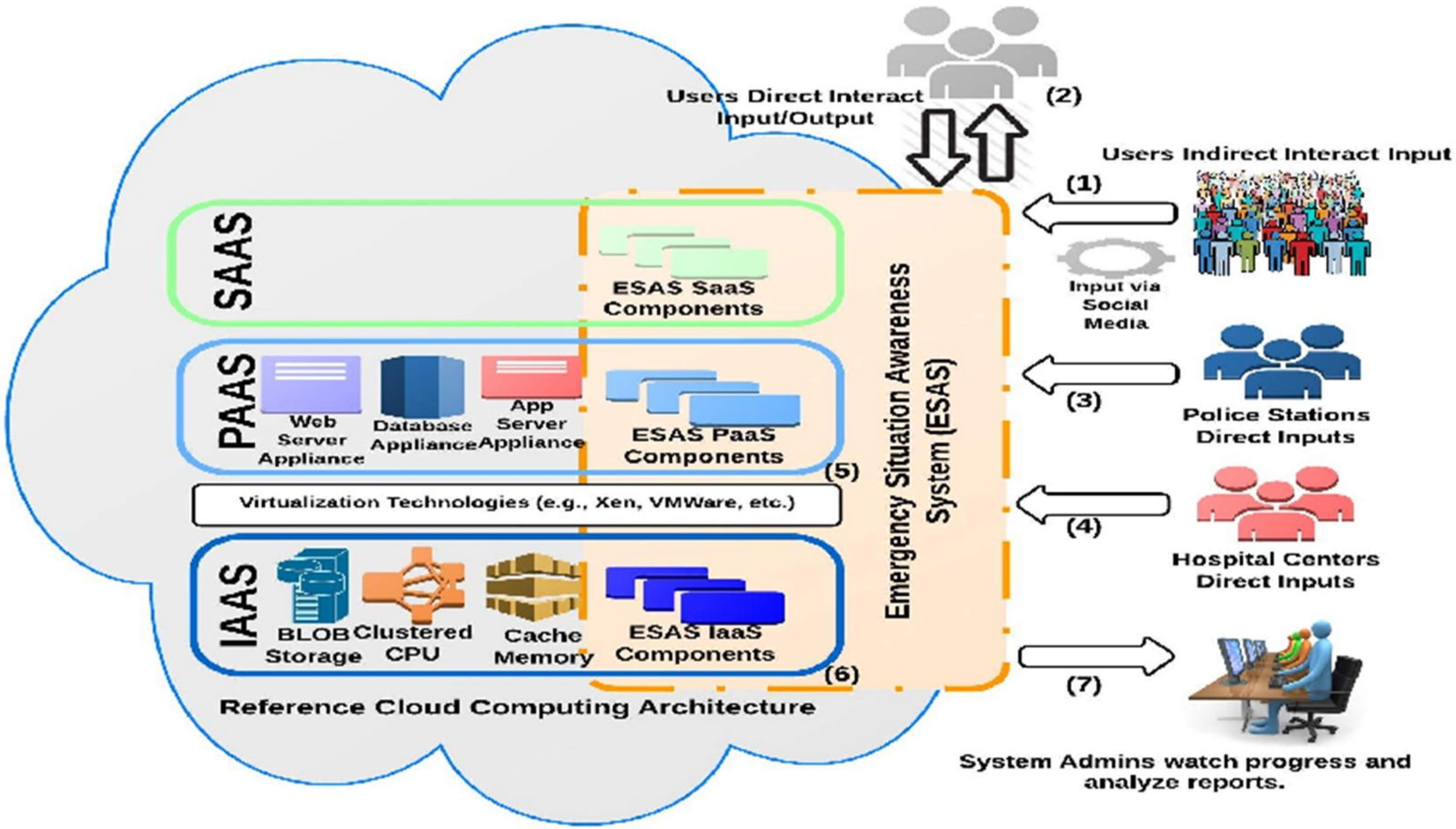
CSE407R01
CLOUD COMPUTING

UNIT 1

What we are going to discuss??

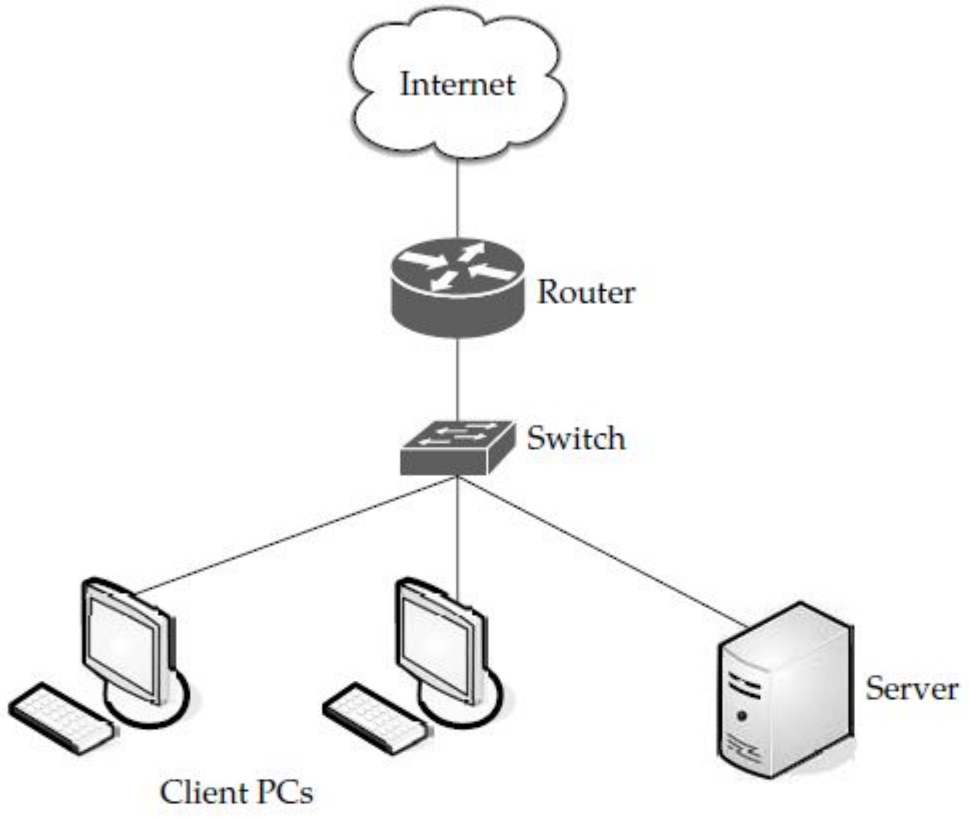
- To learn about the basic concepts of new evolving/emerging technologies in cloud computing
- To learn about the fundamentals of the cloud computing ecosystem and its characteristics
- To learn about the advantages and disadvantages of cloud computing
- To evaluate the cloud's business impact and economics
- To identify the difference between cluster, grid and cloud computing
- To identify the drivers of cloud computing adoption and discuss future of cloud (FoC)

Introduction to cloud computing

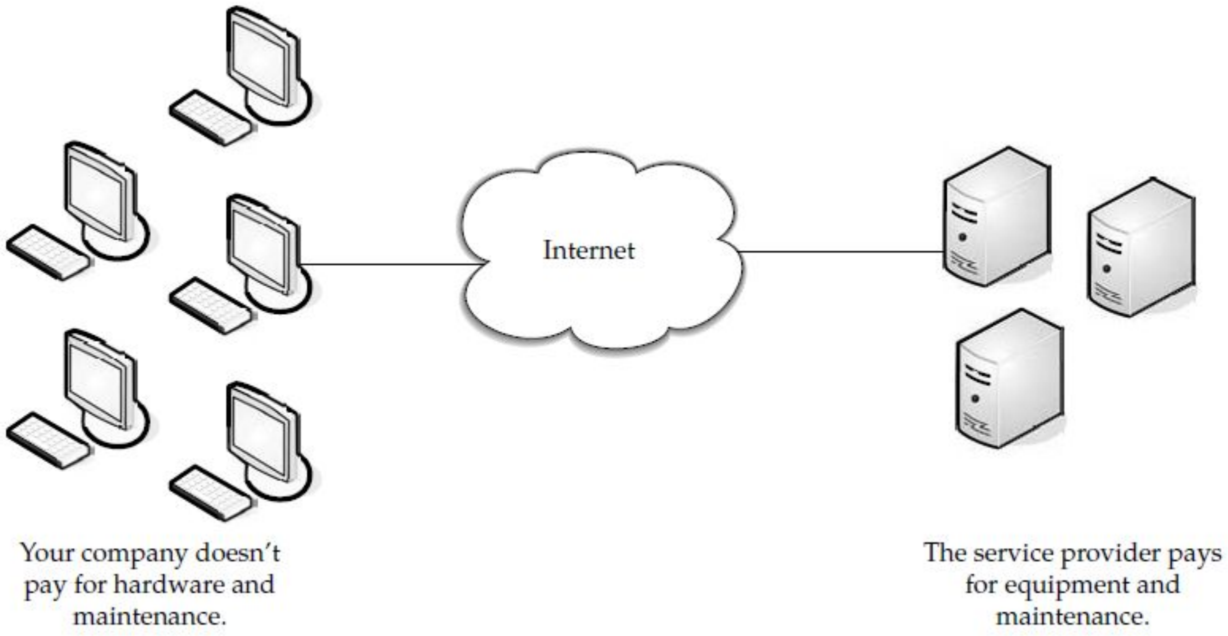


Cloud Computing: Cloud Components

- Cloud computing is an **UMBRELLA** term used to **refer to Internet based development and services.**
- A number of characteristics define cloud data, applications services and infrastructure
- **Remotely hosted:** Services or data are hosted on remote infrastructure.
- **Ubiquitous (found Everywhere):** Services or data are available from anywhere.
- **Commodified:** The result is a utility computing model similar to traditional that of traditional utilities, like gas and electricity - you pay for what you would want!



A cloud is used in network diagrams to depict the Internet



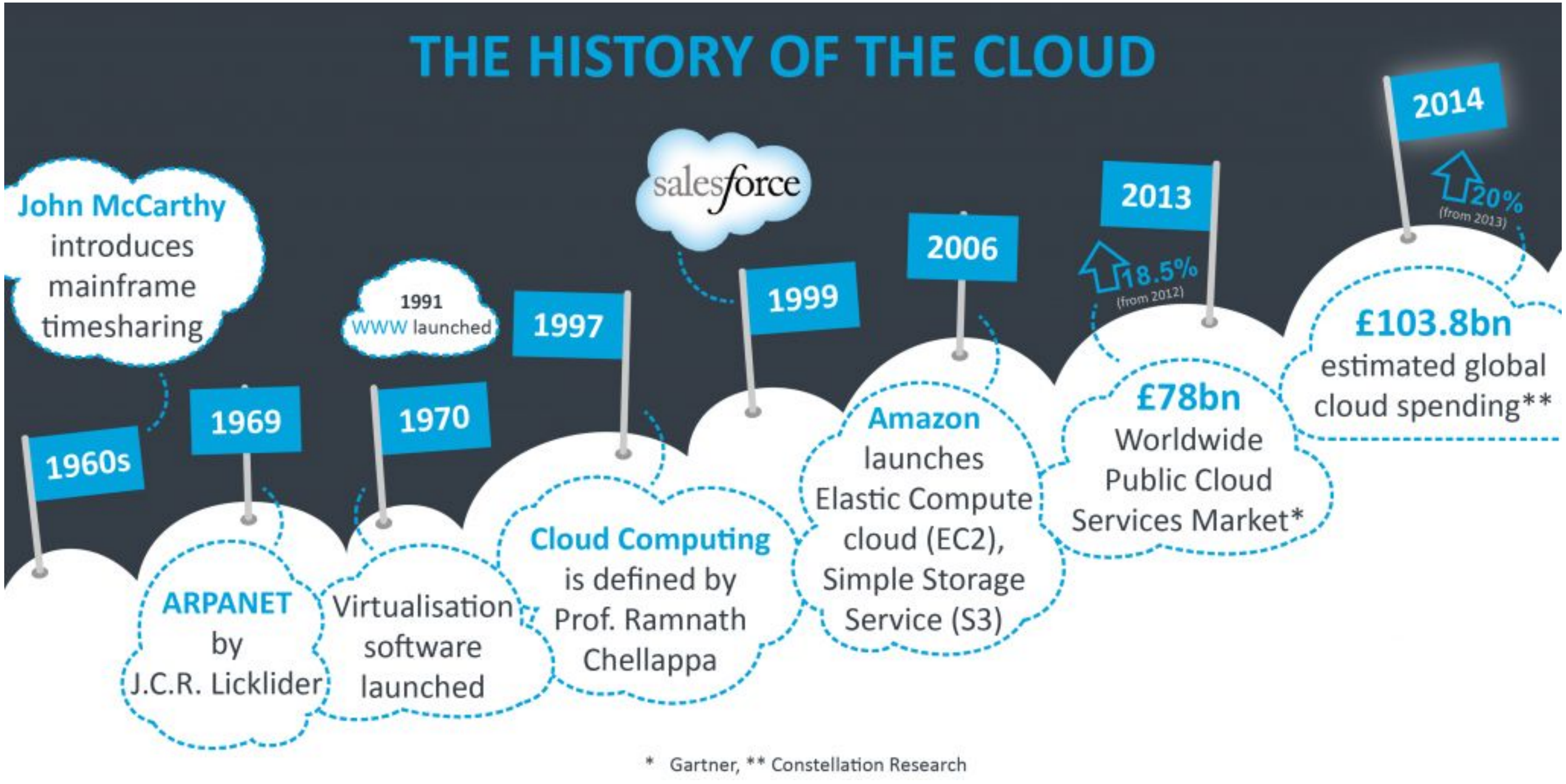
64% of companies have reduced waste and lowered energy consumptions after shifting to the cloud increased data efficiency and utilization

Layer	Description	Examples
INFRASTRUCTURE	**Hardware** – Physical devices in a data center providing a foundation for the model.	Cisco UCS, HP ConvergedSystem, VCE vBlock, etc.

Layer	Description	Examples
HYPERVISOR	**Virtualization** – Provides virtualized compute, storage, and networking.	Hyper-V, KVM, Xen, NSX, ACI, etc.
INFRASTRUCTURE	**Hardware** – Physical devices in a data center providing a foundation for the model.	Cisco UCS, HP ConvergedSystem, VCE vBlock, etc.

Layer	Description	Examples
Software-Defined Data Center	**Cloud API** – Enables the creation of virtualized assets tied to resource pools or users.	vSphere, OpenStack, AWS, Azure, GCP, etc.
HYPERVISOR	**Virtualization** – Provides virtualized compute, storage, and networking.	Hyper-V, KVM, Xen, NSX, ACI, etc.
INFRASTRUCTURE	**Hardware** – Physical devices in a data center providing a foundation for the model.	Cisco UCS, HP ConvergedSystem, VCE vBlock, etc.

History of cloud computing



Fundamentals of the cloud computing ecosystem

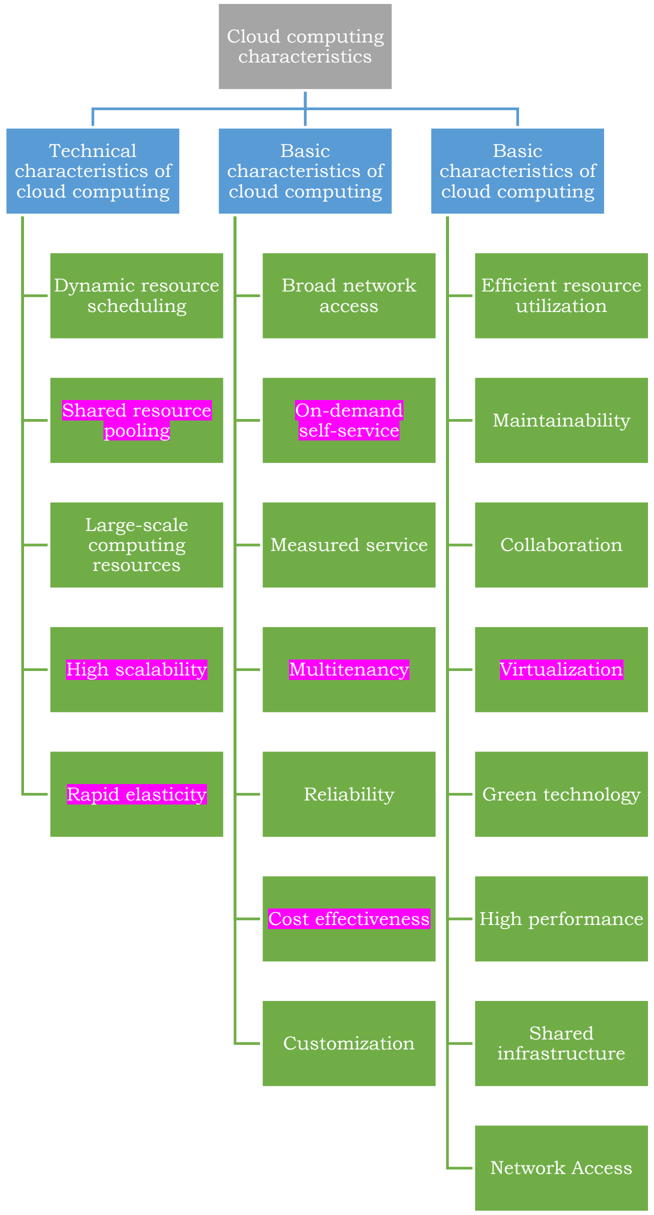
- The Cloud Computing Ecosystem (CCE) is a **dynamic** and **complex community** of the cloud computing system components and the stakeholders.
- The interdependent components of the CCE include
  - *Cloud Consultants, Cloud Service Providers, Cloud End-users (Customers), Cloud Product Manufacturers, Cloud Engineers, Cloud Partners, High Speed Network, The Cloud Management Environment As Well As The Cloud Computing Infrastructures And IT Resources That Are Provisioned As Services.*

Roles	Responsibilities
Cloud stakeholders	The three primary stakeholders of the cloud include **the end users, the cloud users and the cloud providers**
Cloud service providers	It provide and **render on-demand, payas-you-go** utility computing services to cloud users
The cloud users	The provisioned services are used by the cloud users to develop personalized products and web
The end users	The **direct consumers** of the products developed by the cloud users
Cloud service brokers	Influencers, professional service organizations, technology consultants, **registered brokers and agents** that assist cloud users to choose appropriate cloud computing solutions that best suit their organizational needs
Cloud resellers	Expansion of cloud service provisioning business globally a reality in the cloud market
Cloud consumers	The main stakeholders of the cloud ecosystem. **End users, customers of cloud resellers**, providers or brokers are the major cloud consumers. The cloud consumer could be a person, **a group of people or an organization that subscribes to and uses cloud service(s)**

Roles	Responsibilities
Cloud carrier	• The intermediate communication medium between **the cloud provider and the consumer.** • It makes cloud services accessible to cloud consumers through network connectivity and network access devices like mobile phones, laptops and other internet-enabled digital devices
Cloud brokers	Cloud brokers are entities that facilitate **efficient and effective use of cloud services while ensuring peak performance and seamless delivery** of such services
Service intermediation	Cloud broker enhances a given service by improving some specific capabilities of a cloud service like access management, performance reporting, and identity management and providing rewards or value-added services to the cloud consumers
Service aggregation	A cloud broker consolidates a number of fixed cloud services into one or new services
Service arbitrage	This is similar to service aggregation except that the services being aggregated are not fixed

Cloud computing characteristics

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA



Advantages and disadvantages of cloud computing

Advantages

- **Economical**
- **Almost unlimited storage**
- **Backup and recovery**
- **Easy access to information**
- **Quick deployment**
- **Automatic software integration**

Disadvantages

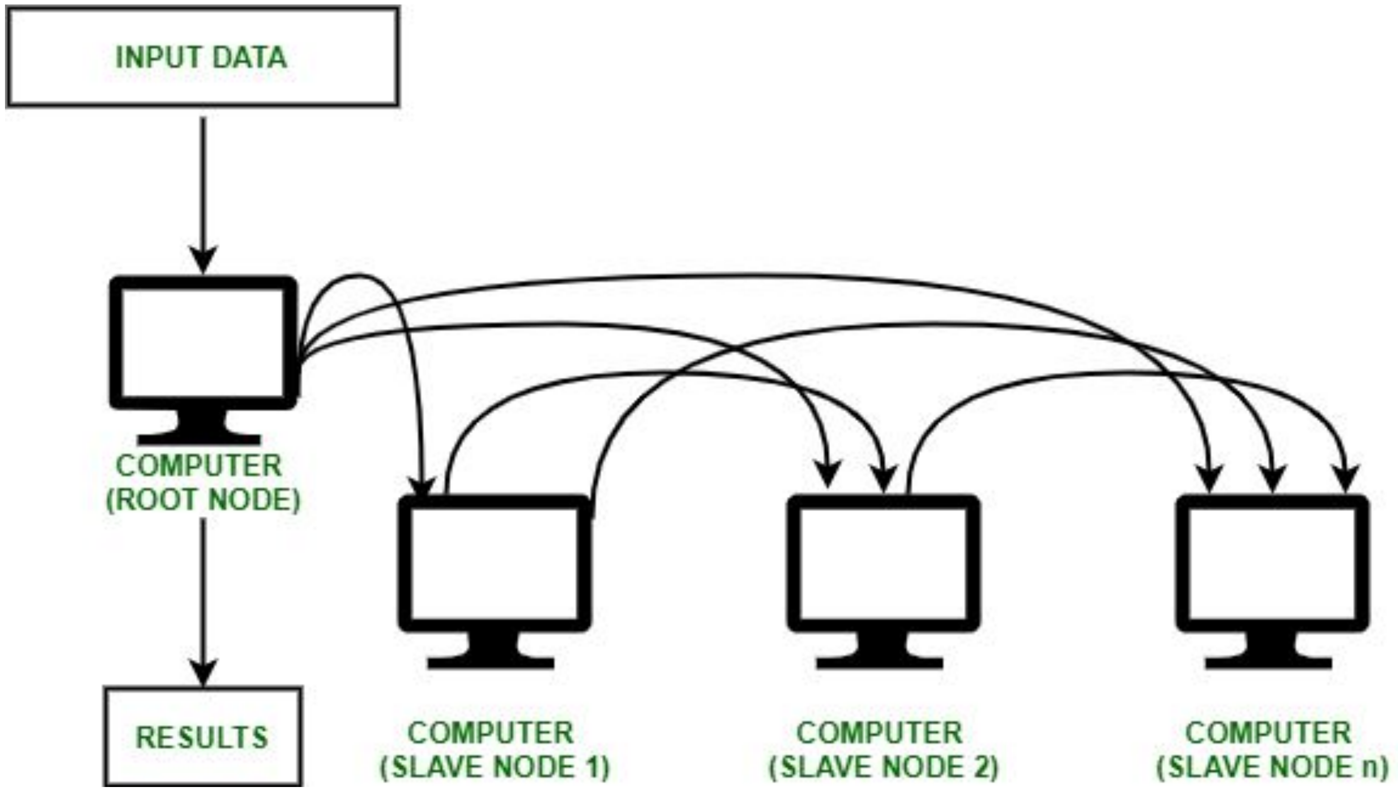
- Technical issues
- Prone to attack
- Security and privacy
- Lack of standards
- Continuously evolving
- Compliance concerns
- Service migration

