

→ Cloud Service Models

- a. Over the cloud, the remote server has to process, manage & store the data, locally and at the data center
- b. And the cloud provider typically charges for the services user has used
- c. These services are further divided into five categories:
 - i. SAAS (Software As-A Service)
 - ii. PAAS (Platform As-A Service)
 - iii. IAAS (Infrastructure As-A service)
 - iv. FAAS (Function As-A Service)
 - v. XAAS (Anything As-A Service)

→ IAAS (Infrastructure As-A Service)

- a. These services are a set of compute, storage & network that are virtualized by the cloud provider, so that user can access and configure resources according to their needs
- b. A user can rent infrastructure in IAAS
- c. This is also called as Hardware As-A Service, as it provides compute/infrastructure over the network
- d. In this, customer pay for the services based on their hourly, weekly or monthly usage
- e. Customers are even charged for the virtual space they use
- f. IAAS provides the user operating system, security, server, deployment tools and database
- g. *Examples:*
 - AWS EC2, Google Compute Engine, Digital Ocean
- h. *The companies that provide Infrastructure As-A Service are*
 - AWS, Blue Stack, IBM, Open Stack, Rackspace, VMWare & many more
- i. *Advantages of IAAS:*
 - i. This service provides the infrastructure and user just have to install the operating system
 - ii. The user can modify the architecture as per their requirement
 - iii. The user has full control over the computing resources
 - iv. User don't have to worry about maintenance
 - v. Web hosting is less expensive than traditional way of hosting
- j. *Key points of IAAS:*
 - i. It allows the user to rent infrastructure like servers, routers, computers, etc.
 - ii. IAAS provider manages the data center

→ PAAS (Platform As-A Service)

- a. This platform is created for the programmers to develop, test & manager the applications
- b. Here cloud providers provide an on-demand environment for developing the software applications
- c. In PAAS, user can build, run, manage Application Program Interface (APIs)
- d. PAAS are hosted in cloud and accessed by the user via browser

- e. The provider host the hardware & software on their own infrastructure, as a result, users are free from installing in-house hardware & software to develop & runs the applications
- f. Here, the user has full control over deployed application & configuration files & setting up the application
- g. *Examples*
Windows Azure, Google App Engine, AWS BeanStalk, OpenSwift, etc
- h. *Advantages of PAAS:*
 - i. Pre-built platform are provided in PAAS, where user only has to access the application
 - ii. It is way simpler module to use & deploy any application/services\
 - iii. As platforms are already built, the user only has to access the services & create the applications
 - iv. It is more time efficient
- i. *Disadvantages of PAAS:*
 - i. Migrating one user application from one PAAS provider is very complex
- j. *Key Points of PAAS:*
 - i. It integrates best service & data base
 - ii. Platforms are built on virtual technologies, so user can scale up & down as per their requirement

→ SAAS (Software As-A Service)

- a. These cloud services provide the user with complete software applications over the internet
- b. All infrastructure, application tool, data, etc are located at data center, managed by service provider
- c. Here, the user doesn't have to install or maintain any software, they only have to use it
- d. These services are available to the user over cloud
- e. *Examples of SAAS Providers*
 - i. Document management like Google Docs, etc
 - ii. Mail services like Gmail, Ymail etc.
 - iii. Social Networks Like FB, etc
 - iv. Business services
 - v. Google App, Azure, Dropbox
- f. *SAAS is further is divided into 2 models*
 - i. Simple Multi-Tenancy
 - 1. Each user has independent resources, that are different from other user
 - ii. Fine Grain Multi-Tenancy
 - 1. The resources are shared among several users but the functionalities remain the same
- g. *Advantages of SAAS:*
 - i. Easy to access
 - 1. User can access the application from anywhere
 - ii. Low cost maintenance
 - 1. User doesn't have to update & maintain the applications

