block storage devices.
Shared services		
Identity service	Keystone	Provides an authentication and authorization service for other OpenStack services. Provides a catalog of endpoints for all OpenStack services.
Image Service	Glance	Stores and retrieves virtual machine disk images. OpenStack Compute makes use of this during instance provisioning.
Telemetry	Ceilometer	Monitors and meters the OpenStack cloud for billing, benchmarking, scalability, and statistical purposes.
Higher-level services		
Orchestration	Heat	Orchestrates multiple composite cloud applications by using either the native HOT template format or the AWS CloudFormation template format, through both an OpenStack-native REST API and a CloudFormation-compatible Query API.
Database Service	Trove	Provides scalable and reliable Cloud Database-as-a-Service functionality for both relational and non-relational database engines.



OpenStack Reference Model



OpenStack Projects

Nova

It provides virtual servers upon demand. Nova is the most complicated and distributed component of OpenStack. A large number of processes cooperate to turn end user API requests into running virtual machines.

Swift

Swift service providing object storage which allows users to store and retrieve files.
Swift architecture is distributed to allow for horizontal scaling, and to provide redundancy as failure-proofing.
Data replication is managed by software, allowing greater scalability and redundancy than dedicated hardware.

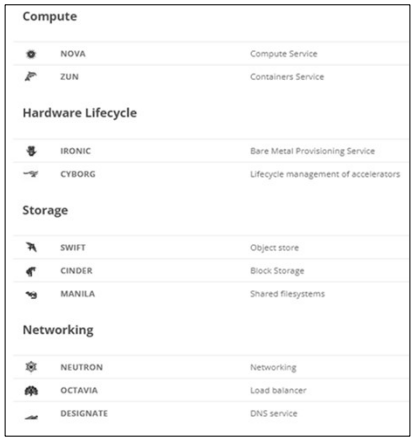
Glance

Glance service that acts as a registry for virtual machine images to allowing users to copy server images for immediate storage.
Images can be used as templates when setting up new servers. Usually the images are stored in the Swift (Object) service.

OpenStack Services

innovate achieve lead

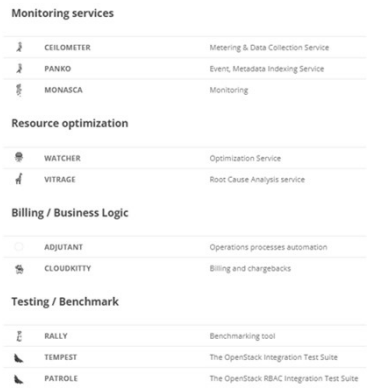
An OpenStack deployment contains a number of components providing APIs to access infrastructure resources. This page lists the various services that can be deployed to provide such resources to cloud end users.



OpenStack Services

innovate achieve lead

Tooling :Those services deliver APIs primarily targeted to cloud admins and deployers, to help with cloud operations.



Software in this section facilitates integration of OpenStack components in adjacent open infrastructure stacks