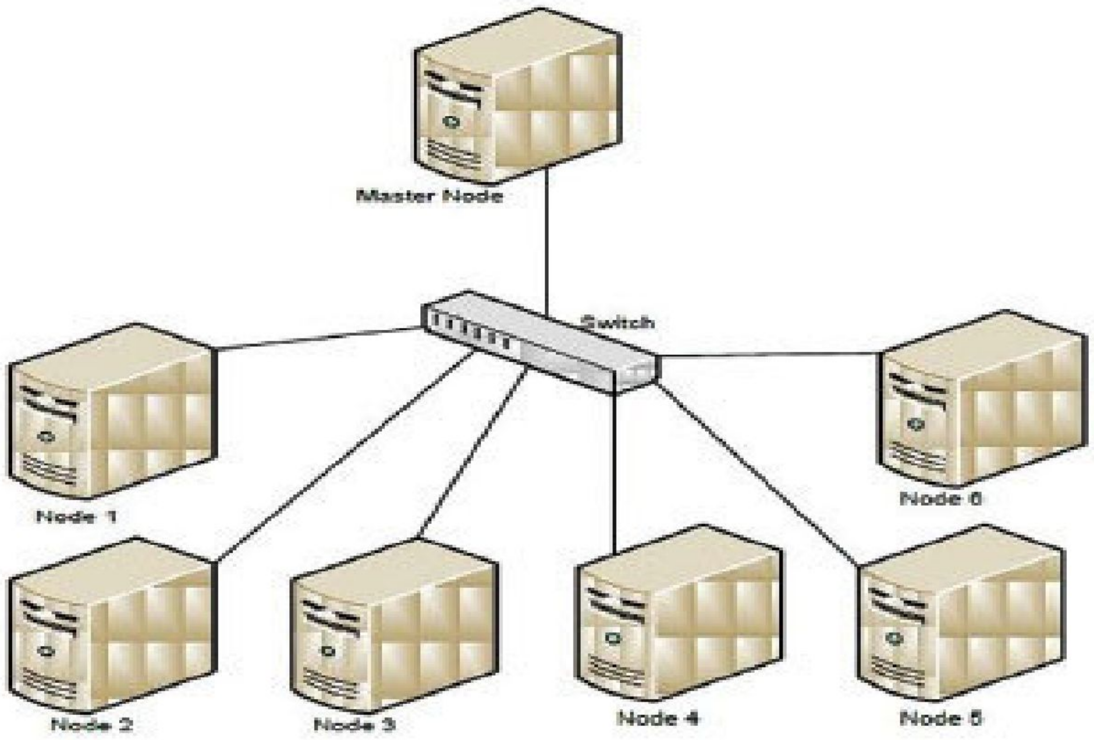
Comparison of traditional and cloud computing paradigms

- Service-oriented Architecture (SOA)
- Cluster computing
- Grid computing
- Virtualization technology

Cluster computing

- A cluster is an embodied set of stand-alone systems interconnected via a local area network or a group of linked computers, working together as a **SINGLE INTEGRATED SYSTEM FOR SCALING WORKLOADS.**
- *Performance improvement, fault tolerance, scalability, huge cost savings, throughput, redundancy, high memory, enormous speed, load balancing and high availability*


INPUT DATA
COMPUTER
(ROOT NODE)
RESULTS
COMPUTER
(SLAVE NODE 1)
COMPUTER
(SLAVE NODE 2)
COMPUTER
(SLAVE NODE n)



- All the connected computers are the **same kind of machines**
- They are **TIGHTLY CONNECTED** through dedicated network connections
- All the computers share a common home directory.

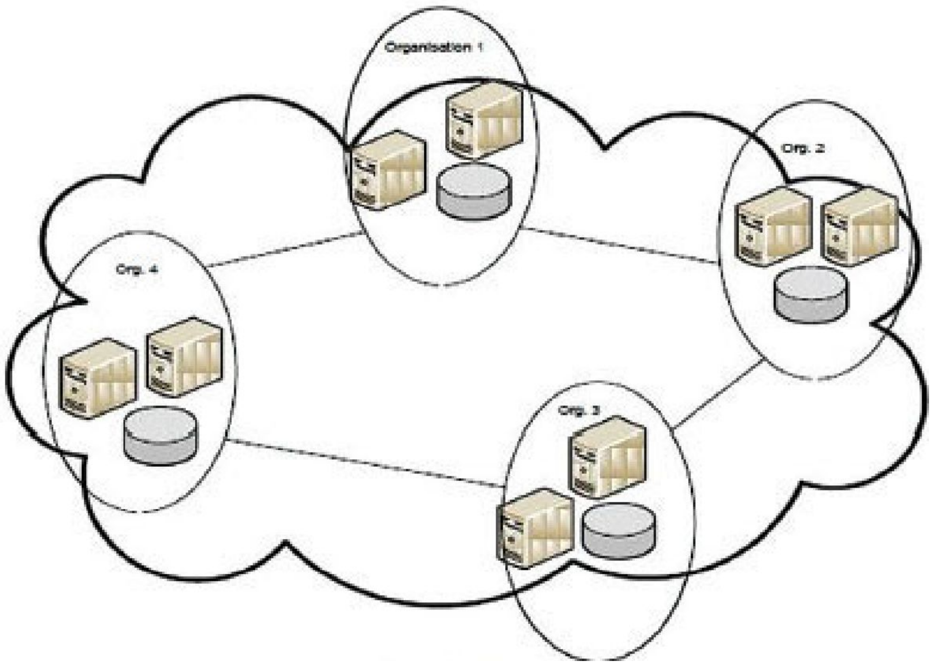
1. ***CLUSTER LOAD BALANCING***
2. ***HIGH–AVAILABILITY CLUSTERS***
3. ***HIGH-PERFORMANCE CLUSTERS***

- **Cost efficiency**
- **Processing speed**
- **Extended resource availability**
- **Expandability**
- **Flexibility**

- **Difficult to manage and organize** a large number of computers
- **Poor performance** in the case of non-parallelizable applications
- **Physical space needed** is considerably greater than that of a single server
- **Increased power consumption** compared to a single server

Grid computing

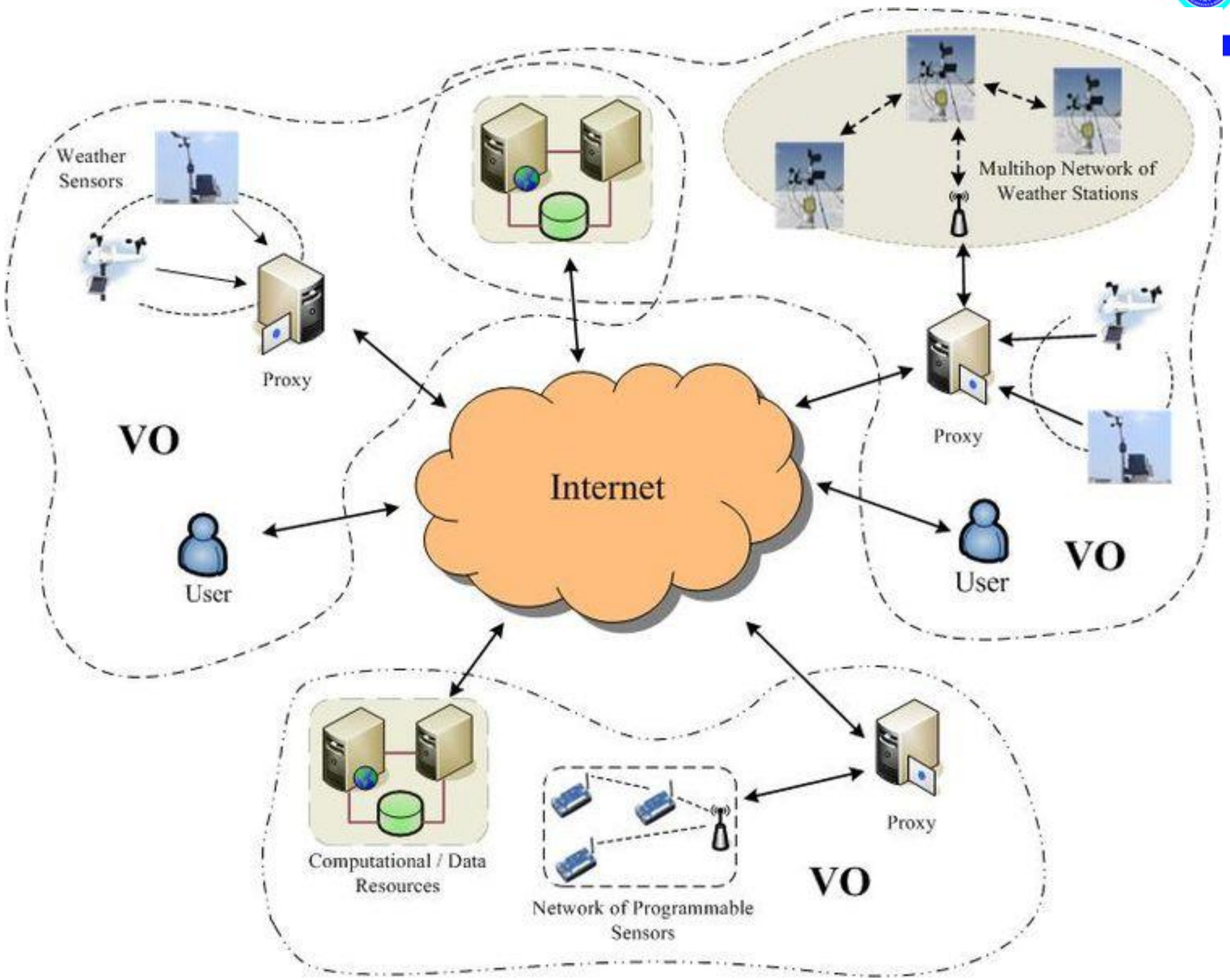
- Grid Computing can be defined as a network of **homogeneous or heterogeneous computers** working together over a long distance to perform a task that would rather be **difficult for a single machine.**
- Grid computing is defined as a **distributed architecture of multiple computers connected** by networks that work together to accomplish a joint task.
- Grid computing is the unification of several computing resources from several supervisory domains into one or more logical entity, coordinated with a high-performance distributed grid and applied to **solving large BATCH PROCESSING PROBLEMS.**
- Grid computing is also referred to as a super virtual computer.
- In grid computing, computing resources are located on **loosely-coupled but geographically dispersed, distributed and heterogeneous networks unlike in cluster computing**.



- **COMPUTATIONAL GRID COMPUTING**
- **DATA GRID COMPUTING**
- **COLLABORATIVE GRID COMPUTING**
- **MANUSCRIPT GRID COMPUTING**

- It is not centralized, as there are **no servers** required, except the **control node** which is just used for controlling and not for processing.
- Multiple heterogeneous machines
- Tasks can be performed **parallelly across various physical locations.**
- It guarantees **optimal resource balancing**

- If a **node on the grid is down, a single point of failure occurs.**
- A super fast interconnect between computer resources is the need of hour.
- **Licensing across** many servers may make it prohibitive for some applications.
- Many groups are unwilling with sharing resources .


Weather Sensors
Proxy
VO
User
Multihop Network of Weather Stations
Proxy
User
VO
Internet
Computational / Data Resources
Network of Programmable Sensors
Proxy
VO

Cloud computing

Key Points

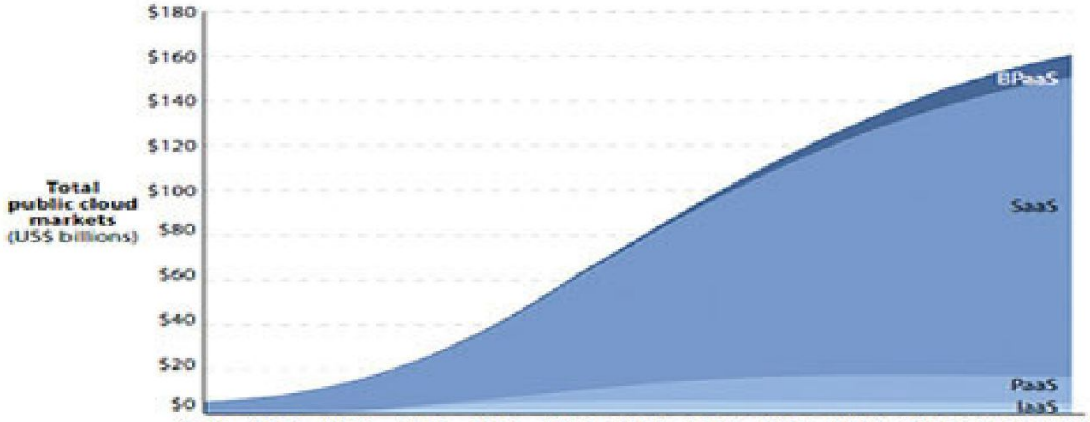
CLUSTER COMPUTING	GRID COMPUTING	CLOUD COMPUTING
Characteristics of Cluster computing **1:Tightly coupled systems** **2: Single system image** **3: Centralized Job management & scheduling system**	Characteristics of Grid Computing **1: Loosely coupled (Decentralization)** **2: Diversity and Dynamism** **3: Distributed Job Management & scheduling**	Characteristic of cloud computing **1: Dynamic computing infrastructure** **2: IT service-centric approach** **3: Self-service based usage model** **4: Minimally or self-managed platform** **5: Consumption-based billing**
In cluster computing, a **bunch of similar (or identical)** computers are hooked up locally (in the same physical location, directly connected with very high speed connections) to operate as a single computer	In grid computing, the computers do not have to be in the **same physical location and can be operated independently**. As far as other computers are concerned each computer on the grid is a distinct computer.	In cloud computing, the computers need not to be in the same physical location.
The cluster computers all have the **same hardware and OS.**	The computers that are part of a grid can run **different operating systems and have different hardware**	The **memory, storage device and network communication are managed by the operating system of the basic physical cloud units**. Open source software such as LINUX can support the basic physical unit management and virtualization computing.

Characteristics	Clusters	Grids	Clouds
Ownership	Single ownership	Multiple ownership	Single ownership
Service pricing	Limited	Private or public assigned	Utility /large user discount
Virtualization	Half	Half	Yes
Resource management	Centralized resource	Distributed resource	Both
Scalable size	100s	1000s	100 to 1000s
Standardized	Yes	Yes	No
Interoperability	Yes	Yes	Not full
Speed / Interconnected network	Dedicated high end with low latency and high bandwidth	Mostly internet with high latency and low bandwidth	Dedicated high end with low latency and high bandwidth
Self-service	No	Yes	Yes
Single system image	Yes	No	Yes/optional included
Multi-tenancy	No	Yes	Yes
Service negotiation	Limited	Yes, SLA-based	Yes, SLA-based
Membership discovery	Membership service discovery	Decentralized information services and centralized indexing	Membership service discovery
Operating system	Windows/Linux	Any standard but dominated by Unix	Uses a hypervisor
Application drivers	Business, data centres, enterprise computing	Collaborative scientific and high-throughput applications	Web App. content delivery, dynamic provisioning
Standards/ interoperability	Virtual Interface Architecture (VIA)	Some open grid forum	Web services (SOAP and REST)
Scalable	No	Half	Yes

Characteristics	Clusters	Grids	Clouds
Failure management	Limited (often failed task / application and restarted	Limited (often failed task / application restarted	Failover, content replication, virtual machine migration from one node to another supported
Capacity	Stable and guaranteed capacity	Varies, but high capacity	Provisioned on-demand capacity
Security	Traditional login / password-based	Public / private pair -based authentication and mapping of a user to an account	Each user and / or application is provided with a virtual machine
Privacy	Medium level of privacy depends on user privileges	Limited support for privacy	High security / privacy is guaranteed. There is support for file **Access Control List (ACL)** settings.
Population	Commodity computers	High-end computing systems (including clusters and servers)	Commodity PCs, high-end servers' network, attached storage
End-user presentation	Presented as a dynamic and diversified system	Presented as a single system image	Presented as a self-services-based usage model

Evaluating the cloud's business impact and economics

The spreadsheet detailing this forecast is available online.



	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
BPaaS ($)	0.15	0.23	0.35	0.53	0.80	1.26	1.95	2.93	4.28	6.00	7.66	9.08	10.02
SaaS ($)	5.56	8.09	13.40	21.21	33.09	47.22	63.19	78.43	92.75	105.49	116.39	125.52	132.57
PaaS ($)	0.05	0.12	0.31	0.82	2.08	4.38	7.39	9.80	11.26	11.94	12.15	12.10	11.91
IaaS ($)	0.06	0.24	1.02	2.94	4.99	5.75	5.89	5.82	5.65	5.45	5.23	5.01	4.78

58161

Source: Forrester Research, Inc.

Business drivers of cloud computing adoption

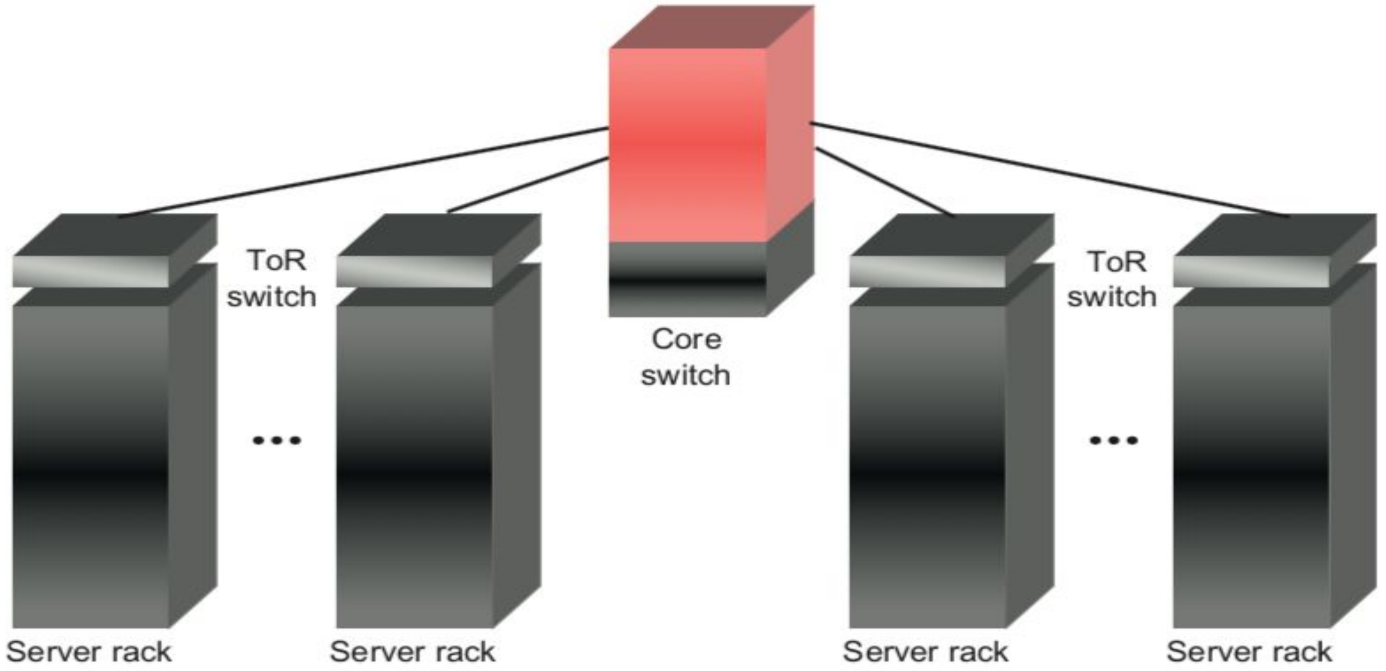
- Key business drivers of cloud computing adoption
- Help to pursue new business opportunities
- Upfront costs reduction
- Potential improvement in business continuity
- Potential reduction in carbon footprint
    - **Server costs**
    - **Storage costs**
    - **Network costs**
    - **Backup and archive costs**
    - **Disaster recovery costs**
    - **Software maintenance costs**
    - **Support personnel costs**

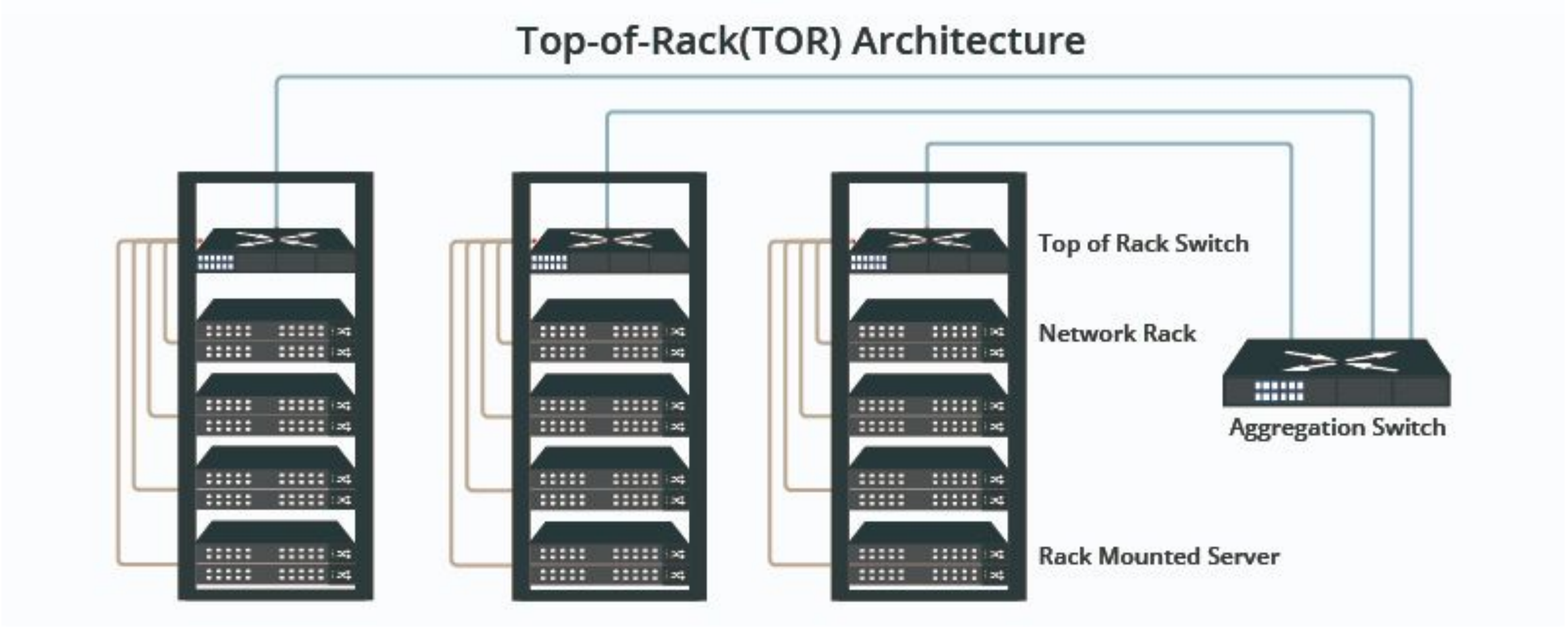
Future of the cloud (FoC)