- iii. User don't require hardware to install the application
- iv. SAAS can be accessed from multiple devices
- h. *Disadvantages of SAAS:*
 - i. User doesn't have control over SAAS application
 - ii. User can only access when they've internet connection
 - iii. Switching between SAAS providers is very difficult
- i. *Key Points of SAAS:*
 - i. A complete software application is provided over internet
 - ii. Components such as infrastructure, tools, data are located at data centers, and managed by service or cloud providers

→ IAAS vs PAAS vs SAAS

IAAS	PAAS	SAAS
It provides you a virtual data center to store information & provide a platform for app development, testing & deployment	It provides virtual platform, tools to test & deploy virtual applications	It provides an application to compute business tasks
It is used by System Admins or IT Admins	It is used by developers	It is used by end users
It provides you infrastructure	It provides you infrastructure & platform	It provides you infrastructure, platform & software

→ On-Premise for IAAS, PAAS, SAAS

- a. On-Premise: This is managed by user or an organization
- b. IAAS : Application, Data, Runtime, Middleware & OS are managed by user or organization
- c. PAAS : Application & Data are managed by user or organization
- d. SAAS : Everything is managed by the vendor

On-Premise	IAAS	PAAS	SAAS
Application	Provided	Provided	Provided
Data	Provided	Provided	Not Provided
Runtime	Provided	Not Provided	Not Provided
Middleware	Provided	Not Provided	Not Provided
OS	Provided	Not Provided	Not Provided
Virtualization	Not Provided	Not Provided	Not Provided

Server	Not Provided	Not Provided	Not Provided
Storage	Not Provided	Not Provided	Not Provided
Networking	Not Provided	Not Provided	Not Provided

→ FAAS (Function As-A Service)

- a. In this kind of model, users are allowed to develop or create a piece of code in the cloud without worrying about the infrastructure.
- b. Microservices are developed over here
- c. *Advantages of FAAS:*
 - i. You can scale up the application as per the requirement, so developer doesn't have to worry about the capacity & server management
 - ii. It is cost effective

→ XAAS (Anything & Everything As-A Service)

- a. These models provide widely used services to the user, user can take benefit of large number of softwares tools, hardware resources, products, technologies at affordable cost without buying them
- b. XAAS provides all kinds of services like PAAS, SAAS, FAAS, CAAS (Communication As-A Service), DBAAS (Database As-A Service), SECAAS(Security As-A Service), STAAS (Storage As-A Service), DRAAS(Disaster Recovery As-A Service), etc.
- c. *Advantages of XAAS:*
 - i. All the above model's advantages
- d. *Disadvantages of XAAS:*
 - i. It gets slower when there are 'n' number of users using the same resource
 - ii. User or organizations rely on XAAS service providers for maintenance (if they're not available, user's/organization's problems will not be resolved)

→ Life Cycle of Cloud Solution

- a. **Defining the purpose**
 - i. This means to understand the requirements of business to determine what type of applications or what type of programs, user or organization need to build or run an application on cloud
- b. **Defining the Hardware**
 - i. Choose the computer service that will provide the right support where you can scale up or down according to the requirement to run an application
 - ii. *Examples*
EC2, Lambda(Serverless Computing), Elastic containers
- c. **Defining the storage**
 - i. Choose storage service where you can have the backup and archive of you data over the internet

ii. Examples

S3, EFS(Serverless Elastic File System), Glacier(for archiving)

d. Defining the network

- i. Defining the network that'll securely deliver data and application with low latency and high transfer speeds

ii. Examples

1. VPC (Virtual Private Cloud, which are used to provide you network routes)
2. Route53 (This provides name services DNS, this connects the user request to internet application running on AS port 53)
3. Direct Connection (sets a pipeline between Organization to Data Center)

e. Defining the security

- i. We set up security for authentication and authorization of user, this provides limited access to the user on certain AWS resources

ii. Examples

1. IAM (Identity & Access Management, to manage which user will have the access to our application)
2. KMS (AWS Key Management System) provides centralized control using crypto key to protect your data)
3. Cognito (The user can sign-in using mobile phone or web apps)

f. Defining the Management Process & Tool

- i. You have complete control over cloud management by defining the management tool which monitors AWS resources and customer application running on the platform

ii. Examples

1. Cloud Watch (to monitor & manage the service provided by the cloud platforms like infrastructure, data storage, etc.)
2. AutoScaling (This continuously monitors your application and checks if it is running properly or not)

