What’s AWS?

Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage and other functionality to help businesses scale and grow. In simple words AWS allows you to do things like-

1. Running web and application servers in the cloud to host dynamic websites. – EC2
2. Securely store all your files on the cloud so you can access them from anywhere. – S3
3. Execute backend codes- Lambda
4. Send bulk email to your customers. – SNS
5. Data warehouse designed for large scale data set storage and analysis. - Redshift

S3 - Simple Storage Service

Overview:

Used to store and retrieve data, create buckets which store objects (actual data and metadata)

Store an infinite amount of data (objects) in a bucket.

Each object can contain up to 5 TB of data. Each object is stored and retrieved using a unique developer assigned key.

Concepts

- **Buckets**: A bucket is a container for objects stored in Amazon S3. Every object is contained in a bucket. Buckets help to better organize the data.
- **Objects**: Objects consist of object data and metadata. Max 5TB data in object.
- **Keys**: A key is the unique identifier for an object within a bucket. Every object in a bucket has exactly one key. The combination of a bucket, key, and version ID uniquely identify each object.

- **Regions:** You can choose any AWS Region where Amazon S3 will store the buckets that you create. You might choose a Region to optimize latency, minimize costs. Objects stored in a Region never leave the Region unless you explicitly transfer them to another Region.
- **Amazon S3 data consistency model:**

  Amazon S3 provides strong read-after-write consistency for PUTs and DELETEs of objects.
  Put: to overwrite an object
  Delete : to Delete it.
  Read-after-write consistency is the ability to view changes (read data) right after making those changes (write data).

Features

- **Storage classes:**

  S3 STANDARD for general-purpose storage of frequently accessed data
  S3 STANDARD_IA for long-lived, but less frequently accessed data
  GLACIER for long-term archive

- **Bucket policies:** A bucket policy is an (IAM) policy. You add a bucket policy to a bucket to grant other people access permissions for the bucket and the objects in it. Only the bucket owner is allowed to associate a policy with a bucket. (eg.only xyz ip address or xyz account can access). By default, all Amazon S3 buckets and objects are private.
- **AWS identity and access management (IAM):** (IAM) is a web service that helps you securely control access to AWS resources. You use IAM to control who is authorized (has permissions) to use resources. You can grant permissions, also add Multi Factor authentication.
- **Access control lists:** Control access to each of your buckets and objects using an access control list (ACL).

2 concepts: Versioning & Lifecycle rules:

Lifecycle rules provide you the ability to configure rules that define what automatically happens to objects stored in your buckets after a specific date or period of time.
Versioning is a means of keeping multiple variants of an object in the same bucket.

Versioning provides protection against overwrites and deletes by enabling you to preserve, retrieve, and restore every version of every object in an Amazon S3 bucket.

EC2 - Elastic Cloud Compute

Overview

They are like Amazon virtual servers. Using them eliminates the need to invest in hardware.

You can use EC2 to launch virtual servers, manage storage you can develop and deploy applications faster.

AWS EC2 instance is like renting a server from AWS on an hourly basis.

Use Cases:

- Hosting environments: hosting a variety of applications, software and websites on the cloud.
- Development and test environments: The scalable nature of EC2 means that organizations now have the ability to create and deploy large scale testing and development environments with ease.
- Backup and disaster recovery: Companies are leveraging EC2 as a medium for performing disaster recovery for both active and passive environments. The fact that the Amazon Elastic Compute Cloud can be turned up quickly in case of an emergency, means that businesses have access to a faster failover with minimal downtime for their applications.

Main Concepts

- **Elastic load balancing:** It helps to distribute incoming application traffic across multiple Amazon EC2 instances. If a server can't handle the traffic then we will add a replica of that server and balance the load between them using the load balancer.
- **Auto Scaling:** It allows us to scale our Amazon EC2 capacity up or down automatically according to conditions that we define. With Auto Scaling, we can ensure that the number of Amazon EC2 instances

we are using increases seamlessly during demand spikes to maintain performance and decreases automatically during demand lulls to minimize costs.