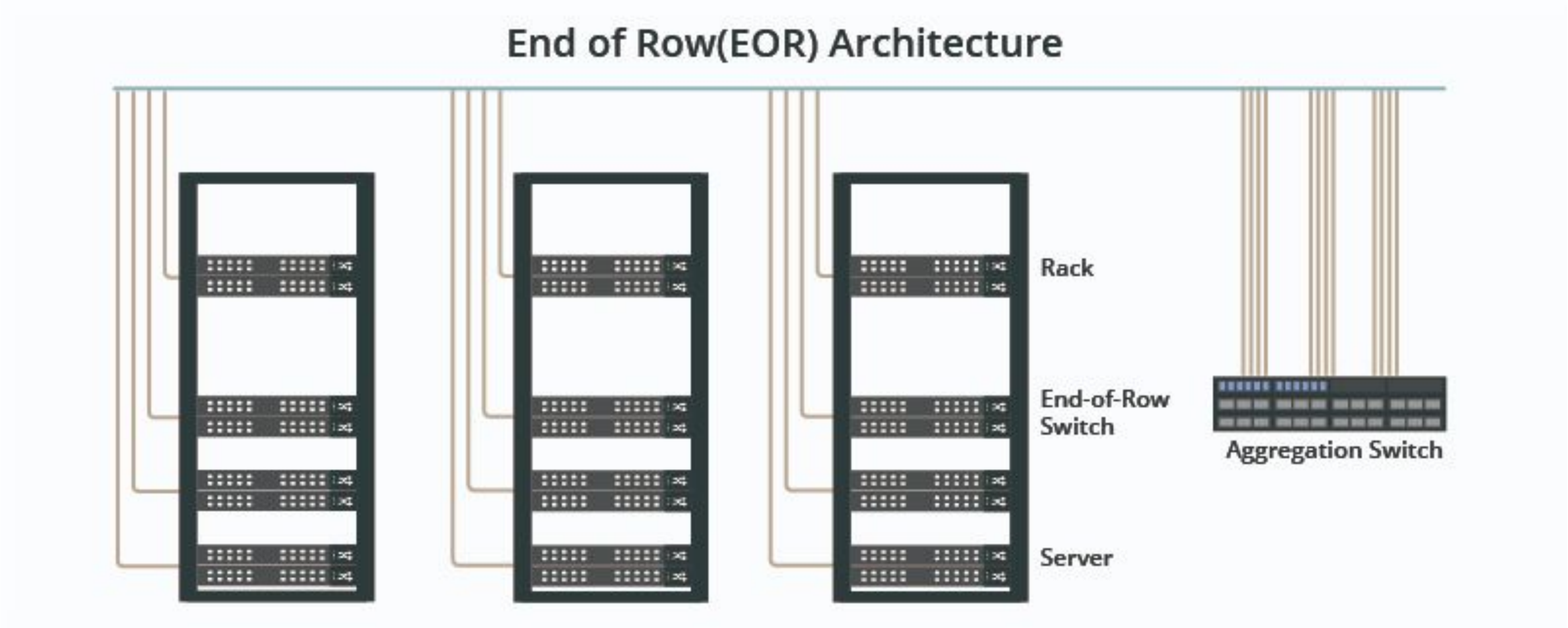
- Factors driving the future of cloud computing
- Defining and comparing services
- Enabling the next generation data centers
- Managing a hybrid world
- Everything as a service

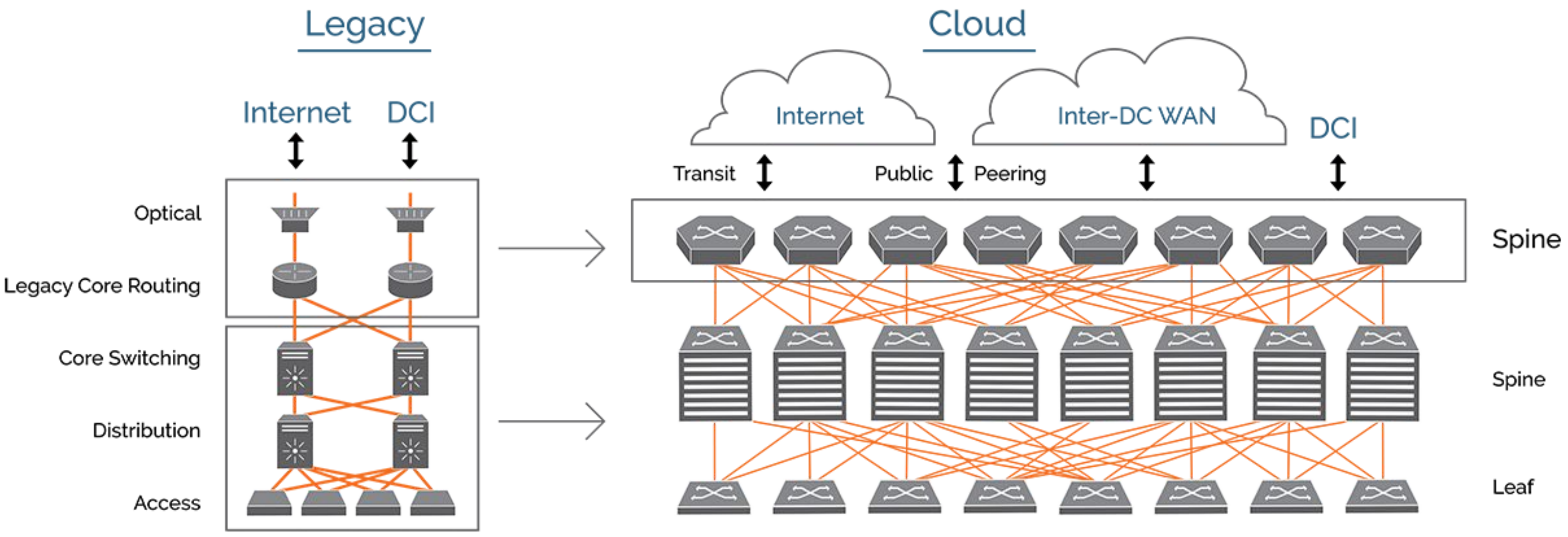
Datacenter

- The ***datacenter* is the collection** of servers where the application to which you subscribe is housed.
- It could be a large room in **the basement of your building or a room full of servers on the other side of the world that you access via the Internet.**
- A growing trend in the IT world is virtualizing servers.
- That is, software can be installed allowing multiple instances of virtual servers to be used.
- In this way, you can have half a dozen virtual servers running on one physical server.

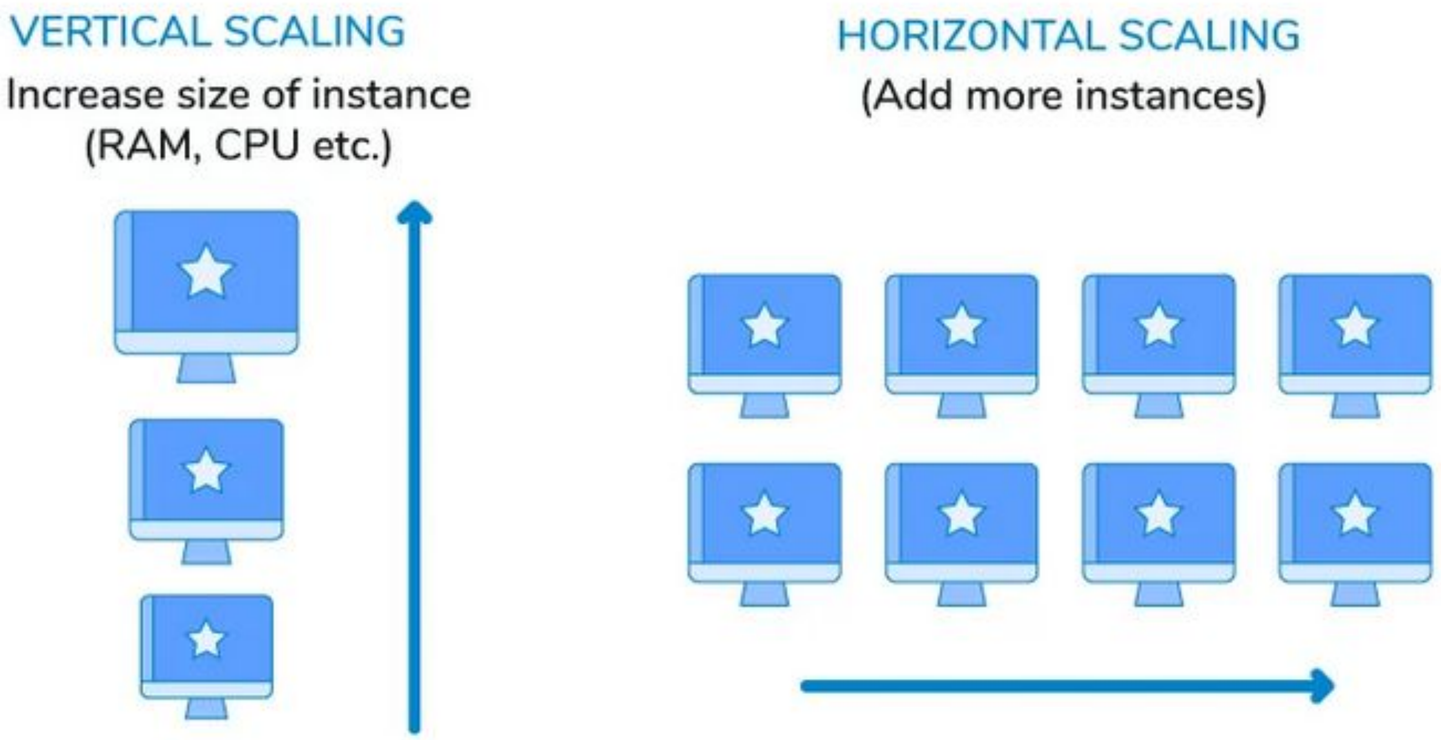



Top-of-Rack(TOR) Architecture
Top of Rack Switch
Network Rack
Aggregation Switch
Rack Mounted Server


End of Row(EOR) Architecture
Rack
End-of-Row
Switch
Aggregation Switch
Server



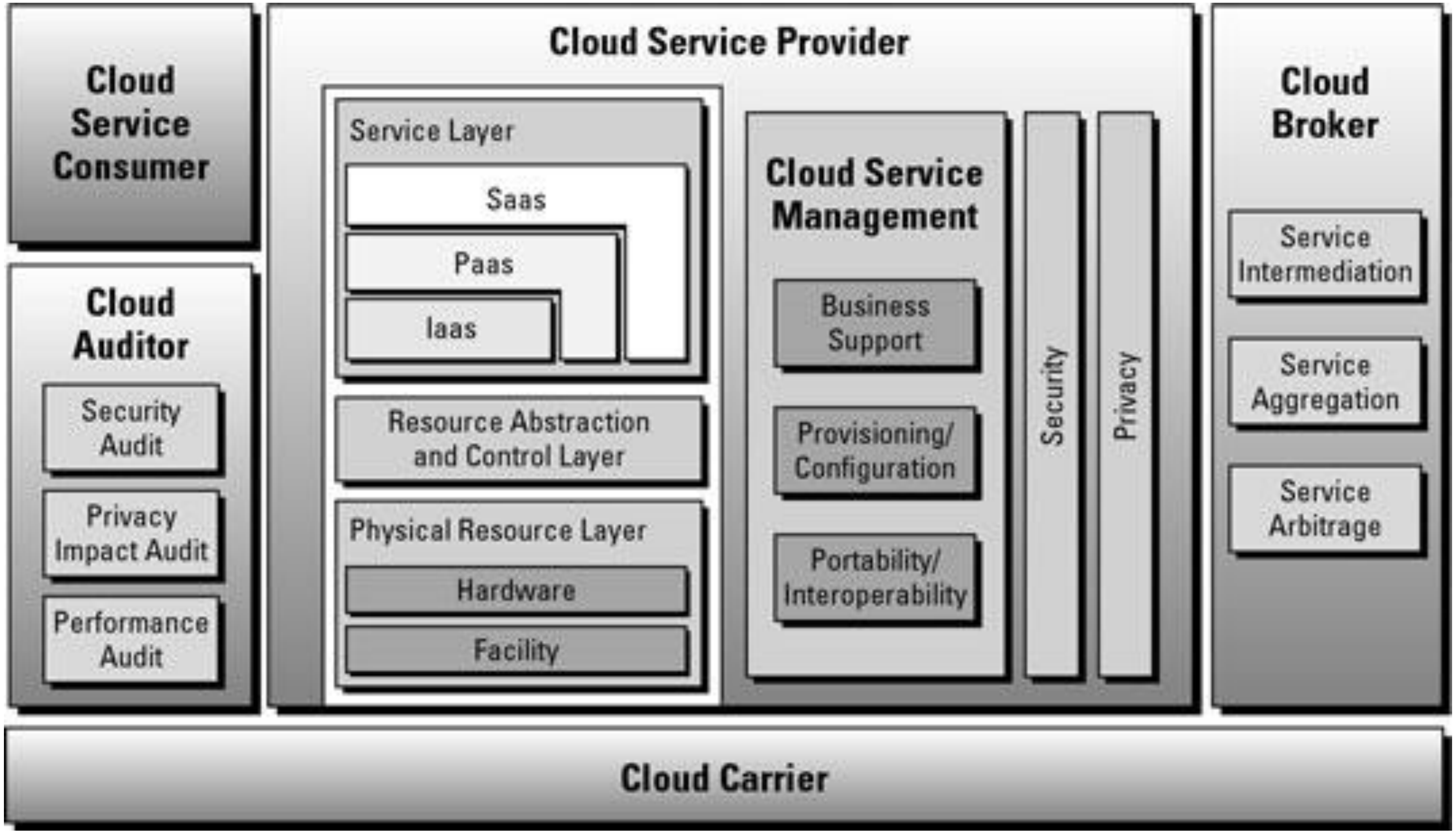
https://www.arista.com/en/products

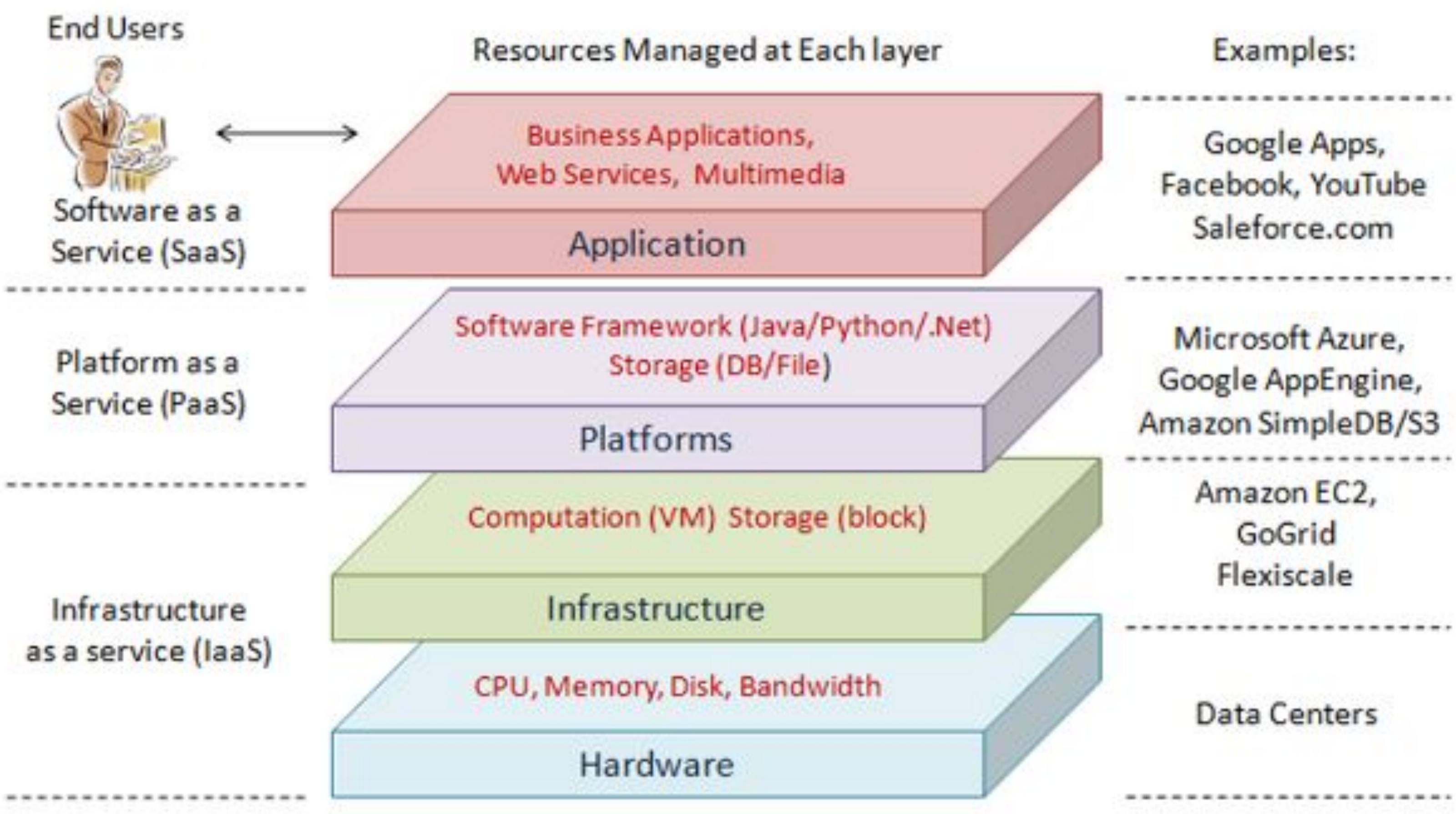

VERTICAL SCALING
Increase size of instance
(RAM, CPU etc.)
HORIZONTAL SCALING
(Add more instances)

Distributed Servers

- But the servers don't all have to be retained in the same location.
- Often, servers are in geographically disparate locations.
- But to you, the **CLOUD SUBSCRIBER**, these servers act as if they're humming away right next to each other.
- This gives the **service provider more flexibility in options and security**.
- *For instance, Amazon has their cloud solution in servers all over the world.*
- *If something were to happen at one site, causing a failure, the service would still be accessed through another site.*
- *Also, if the cloud needs more hardware, they need not throw more servers in the safe room—they can add them at another site and simply make it part of the cloud.*

Cloud Computing Ecosystem (CCE)





https://link.springer.com/content/pdf/10.1007/s13174-010-0007-6.pdf

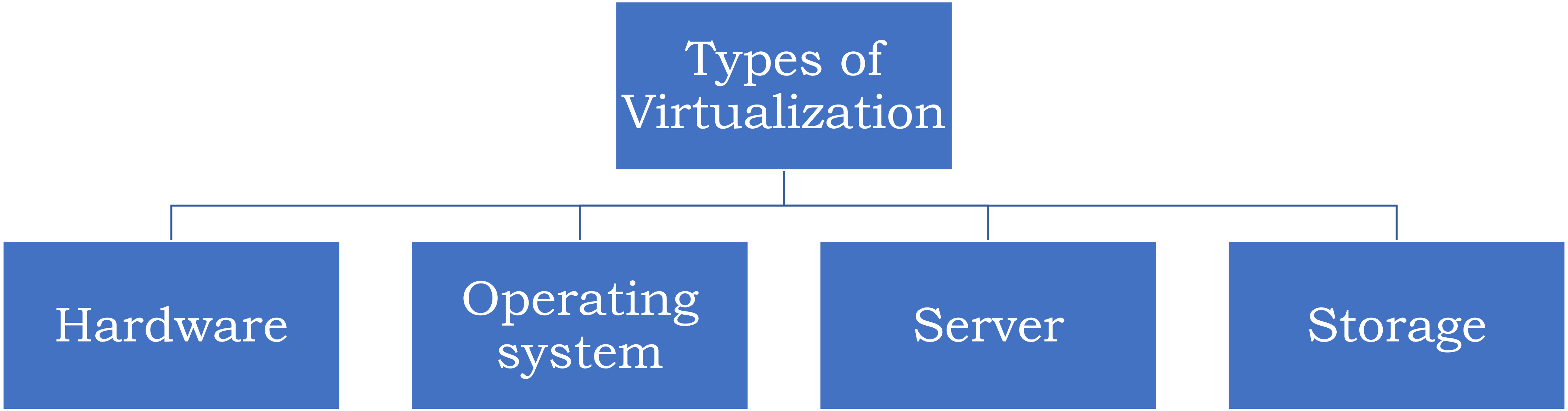
Virtualization

- **Virtualization** is the "***creation of a virtual (rather than actual) version of something,*** such as a server, a desktop, a storage device, an operating system or network resources".