→ Cloud based Architecture

There are two main components

a. Front-End

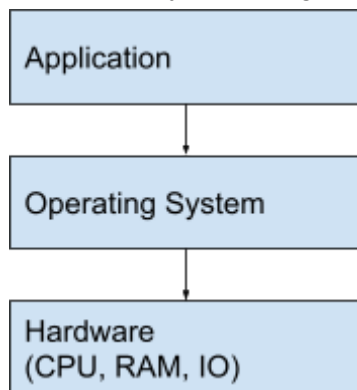
- i. Client / End-User
- ii. It consists of all the applications and interfaces that are used by client to access the cloud resources
- iii. A Front-End can be Web-Server, User, Mobile Devices

b. Back-End

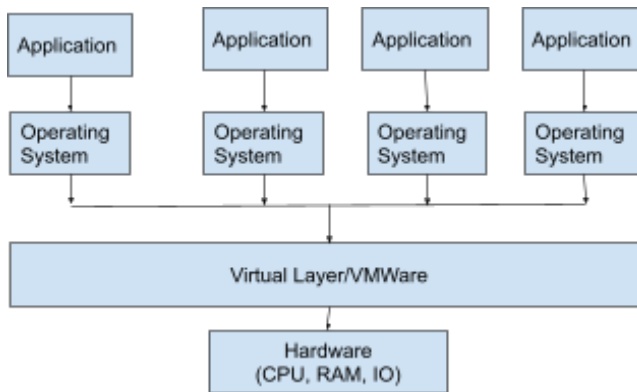
- i. It is the cloud itself, and it consists of infrastructure such as database, computing resources, deployment model, that are required to build a cloud
- ii. Components of backend:
 1. *Infrastructure*
 - a. compute hardware devices, server, network equipments
 2. *Storage*
 - a. Over the cloud, we have huge amounts of data to manage. These are managed through S3, EFS & Glacier
 3. *Services*
 - a. These are services provided by the cloud providers to manager cloud according to user/organization's requirement (IAAS, PAAS, SAAS, FAAS, XAAS)
 4. *Applications*
 - a. These are the software or platform that the client wants to access
 5. *Management*
 - a. This provides you with coordination between all the backend components.
 - b. This also monitors whether all the resources in the cloud are working properly or not
 6. *Security*
 - a. This provides security for infrastructure, provides authentication and authorization to define who can use the resources

→ Virtualization

- a. It creates virtual system which actually uses storage, operating, compute (CPU & RAM), network resources of host machine
- b. A single computer can have multiple operating systems running parallel all because of the virtualization layer
- c. Traditional System Diagram



d. Virtual System Diagram

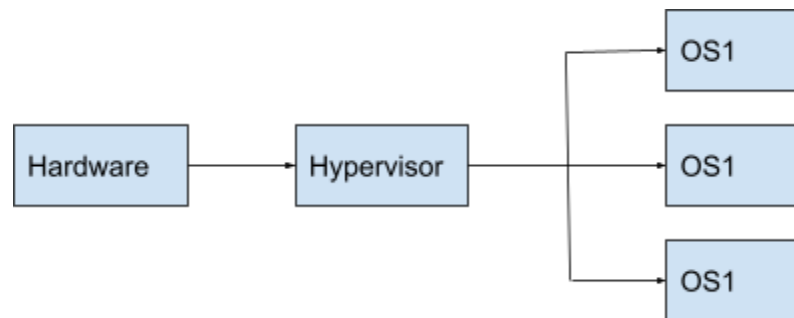


→ Hypervisor

- a. It is a form of virtualization software that are used to allocate the resource
- b. It is hardware virtualization technique that allows multiple guest OS to run on single host machine at the same time
- c. It can also be referred as Virtual Machine Manager
- d. There are two types of hypervisor

i. Type 1 Hypervisor

- 1. It just runs directly on host machine and it does not require any server based operating system
- 2. It has direct access to hardware resource