- Auto Scaling is responsible for the number of instances behind ELB however ELB is responsible for distributing traffic within the EC2 instances.

EC2 Steps to create an instance:

1. Choose an AMI. An ***Amazon Machine Image*** *(AMI)* is a template that is used to create new instances. It contains **software** information, Operation systems information, Volume information and access permissions.
    a. E.g. Linux AMI, Windows AMI
    b. 2 types of AMI. Predefined AMI and custom-made AMI
2. Choose an **instance type**. It specifies the **hardware** specification that is required. There are **5** types
    a. Compute Optimized- for situations that require lot of processing power
    b. Memory Optimized- for setting up something that has to do with in memory cache
    c. GPU Optimized- for setting up something with large graphical requirement (Gaming system)
    d. Storage optimized- for storage servers
    e. General Purpose- Not sure or when everything is balanced (CPU, memory, storage, and networking capacity)

3. **Configure the instance**. Here you have to specify a number of instances, kind of network, if you assign public IP to it, to do you assign any IAM role (authentication), specify shutdown behavior (stopping and terminating. Stopping is temporary shutting down the system. Terminating is permanently closing). Along with this there are payment options available.
4. Adding **storage**.
    a. Ephemeral storage (Temporary and free)
    b. Amazon Elastic Block store (permanent and paid)
    c. Integrate with S3

5. Add **tags**. They are used for identifying the machine.
6. **Configure security groups**. Specify a firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances.
7. Secure login information for your instances using ***key pairs*** (AWS stores the public key, and you store the private key in a secure place). At the basic level, a sender uses a public key to encrypt data, which its receiver then decrypts using another private key. These two keys, public and private, are known as a key pair.

Once the instance is created you can deploy your application on it.

Amazon Redshift

Overview

Amazon Redshift is a fully-managed petabyte-scale cloud-based data warehouse.

Designed for large scale data set storage and analysis.

Designed for Online Analytical Processing OLAP, not OLTP(which needs role based storage)

Cost effective data warehouse

Uses ODBC and JDBC connections to SQL or BI tools

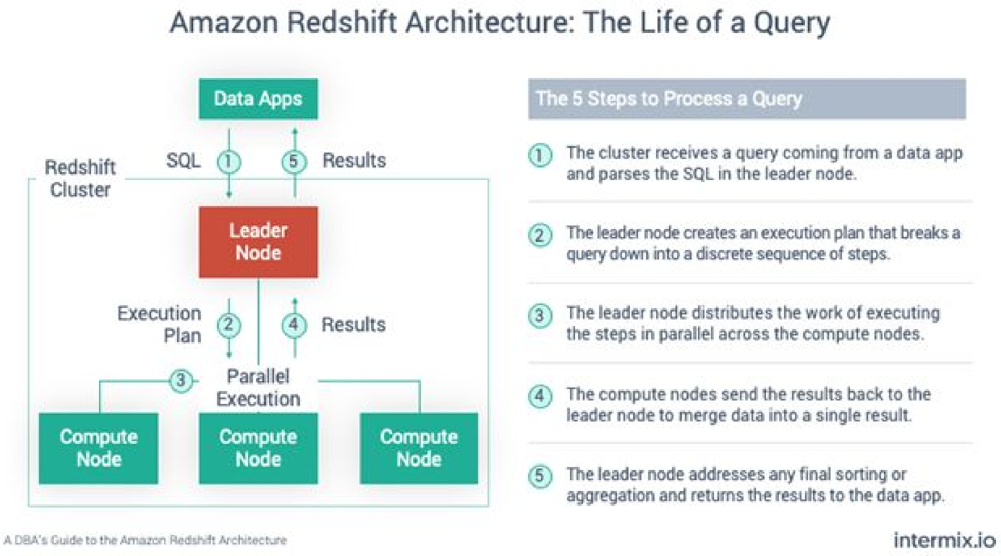
Monitor consumption using CloudWatch

Use Redshift to unify Data Warehouse and data lake

With Redshift Spectrum**,** an analyst can perform SQL queries on data stored in Amazon S3 buckets. This can save time and money because it eliminates the need to move data from a storage service to a database, and instead directly queries data inside an S3 bucket.