- **"BEING ON OR SIMULATED ON A COMPUTER OR COMPUTER NETWORK "**

- The machine on which the **virtual machine is going to create is known as Host Machine** and that virtual machine is referred as a Guest Machine

Types of Virtualization



SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act, 1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

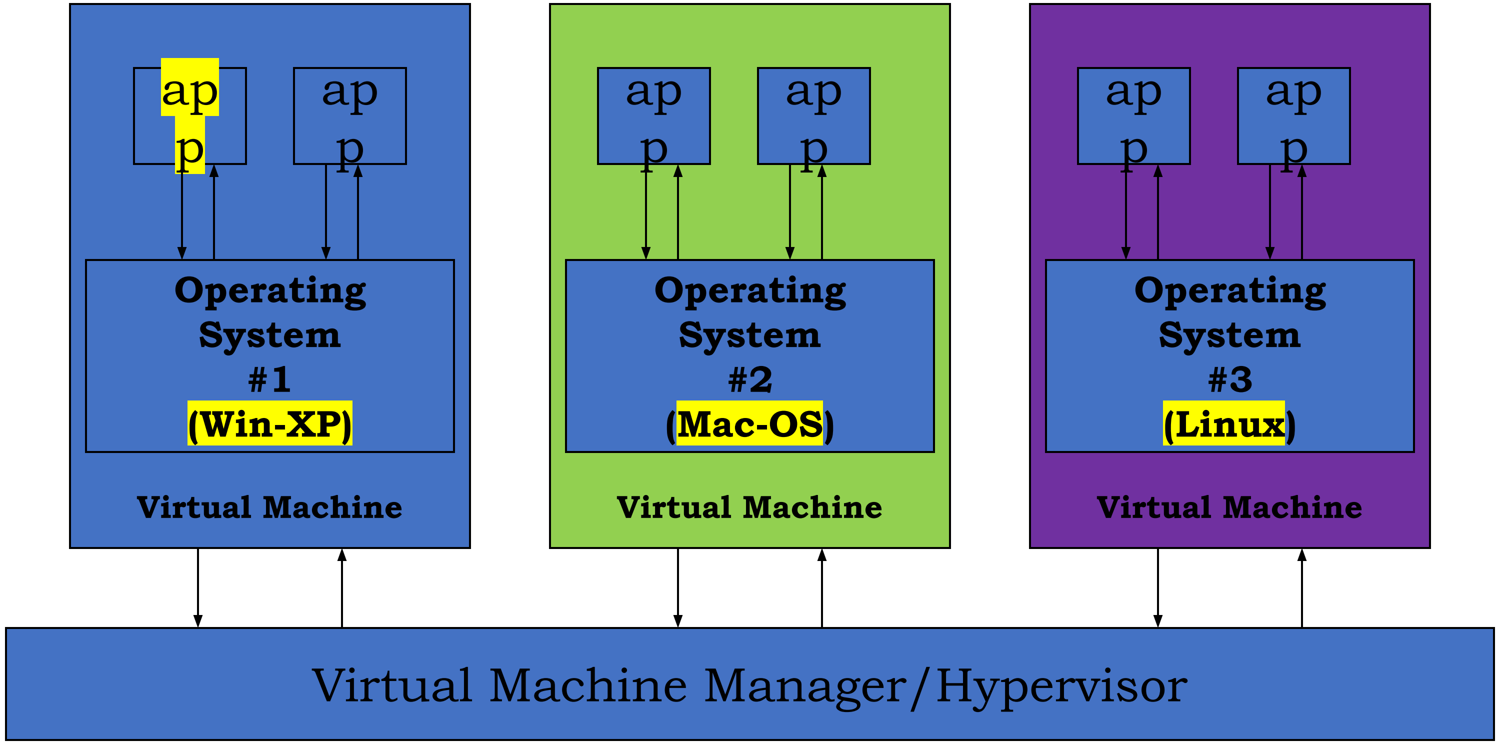
Type	Definition with technology	Types with pros and cons
Hardware Virtualization	When the virtual machine software or virtual machine manager (VMM) is **directly installed on the hardware system** is known as hardware virtualization. The main job of hypervisor is to **control and monitoring the processor, memory and other hardware** resources. After virtualization of hardware system we can install different operating system on it and run different applications on those OS.	• **Para-Virtualization** • **Full Virtualization** • **Emulation Virtualization** **Efficient Backup and Recovery** **Software Redundancy** **Economical Benefits** **Reducing the scalability and reliability of the server consolidation to the rapid increase in CPU overhead when it doesn't have proper compatibility with processor**

Type	Definition with technology	Types with pros and cons
Operating System Virtualization	When the virtual machine software or virtual machine manager *(VMM) is installed on the* ***Host operating system*** **instead of directly on the hardware system** is known as operating system virtualization Operating System Virtualization is mainly used for **testing the applications on different platforms of OS.**	• ***Linux OS Virtualization*** • ***Windows OS virtualizations*** **Saves money and time as it doesn't require more maintenance due to the non-requirement of any hardware.** The **container should have the same OS version** and the same patch level as the base OS. If the base OS crashes, all virtual containers become unavailable inaccessible

Type	Definition with technology	Types with pros and cons
Server Virtualization	When the virtual machine software or virtual machine manager *(VMM) is directly installed on the* ***Server system*** **is known as server virtualization.** Server virtualization is done because a **single physical server can be divided into multiple servers on the demand basis and for balancing the load.**	• **Para-Virtualization** • **Full Virtualization** • **Emulation Virtualization** **Server consolidation** **Reduced hardware and facilities cost** **Simplified physical infrastructure** **Risk and availability** **VM sprawl** **Resource shortages**

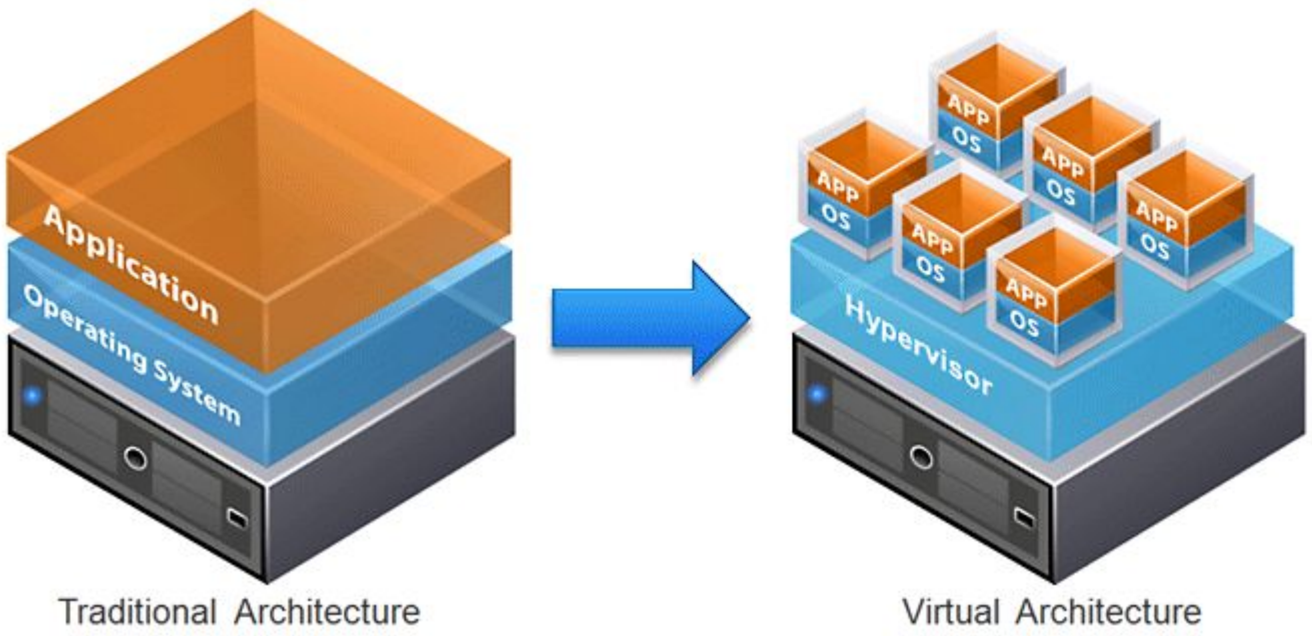
Type	Definition with technology	Types with pros and cons
Storage Virtualization	Storage virtualization is the *process of grouping the* ***physical storage from multiple network storage devices*** *so that it looks like a single storage device.* Storage virtualization is also implemented by using software applications Storage virtualization is mainly done for **back-up and recovery purposes**.	• **Block Virtualization** • **File Virtualization** **Easy Retrieval and Upload of Data** **Security**

What's VT for?

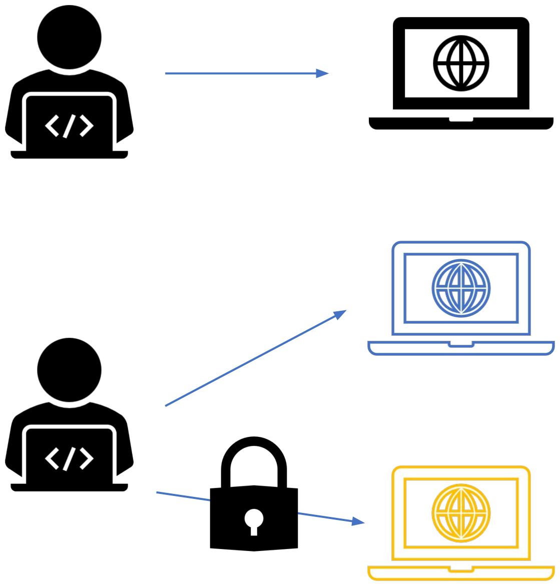


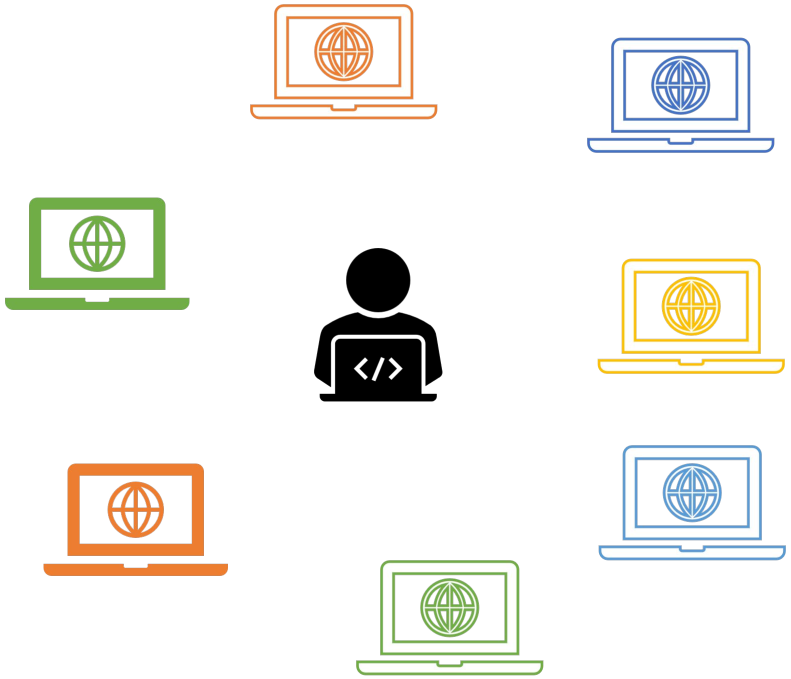
Each operating system was designed to be in total control of the system, which makes it impossible for two or more **operating systems to be executing concurrently on the same platform** – unless **'total control' is taken away from them by a new layer of control-software:** the VMM

- Creation of a virtual machine over existing operating system and hardware is known as **Hardware Virtualization.**
- A Virtual machine provides an environment that is logically separated from the underlying hardware.



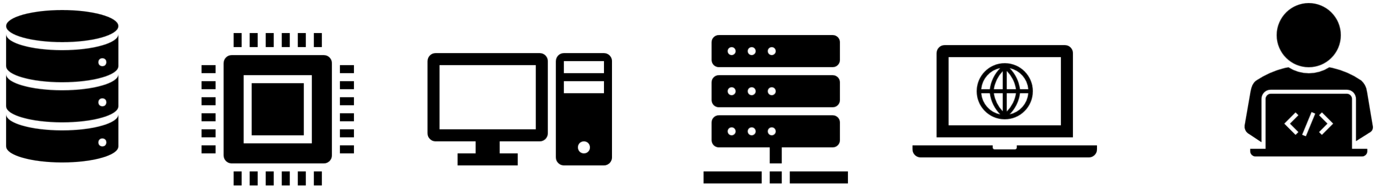
What is the use of virtualization for building a cloud?





Dia 1 ***one client using one system***
Dia 2 one client uses **one system with 2 OS** (boot in to one, that **OS will lock in the hardware and doesn't allow to share it with the second one installed on the computer**)
Dia 3 One client can have many OS by using Virtualization technology

How Virtualization is merged with cloud



companies like Amazon,Microsoft, Google and IBM (not limited to) thought they will build large ***datacenters and rent those servers, software and development platforms***.

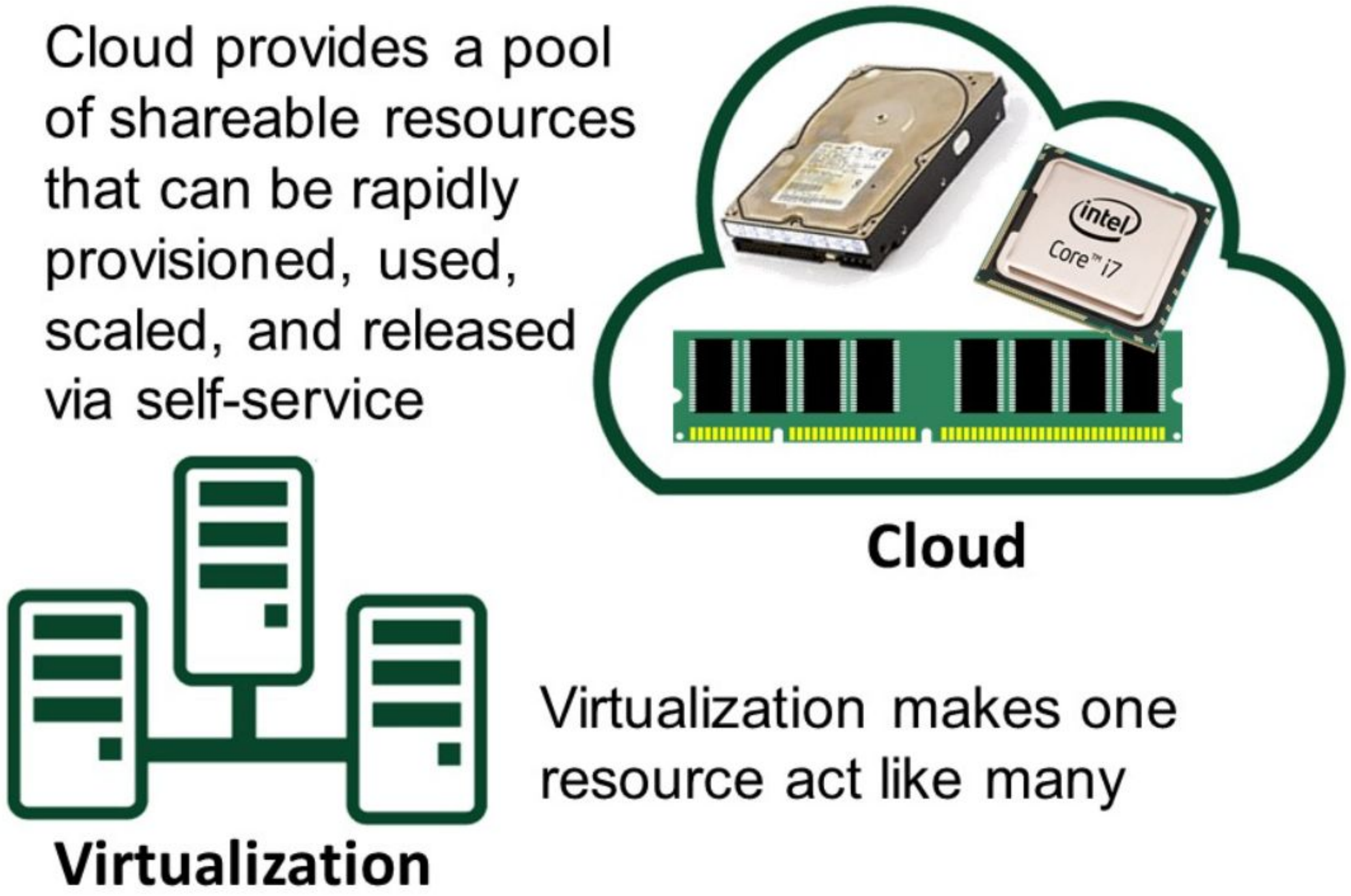
- Now say 10000 people go to a cloud provider and ask for 10000 servers with different configuration, some out of 10000 users say they will use it for an hour, some say they wanted to use it for a day, a month, a year and so on.

- So the cloud provider can't give them access to physical servers as that is not profitable and too costly.

- They use virtualization to build 10000 virtual machines on few (3–4 or 5) servers and give away those virtual machines to the users asking for.

- **THOSE ARE CALLED CLOUD INSTANCES**

Cloud Computing Vs Virtualization

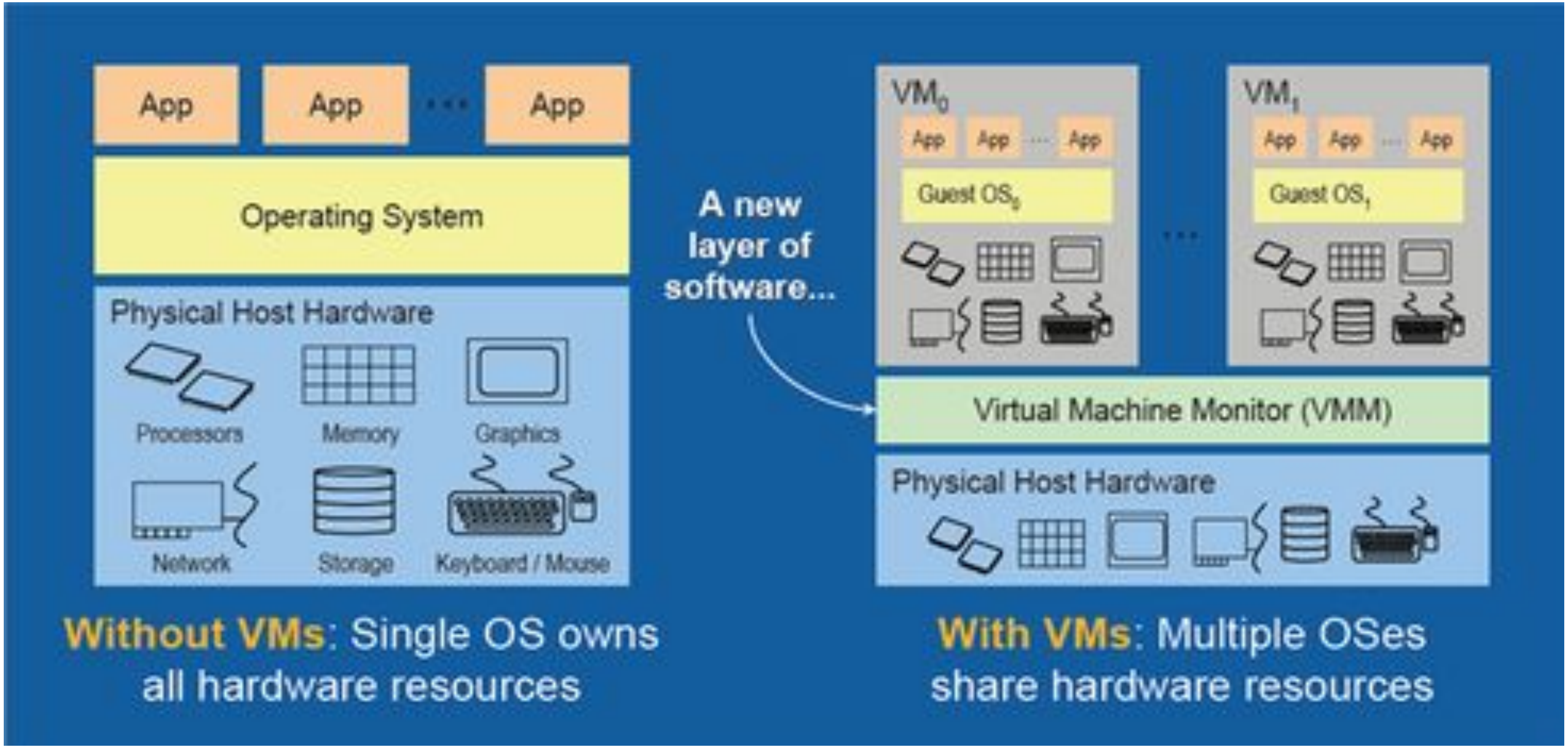


- Virtualization is the fundamental technology that powers cloud computing.
- Virtualization is software that manipulates hardware, while cloud computing refers to a **service that results from that manipulation.**
- Virtualization is a foundational element of cloud computing and **helps deliver on the value of cloud computing**

CLOUD COMPUTING	VIRTUALIZATION
Cloud computing is used to **PROVIDE POOLS AND AUTOMATED RESOURCES THAT CAN BE ACCESSED ON-DEMAND**.	While It is used to make various simulated environments through a **PHYSICAL HARDWARE SYSTEM.**
Cloud computing **SETUP IS TEDIOUS, COMPLICATED.**	While **VIRTUALIZATION SETUP IS SIMPLE** as compared to cloud computing.
Cloud computing is **high scalable and very flexible**	While virtualization is **LOW SCALABLE AND LESS FLEXIBLE** compared to cloud computing.
In the condition of disaster recovery, cloud computing **RELIES ON MULTIPLE MACHINES**.	While it relies on single **PERIPHERAL DEVICE**.
In cloud computing, the **WORKLOAD IS STATELESS**.	In virtualization, the **WORKLOAD IS STATEFUL**.
The total cost of cloud computing is **HIGHER THAN VIRTUALIZATION.**	The total cost of virtualization is **LOWER THAN CLOUD COMPUTING**.
Cloud computing requires **MANY DEDICATED HARDWARE**.	While **SINGLE DEDICATED HARDWARE** can do a great job in it.
Cloud computing provides **UNLIMITED STORAGE SPACE.**	While storage space depends on **PHYSICAL SERVER CAPACITY IN VIRTUALIZATION.**
In Cloud Computing, Configuration is **IMAGE BASED.**	In Virtualization, Configuration is **TEMPLATE BASED**.
In cloud computing, we utilize the **ENTIRE SERVER CAPACITY AND THE ENTIRE SERVERS ARE CONSOLIDATED.**	In Virtualization, **THE ENTIRE SERVERS ARE ON-DEMAND**.
In cloud computing, **THE PRICING PAY AS YOU GO MODEL, AND CONSUMPTION IS THE METRIC ON WHICH BILLING IS DONE**.	In Virtualization, the pricing is totally dependent **ON INFRASTRUCTURE COSTS.**

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT I THINK TRANSPARENCY I THINK SASTRA

KEY POINTS	CLOUD COMPUTING	VIRTUALIZATION
SCALABILITY	Cloud can be extended as much as you want.	Virtual machine configuration limits its scalability.
QUICK SETUP	Setting up the cloud is a very tedious task.	It is very simple to set up a virtual environment.
FLEXIBILITY	It is very flexible for user access. A user can access its cloud from any location with internet (depending upon permission).	Proper authentication is required before accessing the virtual machines.
SERVICE TYPE	IaaS	SaaS
DEDICATED HARDWARE	Multiple hardware creates a cloud computing	Dedicated hardware required for multiple virtual machines
INTEGRATION	Cloud integration allows future expansion of Users, applications, etc.	Virtualization integration allows the expansion of new machines within the same infrastructure.
DEPENDENCY	Multiple users can access the network using the same link.	Multiple OS can be installed on a single server/computer
ACCESSIBILITY	It can be accessed from all over the world. (Internet-based cloud)	Proper permissions are required for accessing from outside the network.
DISASTER RECOVERY	Not depend upon one machine.	Single machine failure can bring done multiple virtual machines.
TYPES	Private Cloud and Public Cloud	Hardware virtualization and Application virtualization.