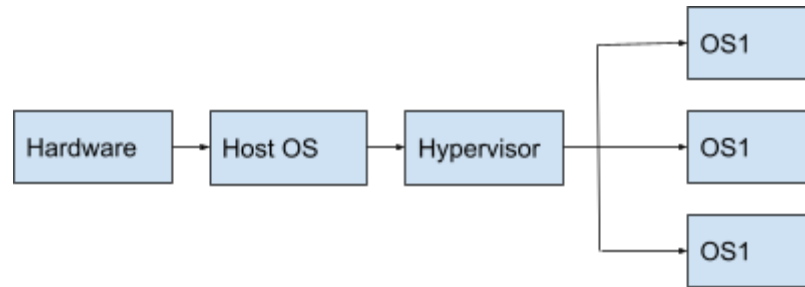


3. Example

- a. VMWare

ii. Type 2 Hypervisor

- 1. Host OS runs on host system
- 2. This kind of hypervisor does not directly runs on hardware, rather than they require application in host system
- 3. The software is installed on OS , the hypervisor makes hardware calls



4. Examples

- a. Oracle VirtualBox, VMWare Workstation

→ Types of Virtualization

- a. Hardware Virtualization
- b. Operating System Virtualization
- c. Server & Storage OS Virtualization

→ EC2 (Elastic Compute Cloud)

- a. EC2 is web service that provide secure & resizable compute capacity in a cloud
- b. EC2 allows you to configure the capacity also provide you the complete control of computing resources
- c. In this, user can scale up & down the resources as per their requirements
- d. As user create and environment or instance as per their requirement, this leads to less wastage of capacity
- e. EC2 is a virtual machine hosted on AWS
- f. There are many pricing options in EC2
 - i. On-Demand Instance
 1. These instances are mostly used for short term applications or unpredictable workload
 2. Here we pay by hours or minutes or seconds, depending on the type of instance
 3. Examples
 - a. Application being developed on EC2, This is the most flexible option to scale up & down
 - ii. Reserved Instance
 1. When the workload is fixed, and the extra requirement is known, then we use these kinds of instances
 2. *Types of Reserved Instances*
 - a. Standard Reserved Instances: Up to 70% Discount
 - b. Convertible Reserved Instances: Up to 55% Discount
 - c. Scheduled Reserved Instances: No Discount

iii. Spot Instance

1. If you urgently need large amount of computing capacity, this is the best instance
2. You can purchase unused capacity at a discount of 90% as well
3. The fluctuation of pricing will depend on supply & demand

iv. Dedicated Instance

1. A physical EC2 server is dedicated for the use, and this is the most expensive option

v. Saving Plans

1. User can get up to 72% of discount, regardless of their instance type
2. The user can commit to specific requirements for 2-3 years

→ EC2 machine hosting setup commands

- a. Open putty and write the ip in hostname
ec2-user@ipaddress
- b. Add key and certificate
Select SSH from left panel
SSH>Auth>Credentials
- c. Use these commands on SSH terminal

```
ec2-user@52.90.33.99
~$ sudo -i                                # run with root or superuser privileges
~# yum update -y                          # update system
~# yum install httpd                      # installing http daemon
~# systemctl status httpd                 # checking status of http daemon
~# systemctl stop httpd                   # stopping http daemon
~# systemctl start httpd                  # starting http daemon
~# systemctl enable httpd                 # enabling http daemon

~# cd /var/www/html
# change dir to /var/www/html to create html file
~# vi index.html
# editing index.html file with custom contents to be
# displayed on EC2 instance site

~# systemctl start httpd                  # starting http daemon
~# systemctl enable httpd                 # enabling http daemon
~# systemctl status httpd                 # checking status of http daemon

# visit ip (52.90.33.99) of EC2 instance to view the website hosted
# on EC2 instance
```