Working

Each Amazon Redshift contains multiple clusters. Every cluster has one leader node and multiple compute nodes.



Amazon Redshift Architecture: The Life of a Query

The 5 Steps to Process a Query

1. The cluster receives a query coming from a data app and parses the SQL in the leader node.
2. The leader node creates an execution plan that breaks a query down into a discrete sequence of steps.
3. The leader node distributes the work of executing the steps in parallel across the compute nodes.
4. The compute nodes send the results back to the leader node to merge data into a single result.
5. The leader node addresses any final sorting or aggregation and returns the results to the data app.

A DBA's Guide to the Amazon Redshift Architecture

intermix.io

Leader Node

The leader node has four major roles:

1. **Communication with data apps**

2. **Distribution of workloads**

3. **Caching of query results**: When a query is executed in Amazon Redshift, both the query and the results are cached in the memory of the leader node, across different user sessions to the same database. When query or underlying data have not changed, the leader node skips distribution to the compute nodes and returns the cached result, for faster response times.

4. **Maintenance of catalog tables**: The system catalogs store schema metadata, such as information about tables and columns. System catalog tables have a PG prefix. A query that references only catalog tables or that does not reference any tables, runs exclusively on the leader node.

Compute Nodes

The compute nodes handle all query processing, in parallel execution (“massively parallel processing”, short “MPP”).

Amazon Redshift provides two categories of nodes:

Dense compute nodes come with solid-state disk-drives (“SDD”) and are best for performance intensive workloads.
Dense storage nodes come with hard disk drives (“HDD”) and are best for large data workloads.
As your workloads grow, you can increase the compute and storage capacity of a cluster by increasing the number of nodes, upgrading the node type, or both.

2 important concepts Reasons for High Speed:

The ability to deliver this level of performance comes with the use of two architectural elements: columnar data storage and massively parallel processing design (MPP).

Massive Parallel Processing (MPP) Explained

Redshift's Massively Parallel Processing (MPP) design automatically distributes workload evenly across multiple nodes in each cluster, enabling speedy processing of even the most complex queries operating on massive amounts of data. Multiple nodes share the processing of all SQL operations in parallel, leading up to final result aggregation.

Columnar Data Storage Explained

Redshift's column-oriented database is designed to connect to SQL-based clients and business intelligence tools, making data available to users in real time.

Data is stored in the form of blocks in a data warehouse. For rows storage blocks are all the details of one entity. (e.g. Block 1: 1, bugs, bunny, NY….). In case of columnar storage blocks would be one whole column. This is how the no of blocks are reduced.

When you query, the data needs to be loaded from disk onto the nodes. Because of columnar storage this data that is to be loaded is reduced. This contributes to the optimization of analytic query performance.

2 imp terms:

Autoscaling by automatically adding or removing nodes. (provides faster querying)

Snapshots are point-in-time backups of a cluster. There are two types of snapshots: automated and manual. Amazon Redshift stores these snapshots internally in Amazon Simple Storage Service (Amazon S3) by using an encrypted Secure Sockets Layer (SSL) connection. AWS can restore the instance from a snapshot, Amazon Redshift creates a new cluster and imports data from the snapshot.

Snowflake

Overview

Snowflake is a cloud based analytic data warehouse.

Snowflake provides a data warehouse that is faster, easier to use.