App
App
App
Operating System
Physical Host Hardware
Processors
Memory
Graphics
Network
Storage
Keyboard / Mouse
A new layer of software...
VM0
App
App
App
Guest OS0
VM1
App
App
App
Guest OS1
Virtual Machine Monitor (VMM)
Physical Host Hardware
Without VMs: Single OS owns all hardware resources
With VMs: Multiple OSes share hardware resources

Related Technologies

- Server Virtualization
  - *Hypervisor-based Virtualization*
  - *Techniques for Hypervisors*
  - *Hardware Support for Virtualization*
- Two Popular Hypervisors
  - **VMware Virtualization Software**
  - **XenServer Virtual Machine Monitor**
- Storage Virtualization
  - **File Virtualization**
  - **Block Virtualization**

Cloud Deployment Model

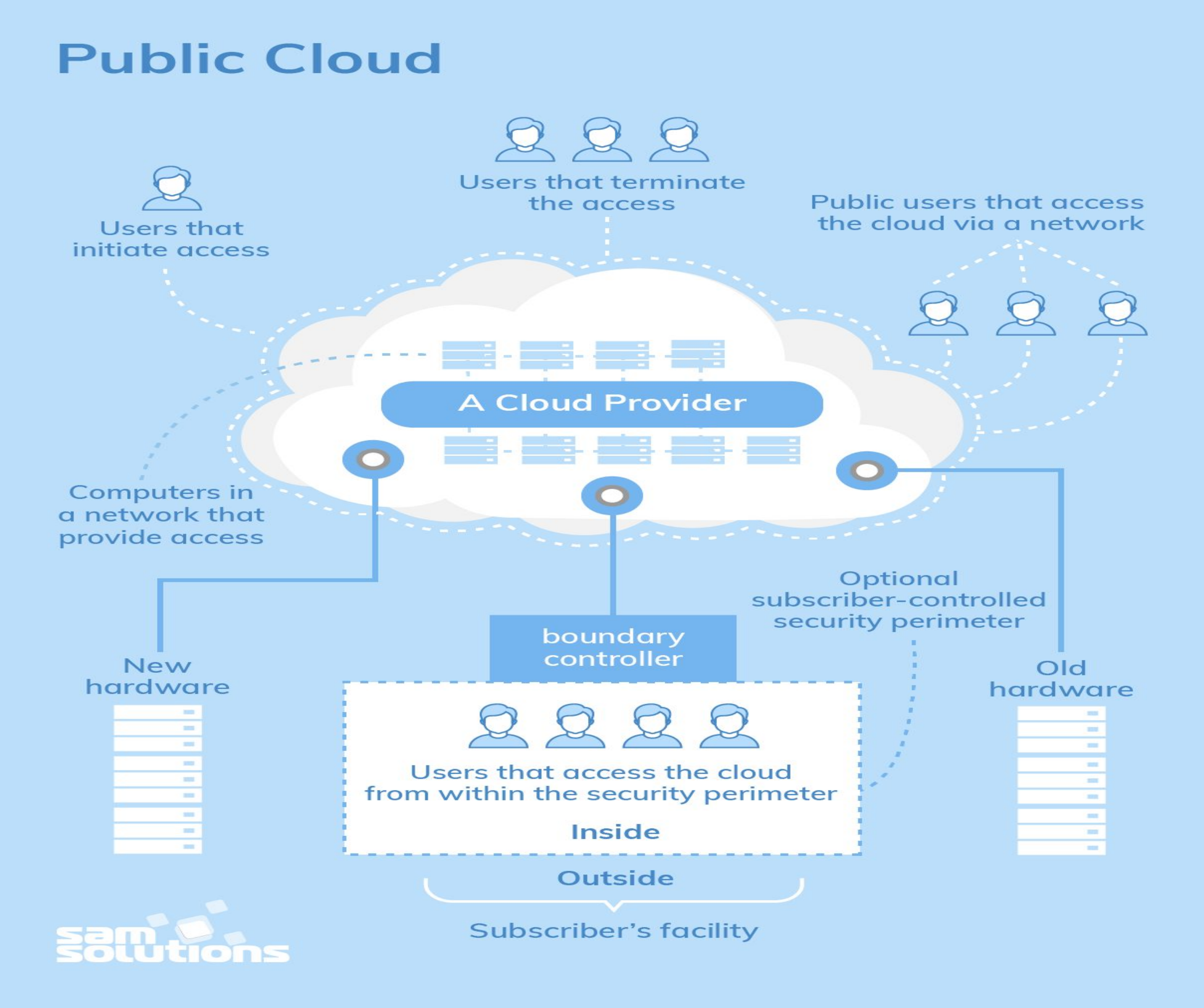
DIFFERENCE	PRIVATE	PUBLIC	HYBRID
Tenancy	Single tenancy: there's only the data of a single organization stored in the cloud.	Multi-tenancy: the data of multiple organizations in stored in a shared environment.	The data stored in the public cloud is usually multi-tenant, which means the data from multiple organizations is stored in a shared environment. The data stored in private cloud is kept private by the organization.
Exposed to the Public	**No:** only the organization itself can use the private cloud services.	**Yes:** anyone can use the public cloud services.	The services running on a private cloud can be accessed only the organization's users, while the services running on public cloud can be accessed by anyone.
Data Center Location	Inside the organization's network.	Anywhere on the Internet where the cloud service provider's services are located.	Inside the organization's network for private cloud services as well as anywhere on the Internet for public cloud services.

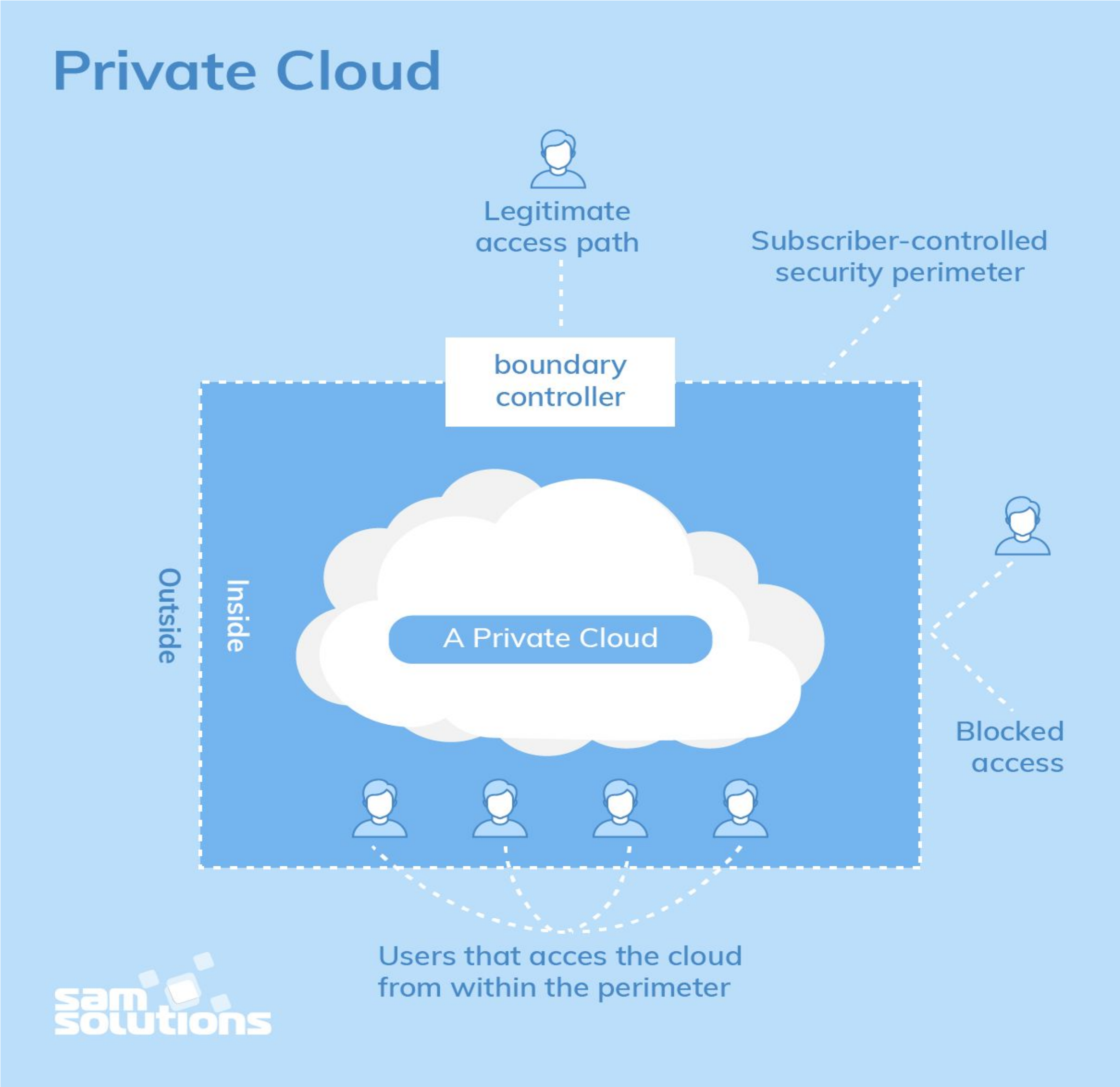
Cloud Deployment Model

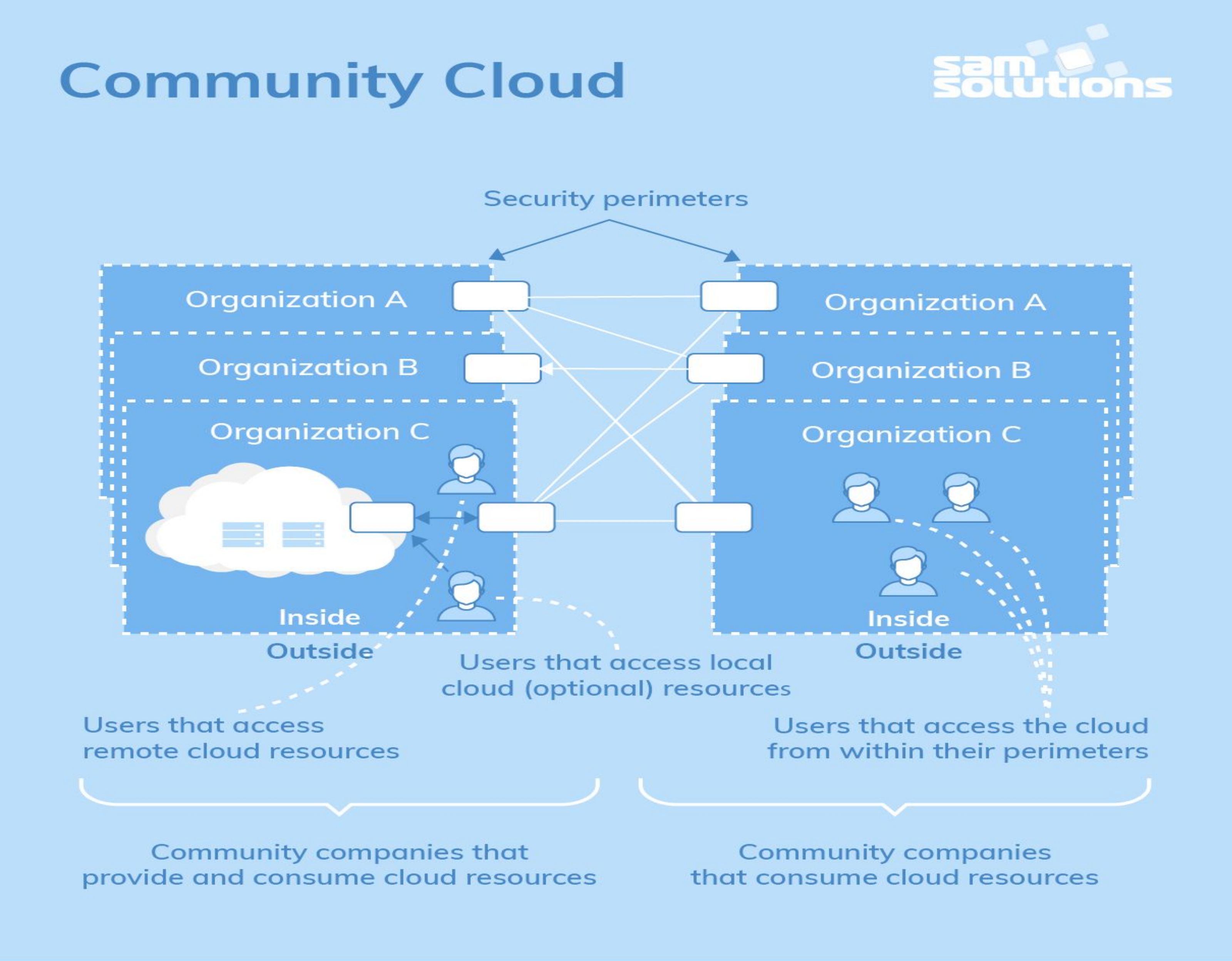
Difference	Private	Public	Hybrid
Cloud Service Management	The organization must have their own administrators managing their private cloud services.	The cloud service provider manages the services, where the organization merely uses them.	The organization itself must **manage the private cloud, while the public cloud is managed by the CSP**.
Hardware Components	Must be provided by the organization itself, which has to buy physical servers to build the private cloud on.	The CSP provides all the hardware and ensures it's working at all times.	The organization must provide **hardware for the private cloud, while the hardware of CSP is used for public cloud services**.
Expenses	Can be quite expensive, since the hardware, applications and network have to be provided and managed by the organization itself.	The CSP has to provide the **hardware, set-up the application** and provide the network accessibility according to the SLA.	The private cloud services must be provided by the organization, including the **hardware, applications and network, while the CSP manages the public cloud services.**

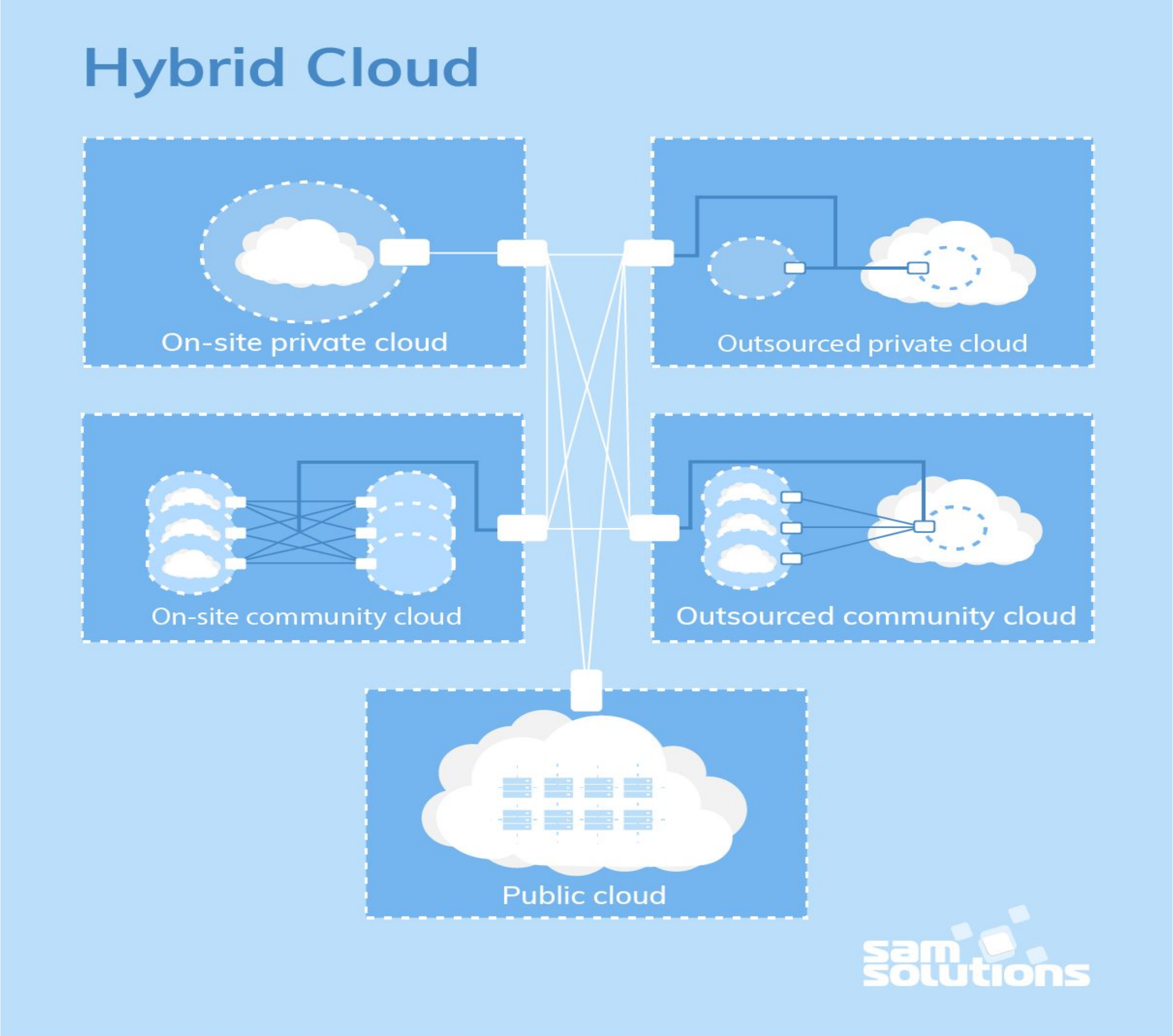
Cloud Deployment Model

	Pros	Cons	Real time Services Providers
Private	• Control over **how a cloud is setup and run** • Control over **privacy and security practices** • Control over the geographical location of data	• **Much higher setup** and maintenance costs • **Less redundancy** and resilience (especially if you host the cloud internally) • **Less scalability**	Hewlett Packard Enterprises, VMware, Dell, Oracle, IBM, Microsoft, and Amazon Web Services.
Public	• **Low price** (sometimes even free) • **Scalable** • Location independence It‘s easy!	• **Security concerns** • The law and location of your data • **Lack of control**	Amazon Web Services Microsoft Azure IBM Cloud Google Cloud Platform
Hybrid	• Keep **sensitive data safe** • Still, get some of the **scalability and cost-effectiveness of public cloud** • Ultimate flexibility	• **Complexity** • Difficulty communicating between cloud models • **More expensive than public or community models**	Microsoft, VMware, Amazon Web Services, Rackspace, Hewlett-Packard, IBM, Cisco, and Dell.


Public Cloud
Users that terminate the access
Users that initiate access
Public users that access the cloud via a network
A Cloud Provider
Computers in a network that provide access
Optional subscriber-controlled security perimeter
boundary controller
New hardware
Old hardware
Users that access the cloud from within the security perimeter
Inside
Outside
Subscriber's facility
sam solutions


Private Cloud
Legitimate access path
Subscriber-controlled security perimeter
boundary controller
Outside
Inside
A Private Cloud
Blocked access
Users that acces the cloud from within the perimeter
sam solutions


Community Cloud
sam solutions
Security perimeters
Organization A
Organization B
Organization C
Inside
Outside
Organization A
Organization B
Organization C
Inside
Outside
Users that access local
cloud (optional) resources
Users that access
remote cloud resources
Users that access the cloud
from within their perimeters
Community companies that
provide and consume cloud resources
Community companies
that consume cloud resources



https://www.sam-solutions.com/blog/four-best-cloud-deployment-models-you-need-to-know/

Some Real time deployment models

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

PUBLIC CLOUD

- **Netflix**
- *Netflix used relational databases in its data centers.*
- Netflix migrated its functioning, content, and delivery network to **Amazon's public cloud — AWS**

PRIVATE CLOUD

- **State Bank of India (SBI)** had to up its game to establish relevance with the digitally aware
- "**MeghDoot**," of about 7500 VMs hosting several financial services applications based on various technologies.
- Meghdoot offers features such as **platform and Infrastructure-as-a-service (PaaS and IaaS), metering and monitoring, Web-based management of cloud resources and enhanced security across layers.**