Scalability: Scale data, user and workload

Elasticity - Dynamically scale up n down. (in traditional ones you cannot take advantage of the new node unless you restructure the data)

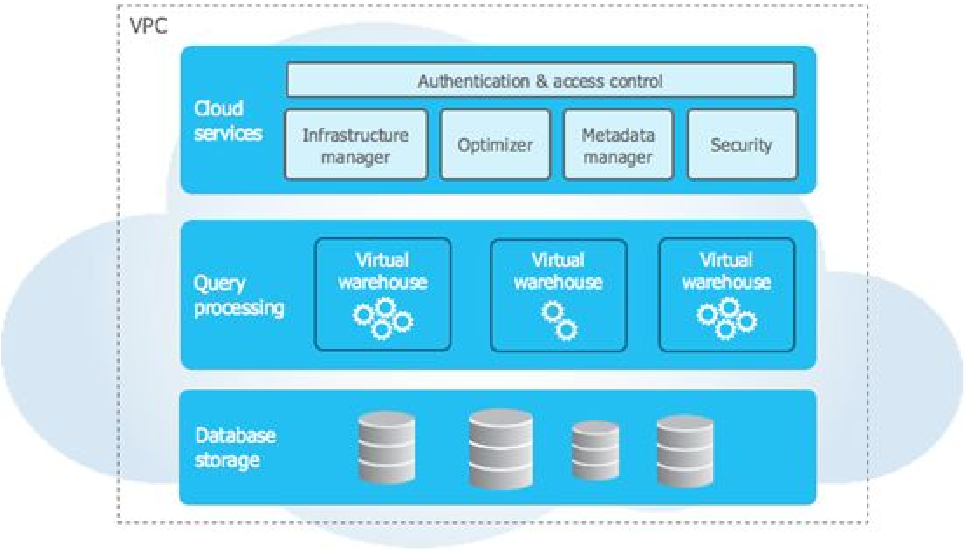
Pay as you use

Hold More types of data in single location

Better supports semistructured data - JSON, XML

Snowflake uses virtual compute instances for its compute needs and a storage service for persistent storage of data. Snowflake cannot be run on private cloud infrastructures (on-premises or hosted).

Snowflake Architecture



Database Storage

When data is loaded into Snowflake, Snowflake reorganizes that data into its internal optimized, compressed, columnar format. Snowflake stores this optimized data in cloud storage.

Snowflake manages all aspects of how this data is stored — the organization, file size, structure, compression, metadata, statistics, and other aspects of data storage are handled by Snowflake. The data objects stored by Snowflake are not directly visible nor accessible by customers; they are only accessible through SQL query operations run using Snowflake.

Query Processing

Query execution is performed in the processing layer. Snowflake processes queries using “virtual warehouses”. Each virtual warehouse is an MPP compute cluster composed of multiple compute nodes allocated by Snowflake from a cloud provider.

Each virtual warehouse is an independent compute cluster that does not share compute resources with other virtual warehouses. As a result, each virtual warehouse has no impact on the performance of other virtual warehouses.

Cloud Services

The cloud services layer is a collection of services that coordinate activities across Snowflake. These services tie together all of the different components of Snowflake in order to process user requests, from login to query dispatch. The cloud services layer also runs on compute instances provisioned by Snowflake from the cloud provider.

Difference between Snowflake and Redshift?

1. **Features**: bundled or not? Redshift bundles compute and storage to provide the immediate potential to scale to an enterprise-level data warehouse. But by splitting computation and storage and offering tiered editions, Snowflake provides businesses the flexibility to purchase only the features they need while preserving the potential to scale.
2. **JSON**: dealbreaker or no big deal? When it comes to JSON storage, Snowflake’s support is decidedly more robust than Redshift. This means that with Snowflake you can store and query JSON with native, built-in functions. When JSON is loaded into Redshift, it’s split into strings, which makes it harder to work with and query.

3. **Security**: Redshift includes a deep bench of customizable encryption solutions, but Snowflake provides security and compliance features oriented to its specific editions so that you have the level of protection most relevant to your data strategy.
4. **Maintenance**: automated or hands-on? Redshift requires more hands-on maintenance for a greater range of tasks that can't be automated, such as data vacuuming and compression. Snowflake has the advantage in this regard: it automates more of these issues, saving significant time in diagnosing and resolving issues.
5. **Integration**. If you are using AWS ecosystem redshift is easy to integrate as compared to snowflake. But for other software's Apache Spark, IBM Cognos, Qlik, and Tableau, Snowflake is easy to use.
6. Redshift doesn't support semi-structured data types like Array, Object. But Snowflake does.
7. Redshift Varchar limits data types to 65k characters. You also have to choose the column length ahead. In Snowflake, Strings are limited to 16MB and the default value is the maximum String size (so there's no performance overhead).

Lambda

AWS Lambda is used to execute backend code without worrying about the underlying architecture, you just upload the code and it runs, it's that simple!

It does so while automatically managing the AWS resources. When we say 'manage', it includes launching or terminating instances, health checkups, auto scaling, updating or patching new updates etc.

You pay only for the compute time you consume - there is no charge when your code is not running.

You can use AWS Lambda

- to run your code in response to events, such as changes to data in an Amazon S3 bucket;
- to HTTP requests using Amazon API Gateway;
- to invoke your code using API calls.

- Data processing. Lambda functions are optimized for event-based data processing. It is easy to integrate AWS Lambda with data sources like Amazon DynamoDB and trigger a Lambda function for specific kinds of data events.
- Task automation. With its event-driven model and flexibility, AWS Lambda is a great fit for automating various business tasks that don't require an entire server at all times. This might include running scheduled jobs that perform cleanup in your infrastructure, processing data from forms submitted on your website, or moving data around between different datastores on demand.

SNS Simple Notification Service

Amazon Simple Notification Service (Amazon SNS) is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients. In Amazon SNS, there are two types of clients—publishers and subscribers—also referred to as producers and consumers.

To get started with Amazon SNS, developers first have to create a topic, which is an access point for subscribers who are interested in receiving notifications about a specific subject. Developers publish a message to a topic when they have an update for subscribers and this action prompts Amazon SNS to distribute the message to all appropriate subscribers.

S3 + SNS Use case:

You first create a topic. Then add subscribers. Then create an S3 bucket that has the content. Create an event in S3 that triggers the SNS.

Data Warehouse v/s Database

DW is specially designed for analytics. It stores the data as well as finds trends in it. Database is just storage.

Data Warehouse vs Data Lake

Data lake is centralised repository for all data (both structured and unstructured data).

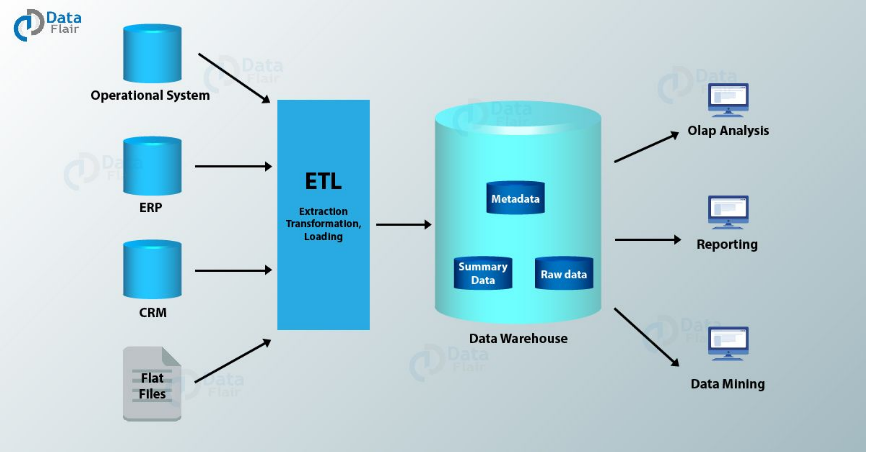
Data Warehouse only structured data.

Data lake is slow in processing. Data Lake is fast.

DW -There is no predefined schema.   DL- There is a predefined schema.

Data Mart vs Data warehouse.

Data mart is a warehouse. But it's only for a specific team. Data mart is smaller and much more focused.



OLTP vs OLAP

BASIS FOR COMPARISON	OLTP Online transaction processing	OLAP Online analytical processing
Basic	It is an online transactional system and manages database modification.	It is an online data retrieving and data analysis system.
Focus	Insert, Update, Delete information from the database.	Extract data for analyzing that helps in decision making.