HYBRID CLOUD

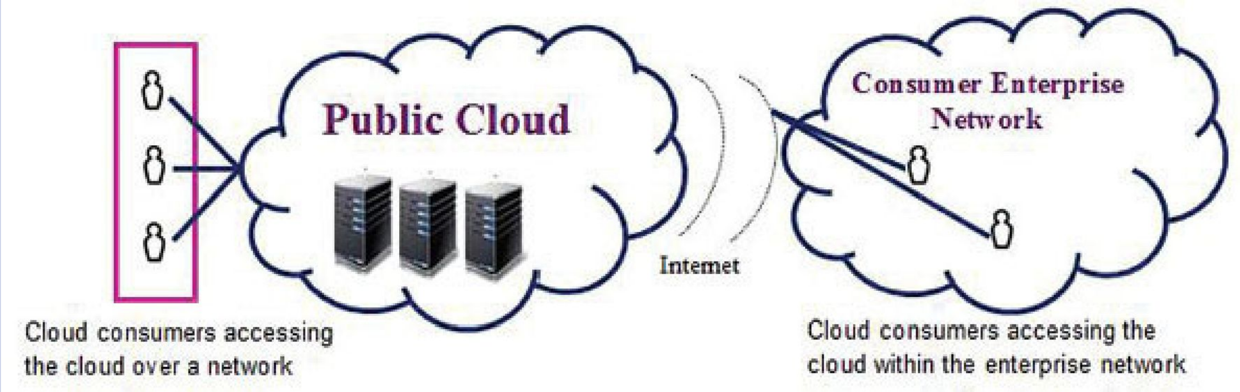
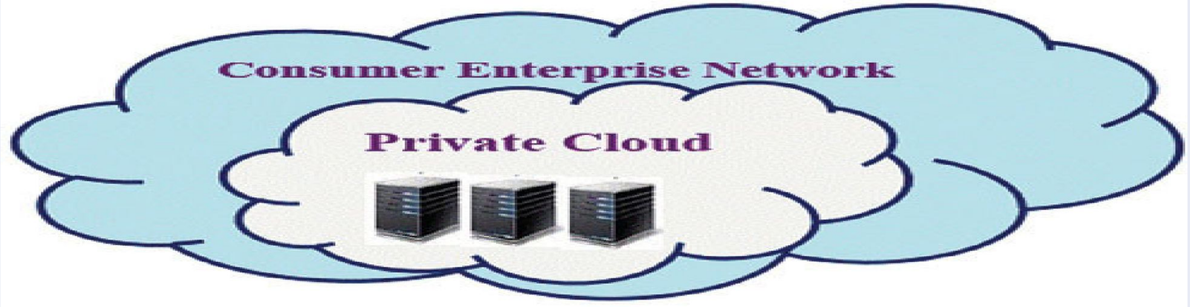
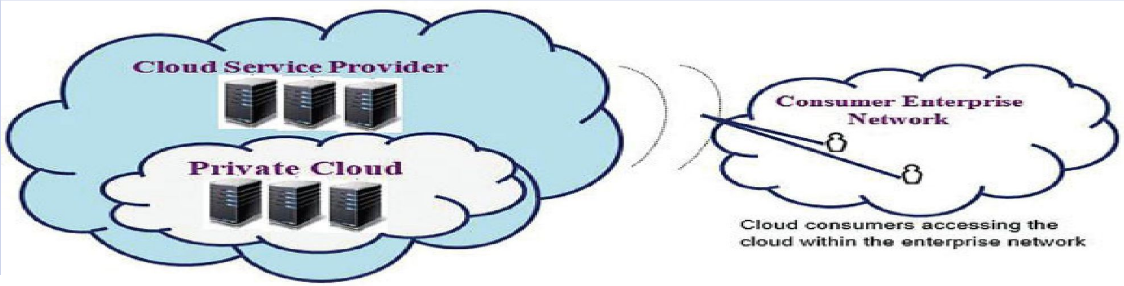
- *BBC has recently signed a 5-year hybrid cloud contract with Object Matrix after the success of the Planet series.*
- *It has been a customer of Object Matrix since 2010 with a* ***MatrixStore*** *on-premise 200TB. After the recent shift of its headquarters, BBC has seen a spike in production.*
- MatrixStore Cloud is a **storage platform that enables creative and production teams** to self-serve access to content from work or remotely from anywhere

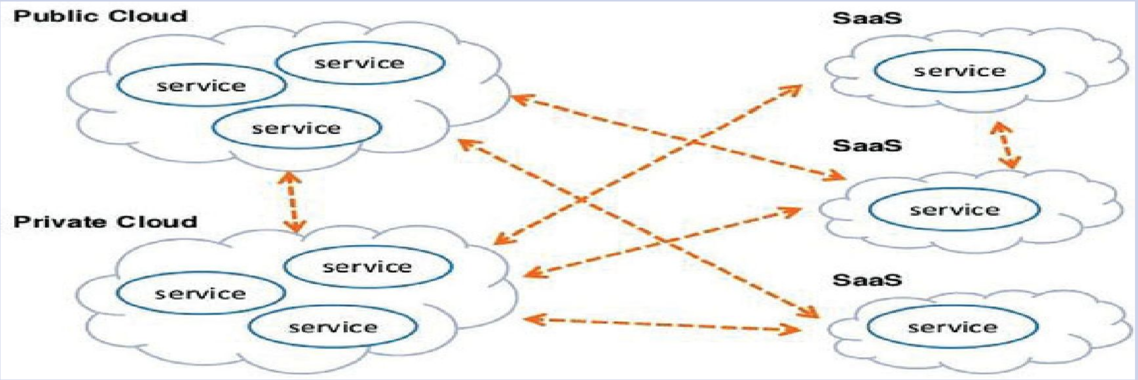
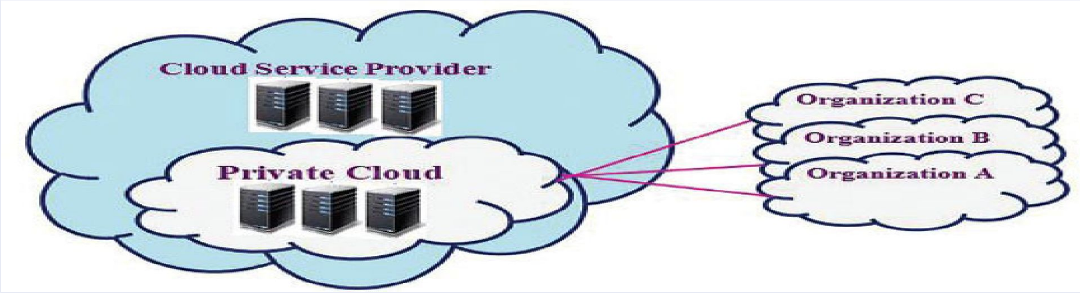
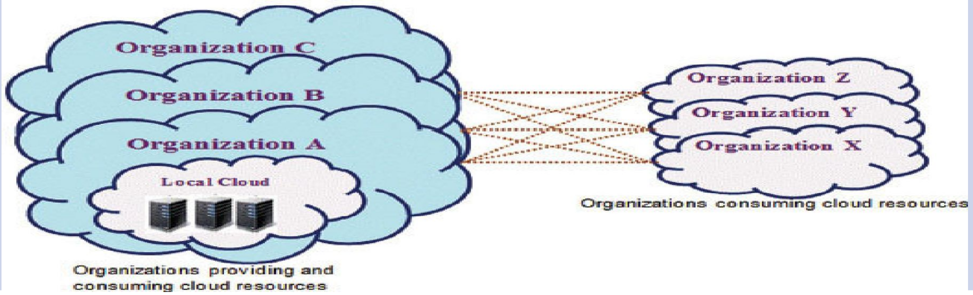
COMMUNITY CLOUD

- CoreHR is using Community Cloud for to assist the support efforts for their customers
- CoreHR required a seamless digital experience for their customers that matched the look and feel of their **company's branding.**
- **Salesforce Sales Cloud Solution**

How to choose between deployment models

- **Scalability** – Is your user activity growing quickly or unpredictable with spikes in demand?
- **Privacy and security** – Do you have any sensitive data that doesn‘t belong on a public server?
- **Ease of use** – How much time and money do you have to invest in learning and training?
- **Pricing model** – What‘s your monthly subscription budget? How much capital can you spend upfront?
- **Flexibility** – How flexible/rigid are your computing, processing, and storage needs?
- **Legal compliance** – Are there any relevant laws in your country or industry?

Cloud	Design
Structure of a public cloud	
On-premise private cloud	
Out-sourced private cloud	

Cloud	Design
Hybrid cloud	
Out-sourced community cloud	
On-site community cloud	

Attribute	Public	Private	Community	Hybrid
Ease of setup and use	Easy	Requires proficiency in IT	Requires proficiency in IT	Requires proficiency in IT
Data privacy and security	Low	High	Relatively high	High
Data control	Little to none	High	Relatively high	Relatively high
Reliability	Vulnerable	High	Relatively high	High
Scalability and flexibility	High	High	Fixed capacity	High
Cost-effectiveness	The most cost-effective	Cost-intensive, the most expensive one	Cost is shared among the community members	Cheaper than a private model but more expensive than a public model
Need for in-house hardware	No	Depends	Depends	Depends
Upfront costs	Low	High	Medium	Medium

Attribute	Public	Private	Community	Hybrid
Ongoing costs	High	Low	Medium	Medium
Security	Low	High	Medium	Medium
Compliance	Low	High	Medium	Medium
Quality of service	Low	High	Medium	Medium
Integration	Low	High	Medium	Medium
Configurability	Low	Medium	Medium	Medium

Services

- The term services in cloud computing is the concept of **being able to use reusable**, fine grained components across a vendor's network. This is widely known as "as a service."
- Offerings with *as a service* as a suffix include traits like the following:
  - Low barriers to entry, making them **available to small businesses**
  - **Large scalability**
  - **Multitenancy**, which allows resources to be shared by many users
  - Device independence, which **allows users to access the systems on different hardware**

ON-PREMISES

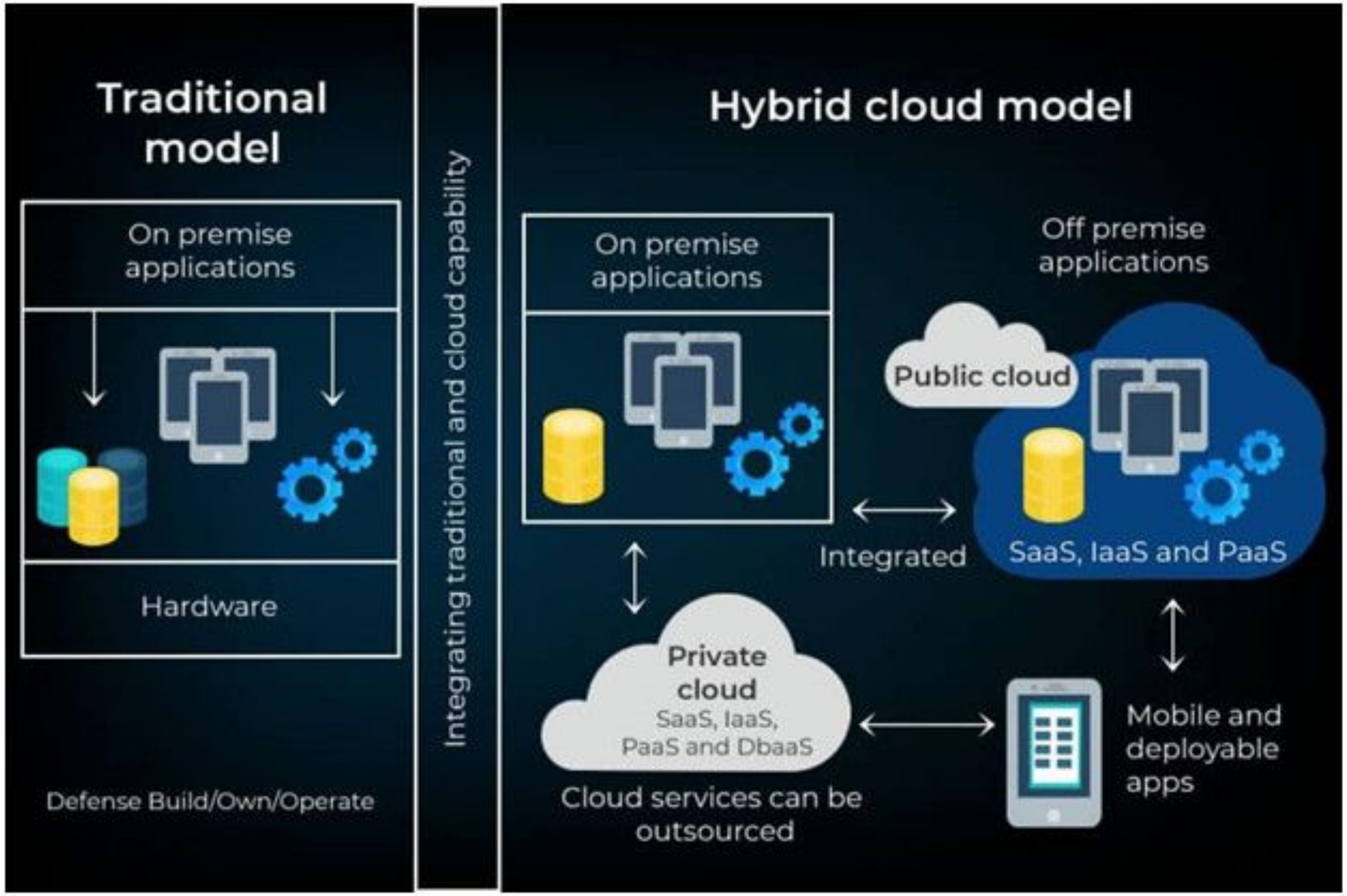
- An on-premises data center **is a group of servers that you privately own and control.**
- Traditional cloud computing (**as opposed to hybrid or private cloud computing models**) involves leasing data center resources from a third-party service provider.

Key Differences	On-Premise	Cloud
Deployment	• In an on-premises environment, **resources are deployed IN-HOUSE and within an enterprise's IT infrastructure.** • An enterprise is responsible for **maintaining the solution and all its related processes**	• While there are different **forms of cloud computing (such as public cloud, private cloud, and a hybrid cloud), in a public cloud computing environment**, resources are hosted on the premises of the service provider but enterprises are able to access those resources and use as much as they want at any given time.

Key Differences	On-Premise	Cloud
Cost	• For enterprises that deploy software on premise, they are responsible for the **ONGOING COSTS of the server hardware, power consumption, and space**	• Enterprises that elect to use a **cloud computing model** only need to pay for the resources that they use, with none of the maintenance and upkeep costs, and the price adjusts up or down depending on **HOW MUCH IS CONSUMED**.

Key Differences	On-Premise	Cloud
Control	• In an on-premises environment, enterprises retain all their data and **are FULLY IN CONTROL of what happens to it, for better or worse**. • Companies in highly regulated industries with **extra privacy concerns are more likely to hesitate to leap into the cloud before others because of this reason.**	• In a cloud computing environment, the question of **ownership of data is one that many companies** – and vendors for that matter, have struggled with. • **Data and encryption keys reside within your third-party provider**, so if the unexpected happens and **there is downtime, you maybe be unable to access that data.**

Key Differences	On-Premise	Cloud
Security	• Companies that have extra **sensitive information, such as government and banking industries** must have a certain level of security and privacy that an on-premises environment provides. • Despite the promise of the cloud, **security is the primary concern for many industries, so an on-premises environment**, despite some of its drawbacks and price tag, make more sense.	• Security concerns remain **the number one barrier to cloud computing deployment**. • There have been **many publicized cloud breaches, and IT departments around the world are concerned**. • From personal information of employees such as **login credentials to a loss of intellectual property, the security threats are real.**



Traditional model
On premise applications
Hardware
Defense Build/Own/Operate
Integrating traditional and cloud capability
Hybrid cloud model
On premise applications
Off premise applications
Public cloud
SaaS, IaaS and PaaS
Integrated
Private cloud
SaaS, IaaS, PaaS and DbaaS
Cloud services can be outsourced
Mobile and deployable apps

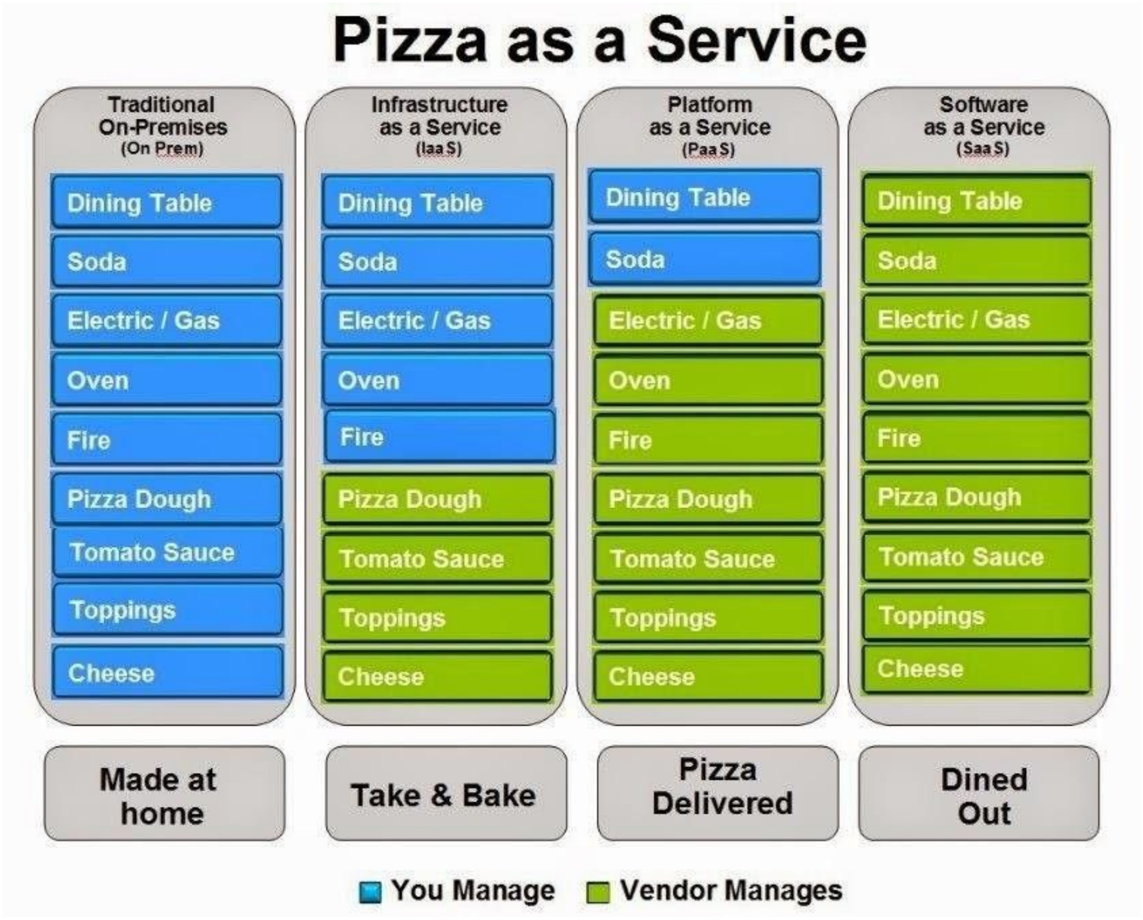
Microsoft Dynamics AX

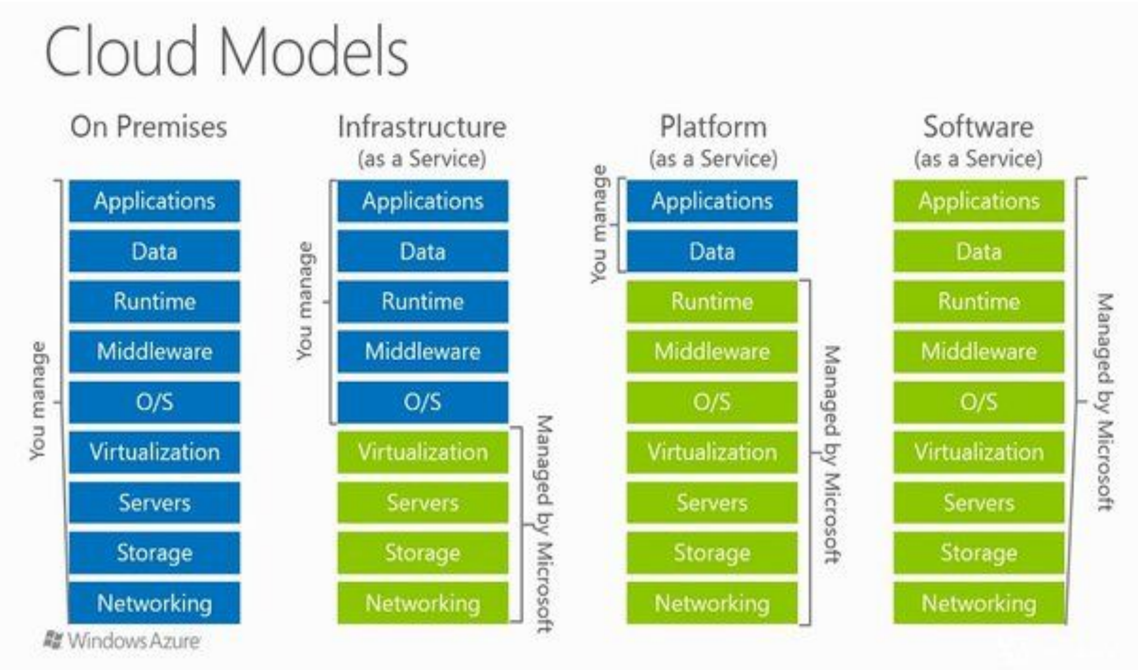
- Microsoft Dynamics AX is the flagship **ERP software** of Microsoft.
- It helps global enterprises **organize, automate, and optimize** their processes ***on-premises, in the cloud, or through hybrid deployment.***
- It offers out-of-the-box functionality that enables mid to large size companies manage their **administrative and operational business functions with ease.**
- Equipped with advanced security and capabilities to meet deployment and development needs of global structures.
- It offers functionality **across business processes, allows scalability and enables seamless integration with existing systems.**
- **Microsoft Dynamics AX** offers core strengths in manufacturing and ecommerce, making it the best ERP solution for the service and wholesale industries.