Data	OLTP and its transactions are the original source of data.	Different OLTPs database becomes the source of data for OLAP.
Transaction	OLTP has short transactions.	OLAP has long transactions.
Time	The processing time of a transaction is comparatively less in OLTP.	The processing time of a transaction is comparatively more in OLAP.
Queries	Simpler queries.	Complex queries.
Normalization	Tables in OLTP databases are normalized (3NF).	Tables in the OLAP database are not normalized.
Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not affected.

AWS Cloud certification prep

SAAS PAAS IAAS

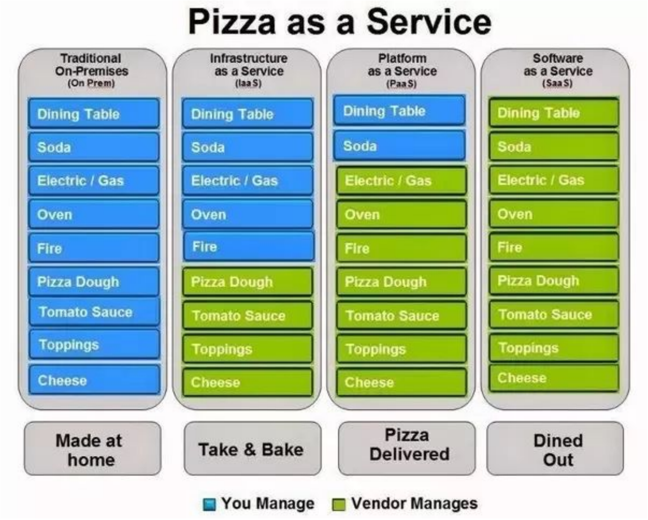
- IaaS: cloud-based services, pay-as-you-go for services such as storage, networking, and virtualization.
- PaaS: hardware and software tools available over the internet.
- SaaS: software that's available via a third-party over the internet.
- On-premise: software that's installed in the same building as your business.

SaaS examples: BigCommerce, Google Apps, Salesforce, Dropbox, MailChimp, ZenDesk, DocuSign, Slack, Hubspot.

PaaS examples: AWS Elastic Beanstalk, Heroku, Windows Azure (mostly used as PaaS), Force.com, OpenShift, Apache Stratos, Magento Commerce Cloud.

IaaS examples: AWS EC2, Rackspace, Google Compute Engine (GCE), Digital Ocean, Magento 1 Enterprise Edition*.

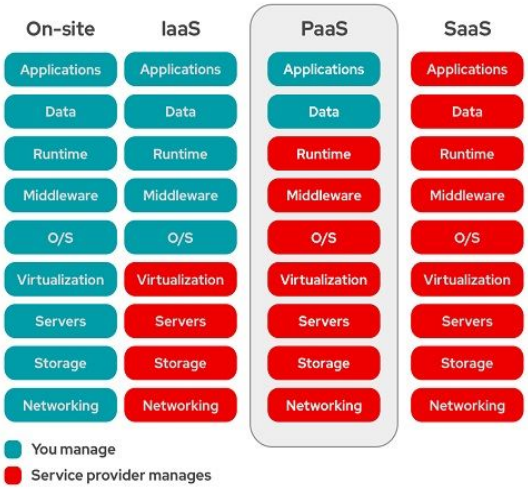
SAAS PAAS IAAS in layman's terms



Pizza as a Service

Traditional On-Premises (On Prem)	Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
🟦 Dining Table	🟦 Dining Table	🟦 Dining Table	🟩 Dining Table
🟦 Soda	🟦 Soda	🟦 Soda	🟩 Soda
🟦 Electric / Gas	🟦 Electric / Gas	🟩 Electric / Gas	🟩 Electric / Gas
🟦 Oven	🟦 Oven	🟩 Oven	🟩 Oven
🟦 Fire	🟦 Fire	🟩 Fire	🟩 Fire
🟦 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough	🟩 Pizza Dough
🟦 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce	🟩 Tomato Sauce
🟦 Toppings	🟩 Toppings	🟩 Toppings	🟩 Toppings
🟦 Cheese	🟩 Cheese	🟩 Cheese	🟩 Cheese
Made at home	Take & Bake	Pizza Delivered	Dined Out

🟦 You Manage 🟩 Vendor Manages



On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

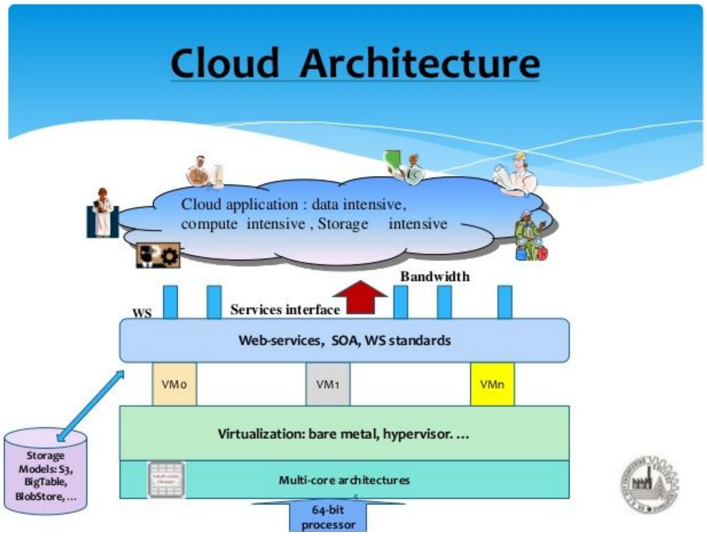
🟦 You manage
🟥 Service provider manages

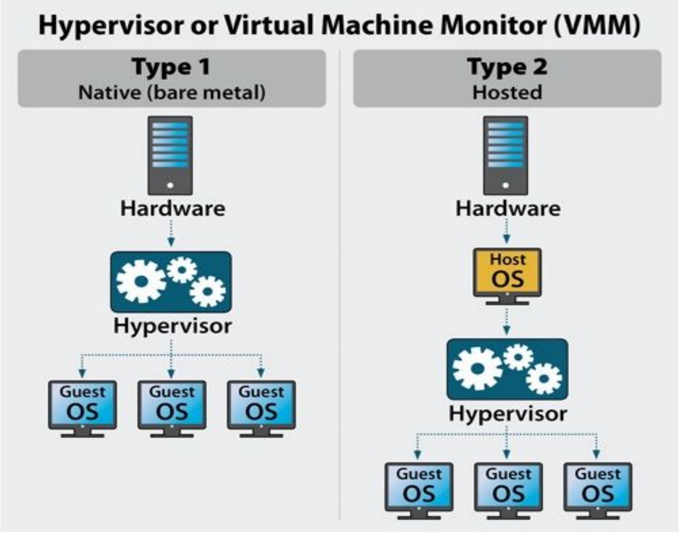
Cloud Computing: Virtualized compute environment (CPU + RAM) over the internet/ intranet.

Virtualization: This basically makes one physical computer act like many computers.

One PC mainly consists: CPU (Processor), Memory (RAM), Storage (Hard disk), Network

Server: CPU & RAM


Cloud Architecture
Cloud application : data intensive, compute intensive , Storage intensive
Bandwidth
WS
Services interface
Web-services, SOA, WS standards
VM0
VM1
VMn
Virtualization: bare metal, hypervisor. …
Multi-core architectures
64-bit processor
Storage Models: S3, BigTable, BlobStore, …



https://www.youtube.com/watch?v=_pPlanX5wQY

Virtualization-techniques-in-cloud-computing.

AWS uses type 1 hypervisors (XEN), azure uses (Hyper V)

Auto-Scalability: Vertical and Horizontal scalability

https://www.udemy.com/course/aws-certified-cloud-practitioner/