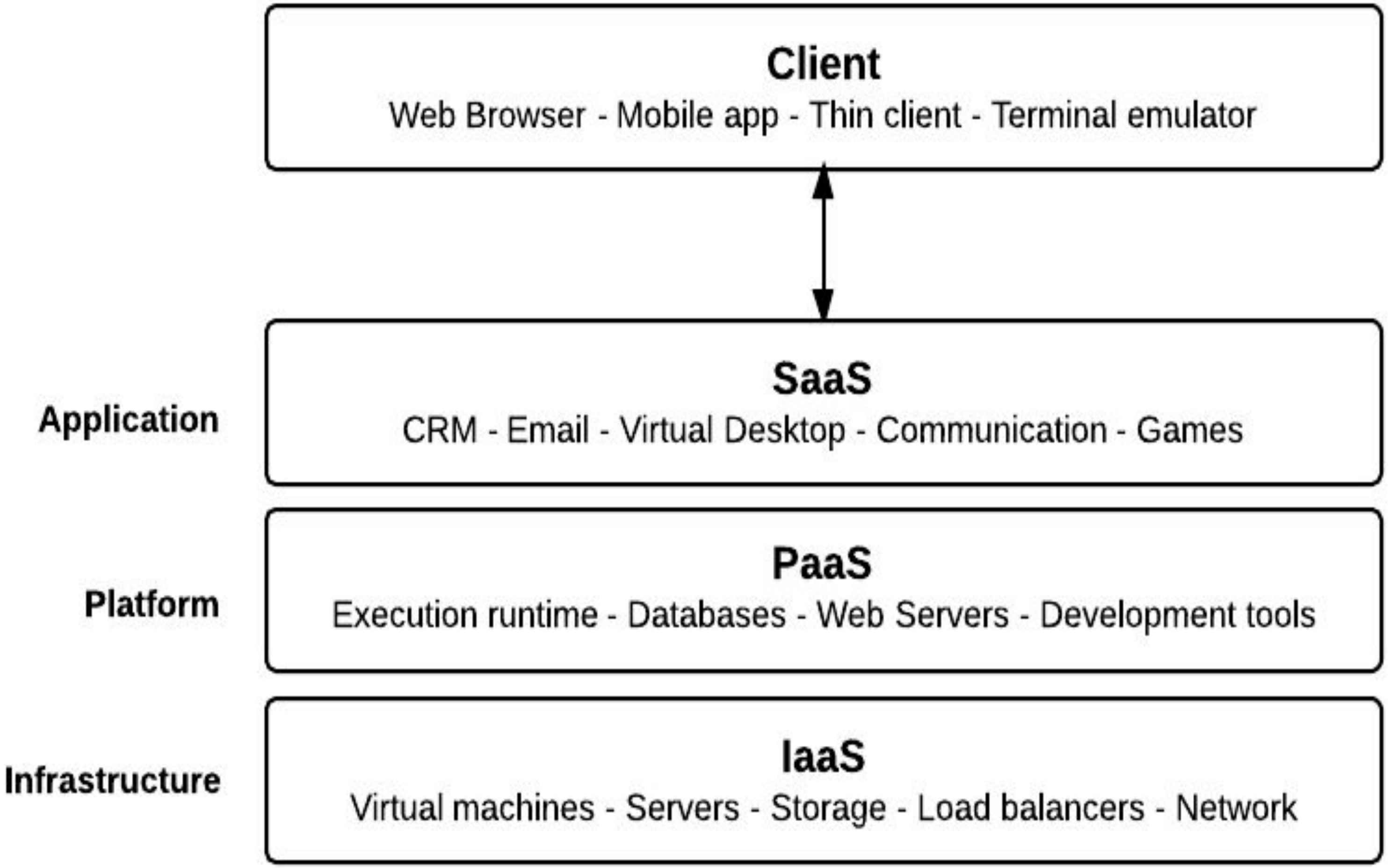
Examples

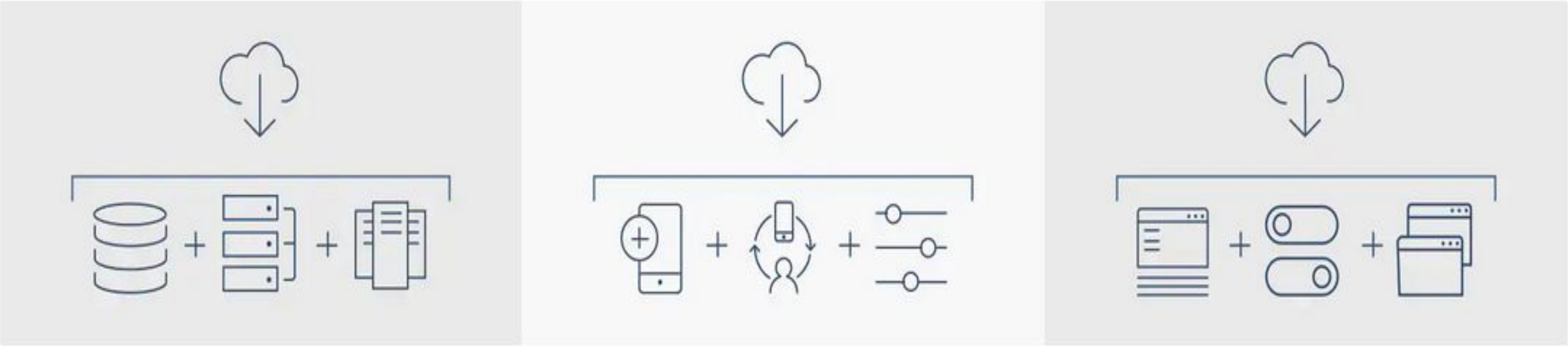
- **Cloud**
- Office 365
- **ADOBE CREATIVE CLOUD** *(SUBSCRIPTION BASED)*
- WebTrends On Demand
- **On-Premises**
- SharePoint 2013
- **ADOBE CREATIVE SUITE (***NON SUBSCRIPTION VERSION'S OF ADOBE APPS* **)**
- WebTrends On-Premises

https://laniteo.de/blog/creative-suite-vs-creative-cloud/


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
Soda
Electric / Gas
Oven
Fire
Pizza Dough
Tomato Sauce
Toppings
Cheese
Made at home
Take & Bake
Pizza Delivered
Dined Out
You Manage
Vendor Manages


Cloud Models
On Premises
Infrastructure (as a Service)
Platform (as a Service)
Software (as a Service)
Applications
Data
Runtime
Middleware
O/S
Virtualization
Servers
Storage
Networking
You manage
Managed by Microsoft
Windows Azure


Client
Web Browser - Mobile app - Thin client - Terminal emulator
Application
SaaS
CRM - Email - Virtual Desktop - Communication - Games
Platform
PaaS
Execution runtime - Databases - Web Servers - Development tools
Infrastructure
IaaS
Virtual machines - Servers - Storage - Load balancers - Network



Infrastructure as a service (IaaS)

A vendor provides clients pay-as-you-go access to storage, networking, servers and other computing resources in the cloud.

Platform as a service (PaaS)

A service provider offers access to a cloud-based environment in which users can build and deliver applications. The provider supplies underlying infrastructure.

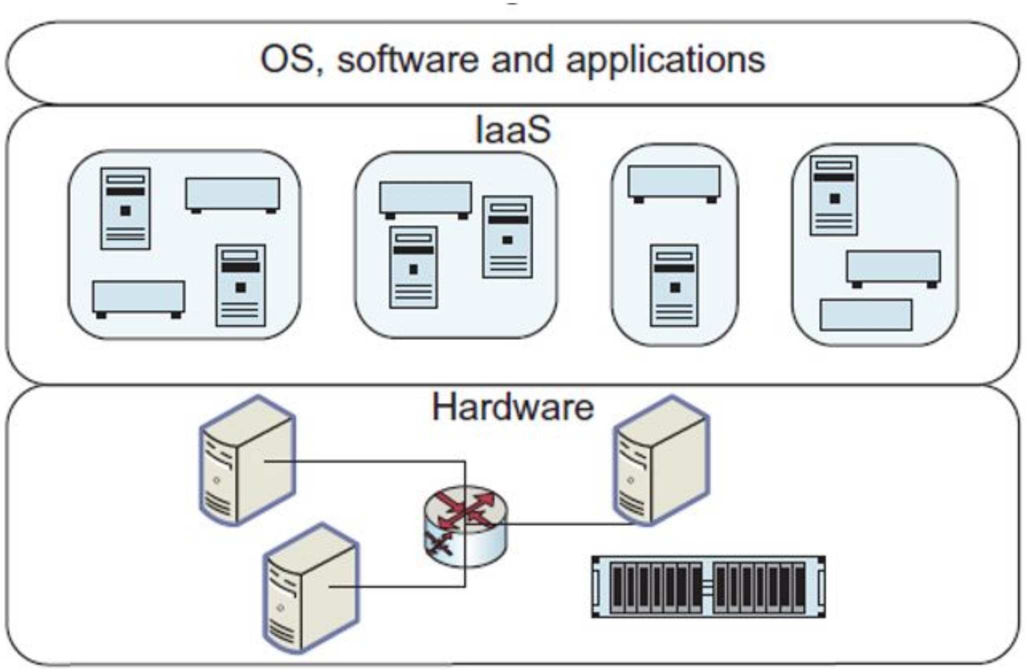
Software as a service (SaaS)

A service provider delivers software and applications through the internet. Users subscribe to the software and access it via the web or vendor APIs.

Infrastructure-as-a-Service (IaaS)

Definition	Characteristics	Services	Service Providers
• IaaS is also known as **Hardware as a Service (HaaS).** • It is a computing **infrastructure managed over the internet**. • The main advantage of using IaaS is that it helps users to avoid the cost and complexity of purchasing and managing the physical servers.	• **RESOURCES** are available as a service • **Services are highly scalable** • **Dynamic and flexible** • GUI and API-based access • **Automated administrative tasks**	**Computing** as a Service includes virtual central processing units and virtual main memory for the Vms that is provisioned to the end-users. **Storage:** IaaS provider provides back-end storage for storing files. **Network**: Network as a Service (NaaS) provides networking components such as routers, switches, and bridges for the Vms. **Load balancers:** It provides load balancing capability at the infrastructure layer.	**Amazon Web Services (AWS), Microsoft Azure, Google Compute Engine (GCE)**

Infrastructure-as-a-Service (IaaS)

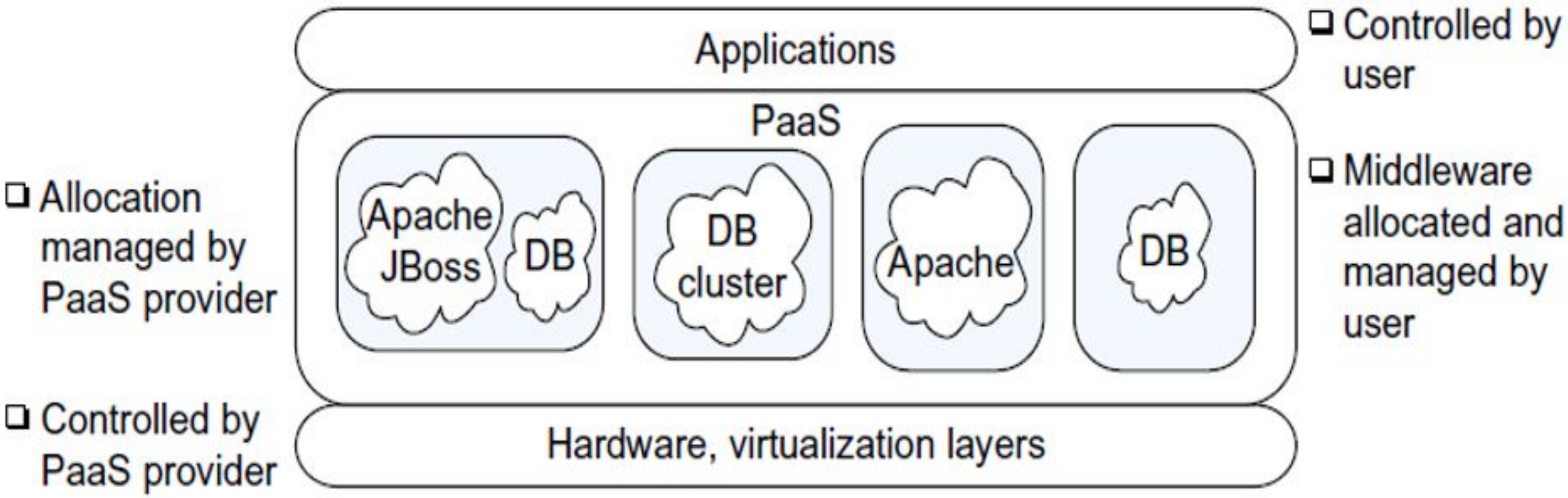


- Allocation managed by IaaS provider
- Controlled by IaaS provider
- Controlled by user
- Virtual resources allocated and managed by user

Platform as a Service (PaaS)

Definition	Characteristics	Services	Service Providers
• PaaS cloud computing platform is created for **the programmer to develop, test, run, and manage the applications**	• Accessible to various users via the same development application. • **Integrates with web services and databases**. • Builds on **virtualization technology**, so resources can easily be **scaled up or down as per the organization's need**. • Support multiple languages and frameworks. • Provides an ability to **"AUTO-SCALE"**.	PaaS includes infrastructure **(servers, storage, and networking)** and platform (**middleware, development tools, database management systems, business intelligence, and more)** to support the web application life cycle	AWS Elastic Beanstalk, Windows Azure, Heroku, Force.com, Google App Engine, Apache Stratos, Magento Commerce Cloud, and OpenShift.

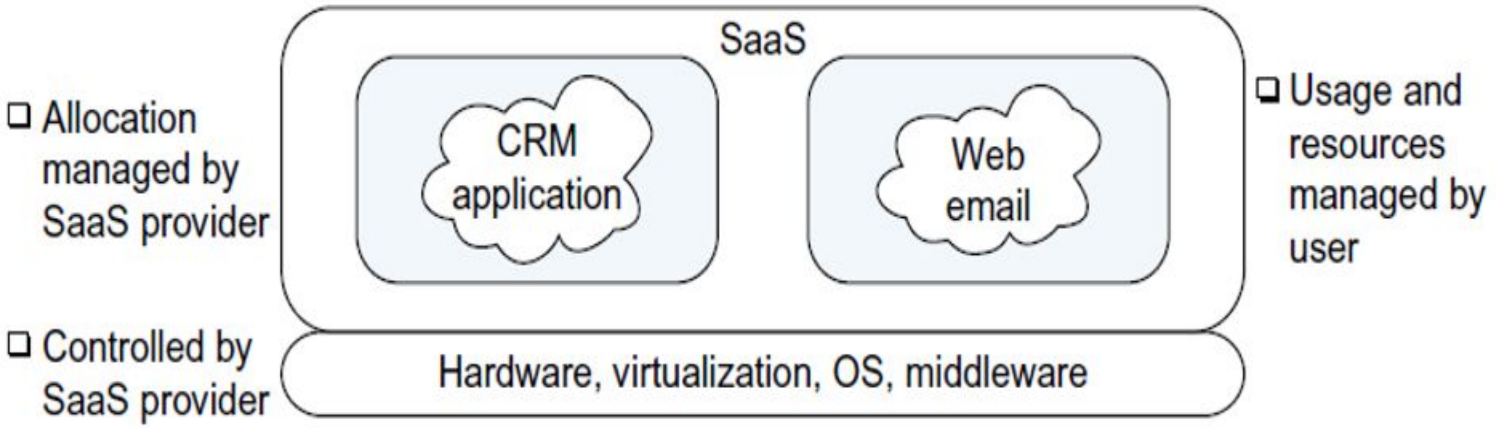
Platform as a Service (PaaS)

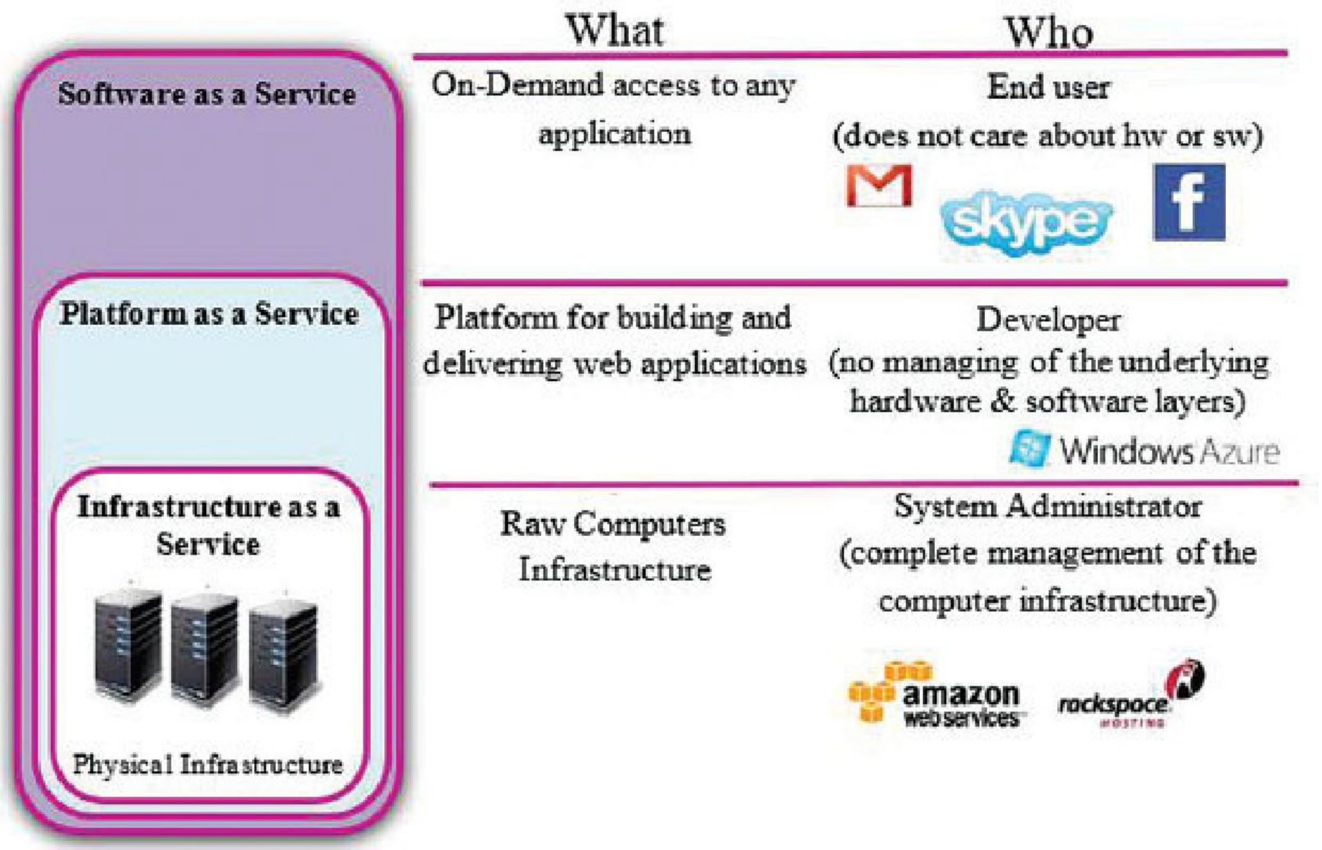


Software as a Service (SaaS)

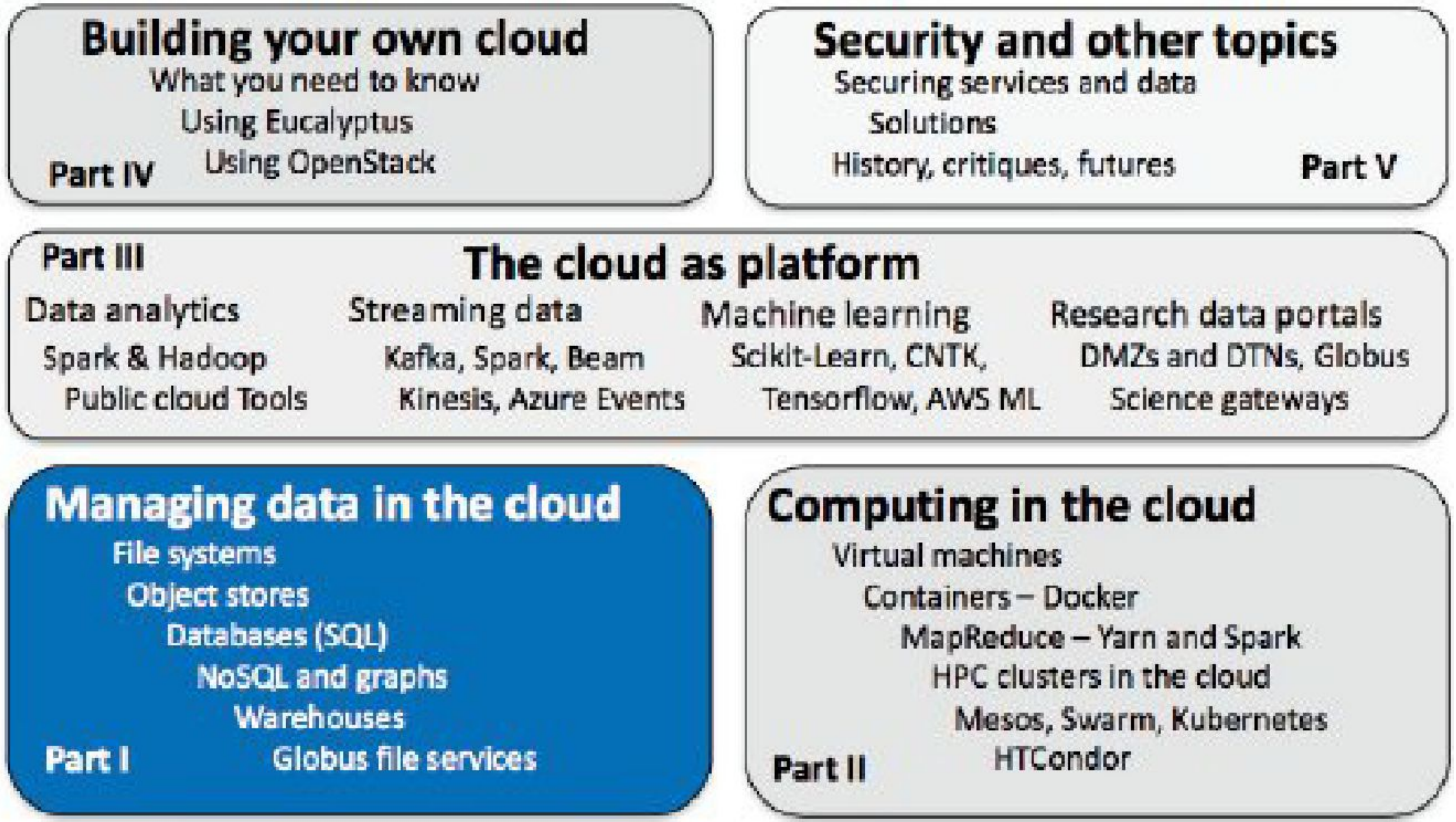
Definition	Characteristics	Services	Service Providers
• SaaS is also known as "**on-demand software".** • It is a software in which the applications are hosted by a **cloud service provider.** • Users can access these applications **with the help of internet connection and web browser**.	• Managed from a central location • Hosted on **a remote server** • Accessible over the internet • Users are not **responsible for hardware and software updates**. Updates are applied automatically. • The services are purchased on the pay-as-per-use basis	• Business Services • Document Management Electronic documents. • Social Networks • Mail Services	Google Apps, Salesforce, Dropbox, ZenDesk, Cisco WebEx, ZenDesk, Slack, and GoToMeeting

Software as a Service (SaaS)




Software as a Service
Platform as a Service
Infrastructure as a Service
Physical Infrastructure
What
Who
On-Demand access to any application
End user
(does not care about hw or sw)
skype
Platform for building and delivering web applications
Developer
(no managing of the underlying hardware & software layers)
Windows Azure
Raw Computers Infrastructure
System Administrator
(complete management of the computer infrastructure)
amazon web services
rackspace HOSTING

Managing Data in the Cloud



Managing Data in the Cloud

Blob storage

- Blob is shorthand for **Binary Large Object,** provides a flat object model for data.
- It is extremely scalable, in ways that are challenging for file systems.

- **AZURE BLOB STORAGE**
- **Page Blobs**
- **Block Blobs**
- **Append Blobs**

File system

- Storage is the well-known model of organizing
- Data into folders and directories. In the cloud, file storage
- Is usually accessed by attaching a virtual disk to a virtual Machine.

- **Cloud file storage (CFS)**

Data warehouses

- That can support and enable search over
- Massive amounts of data.

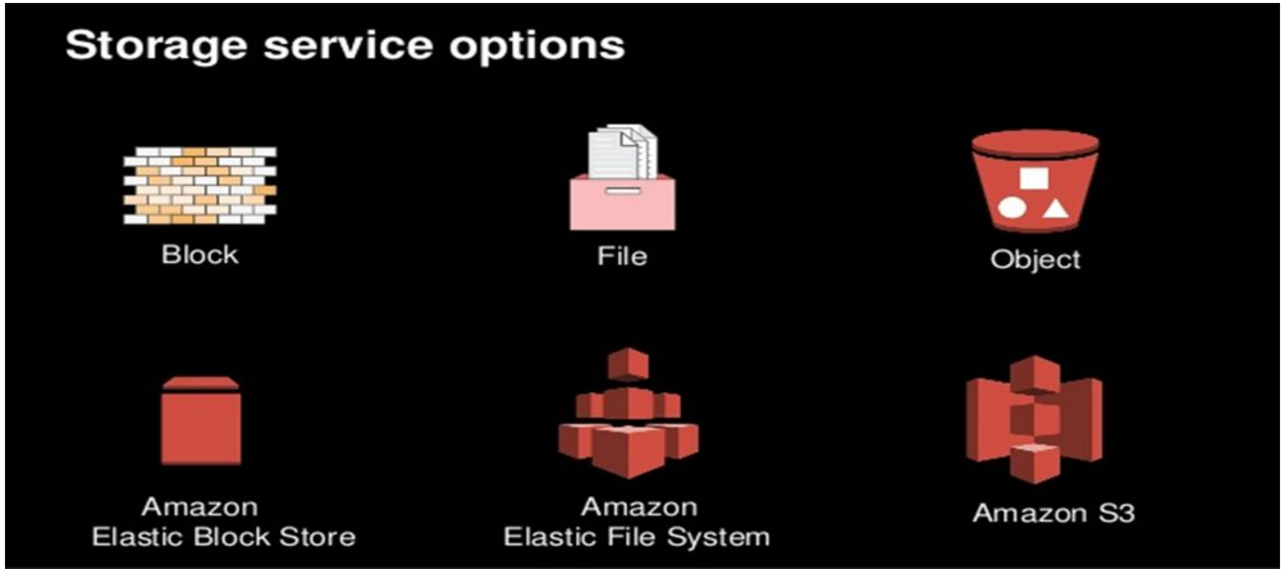
- **Azure Synapse Analytics**
- **Amazon Redshift**
- **Google BigQuery**

Databases

- Relational databases, which have a formal Algebra
- The structured query language, SQL.
- Tables and nosql databases, which are more
- Easily distributed over multiple machines.
- Graph databases, in which data are represented
- As a graph of nodes and edges
- **Amazon Aurora, Google Cloud SQL**
- **Amazon DynamoDB, Amazon SimpleDB[, Oracle NoSQL Database Cloud Service**

Storage as a Service

- IaaS platforms for storage as a service and then compute as a service.
- Storage as a Service (sometimes abbreviated as SaaS) takes a detailed look at key Amazon Storage Services:
  - **Amazon Simple Storage Service (S3)**, which provides a highly reliable and highly available object store over HTTP.
  - **Amazon SimpleDB**, a key-value store
  - **Amazon Relational Database Service (RDS),** which provides a MySQL instance in the cloud

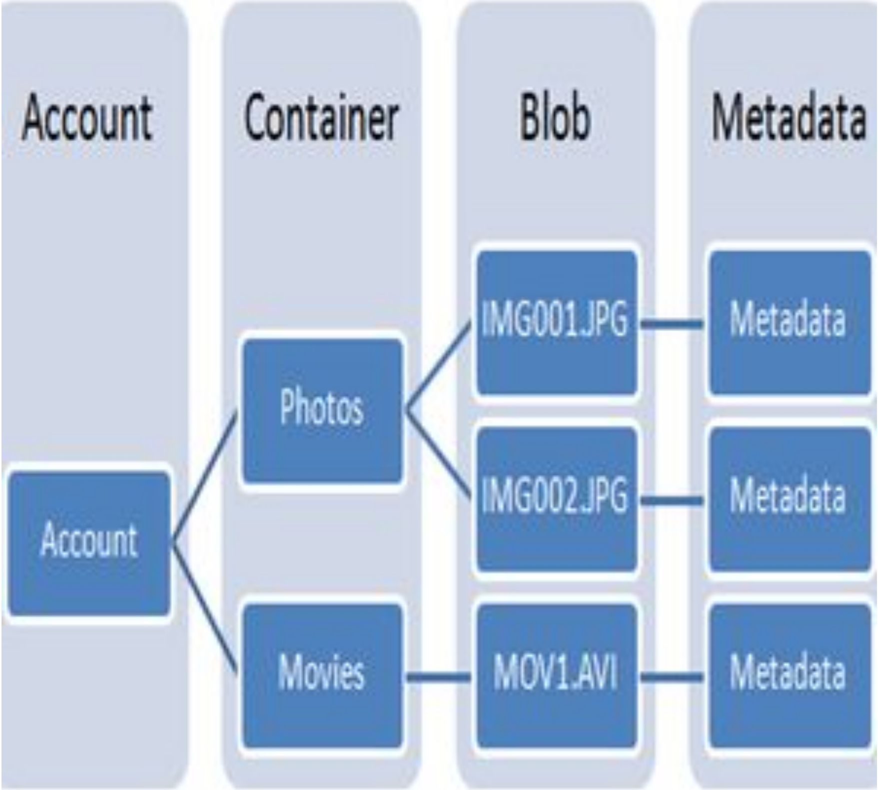


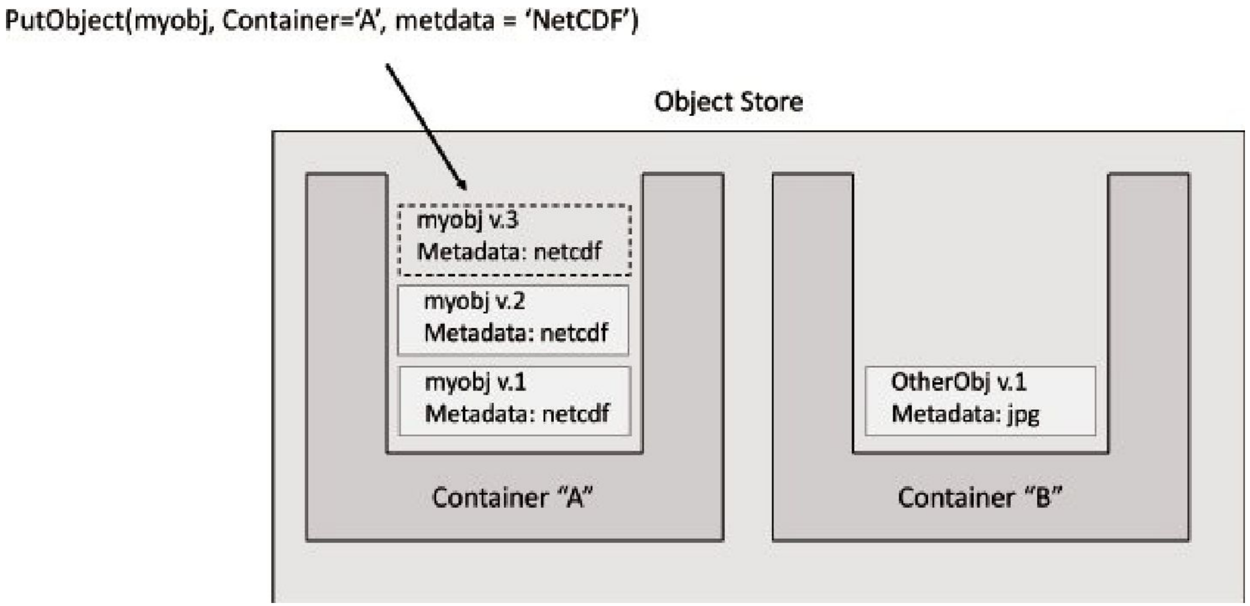
File Systems

- A file system is a process of managing how and where data on a storage disk, which is also referred to as **FILE MANAGEMENT OR FS.**
- The standard API for the Unix-derived version of the file system is called the **Portable Operating System Interface (POSIX).**
- **Create, read, write, and delete files located within directories**.
- The file system model also has disadvantages as a basis for science and engineering, particularly as **data volumes grow.**

Object Stores

- The object storage model, like the file system model, **stores unstructured binary objects**.
- In the database world, objects are often referred to as **BLOBS, FOR BINARY LARGE OBJECT**, and we use that name here when it is consistent with terminology adopted by cloud vendors.
- **VERSIONING** helps you recover accidental overwrites and deletes.
- The object can be restored to a prior version, and even deletes can be undone.
- This guarantees that data is never unintentionally lost.


Account
Container
Blob
Metadata
Account
Photos
Movies
IMG001.JPG
IMG002.JPG
MOV1.AVI
Metadata
Metadata
Metadata


PutObject(myobj, Container='A', metdata = 'NetCDF')
Object Store
myobj v.3
Metadata: netcdf
myobj v.2
Metadata: netcdf
myobj v.1
Metadata: netcdf
Container "A"
OtherObj v.1
Metadata: jpg
Container "B"

Relational Databases

- A database is a structured **collection of data** about entities and their relationships.
- A database management system (**DBMS**) is a software suite designed to **safely store and efficiently manage databases** and to assist with the maintenance and discovery of the relationships that databases represent.
- A **DBMS** encompasses three components: its data model (which defines how data are represented), its query language (which defines how the user interacts with the data), and support for transactions and crash recovery (to ensure reliable execution despite system failures).

- ACID semantics
- Atomicity (the entire transaction succeeds or fails)
- Consistency (the data collection is never left in an invalid or conflicting state)
- Isolation (concurrent transactions cannot interfere with each other)
- Durability (once a transaction completes, system failures cannot invalidate the result).

NoSQL Databases

- NoSQL is a database technology driven by Cloud Computing, the Web, Big Data and the Big Users.

- NoSQL Database, also known as **"Not Only SQL"** is an alternative to SQL database which does not require any kind of fixed table schemas unlike the SQL.

- NoSQL generally scales horizontally and avoids major join operations on the data.

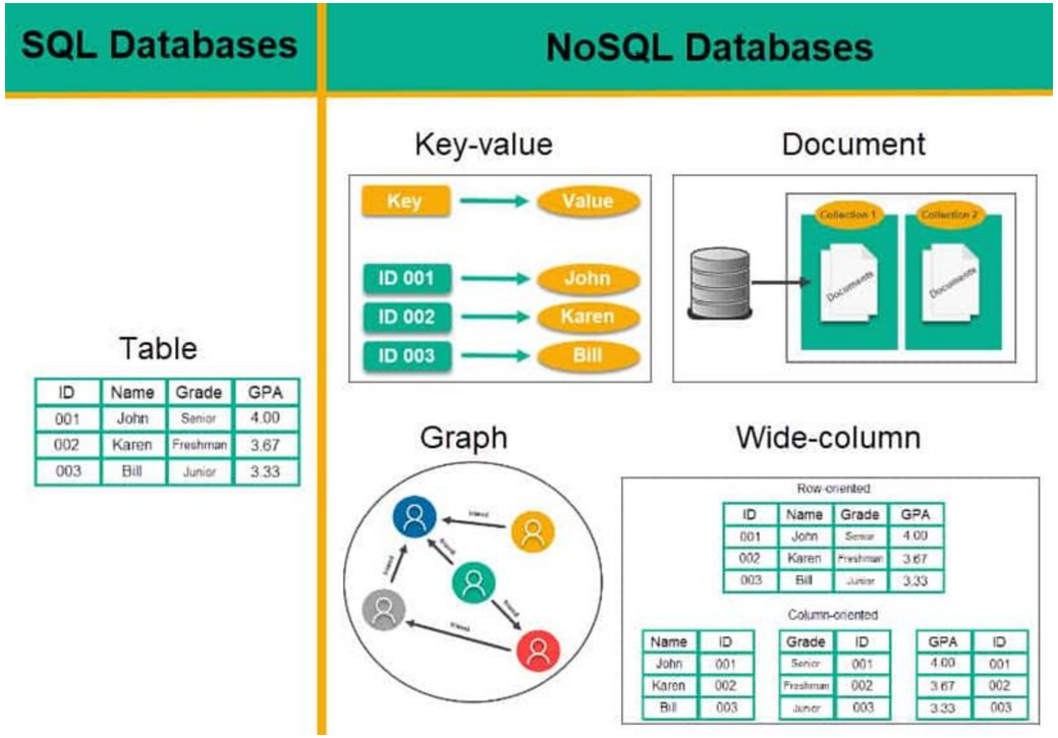
Types of No SQl data base

- **Key Value pair**
- **Document Based**
- **Graph database**
- **Column Oriented database**

NonRelational

- Couch DB - D
- Mongo DB - D
- Cassandra – C
- Neo4j – G
- Infinitegraph - G
- Aerospike - KV
- Riak - KV

NoSQL Databases

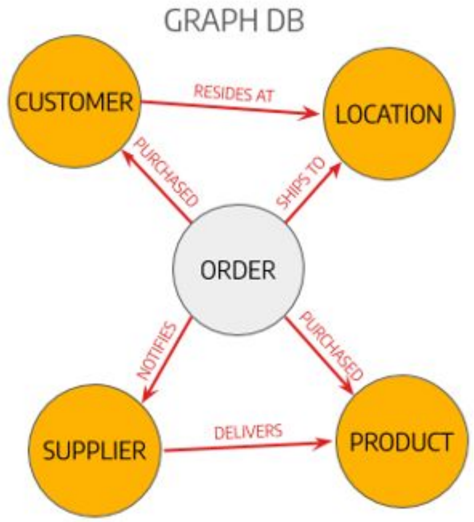


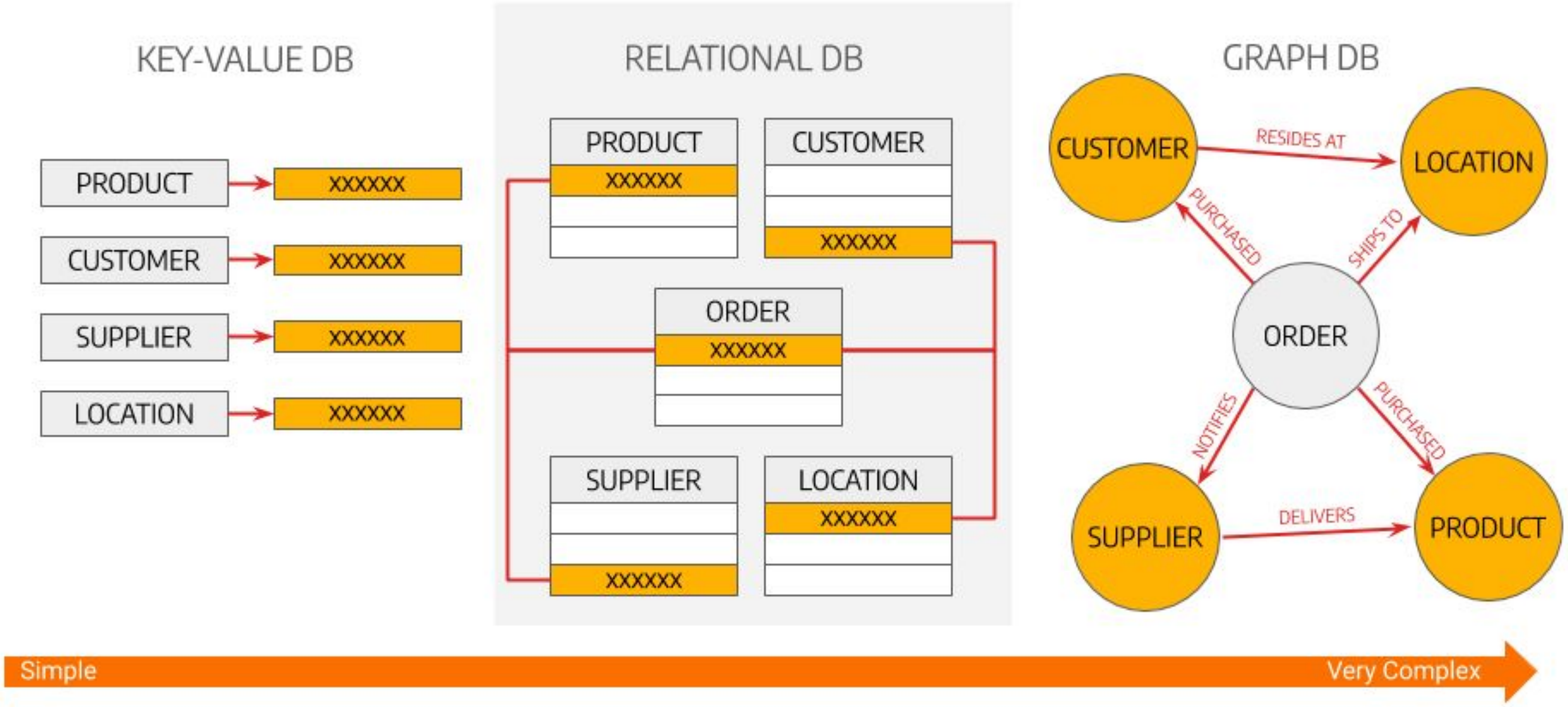
Challenges of scale: The CAP theorem

- *Consistency indicates that all computers see the same data at the same time.*
- *Availability indicates that every request receives a response about whether it succeeded or failed.*
- *Partition tolerance indicates that the system continues to operate even if a network failure prevents computers from communicating.*

Graph Databases

- A graph is a data structure in which edges connect nodes.
- Graphs are useful when we need to search data based on relationships among data items





Ability to handle complex interconnected data

Data Warehouses

- The term data warehouse is commonly used to refer to data management systems optimized to support **analytic queries that involve reading large datasets.**
- Data warehouses have different design goals and properties than do DBMSs

The Cloud Storage Landscape

- The Cloud Storage Landscape
  - File Systems
    - Elastic Block Store (EBS) and Elastic File System (EFS), Google Compute Engine, Azure File Storage
  - Object Stores
    - Simple Storage Service (S3), Cloud Storage, Azure Storage
  - NoSQL Services
    - Amazon's DynamoDB, Amazon Elastic MapReduce, Cloud Bigtable, Cloud Datastore, Azure Table, Azure HDInsight, DocumentDB

The Cloud Storage Landscape

- The Cloud Storage Landscape
  - Relational Databases
    - **Relational Database Service (RDS), Amazon Aurora, Google’s Cloud SQL, Azure’s SQL Database**
  - Warehouse Analytics
    - **Amazon Redshift, BigQuery, Azure Data Lake**
  - Graphs and More
    - **Amazon’s Titan, Simple Queue Service, Elasticsearch Service, Amazon Kinesis, Graph Engine**
  - OpenStack Storage Services and Jetstream
    - **OpenStack Shared File Systems**

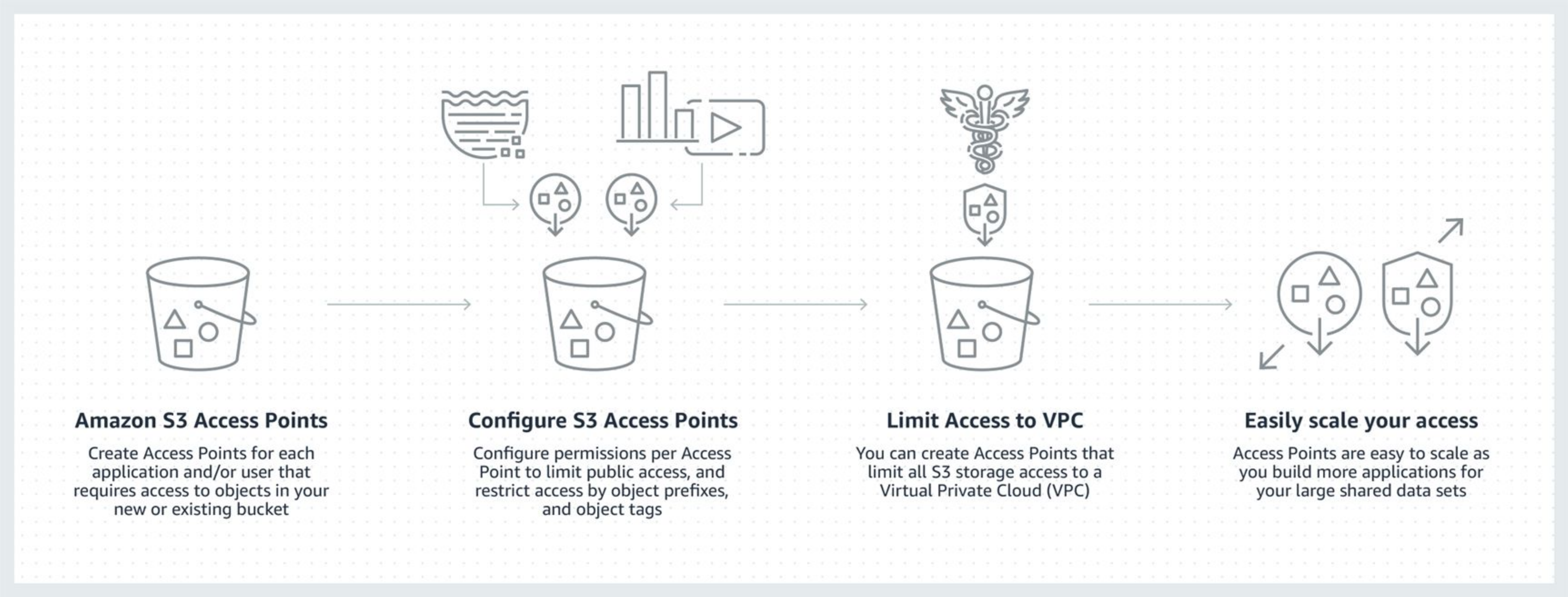
File Systems

- File systems (also referred to **as file shares**) are virtual data drives that can be attached to virtual machines
  - ***Amazon's Elastic Block Store (EBS) and Elastic File System (EFS)***
- EBS provides a **BLOCK STORAGE SERVICE FOR EC2**.
- It is possible to request an EBS disk volume of a particular size and attach this volume to **one or multiple EC2 instances** using the **INSTANCE ID returned during the time the volume is created.**
- EBS is a device that you can mount onto a single **Amazon EC2 compute server instance at a time**
- It is designed for applications that require low-latency access to data from a single EC2 instance

- **GOOGLE COMPUTE ENGINE** has a different attached storage model.
- There are three types of attached disks (and also a way to attach an object store).
- The cheapest, persistent disks, can be up to 64 TB in size
- The **AZURE FILE STORAGE** service allows users to create file shares in the cloud that can be accessed by a special protocol.
- *Server Message Block (SMB)*that allows Microsoft Windows VMs and Linux VMs to mount these file shares as standard parts of their file system.

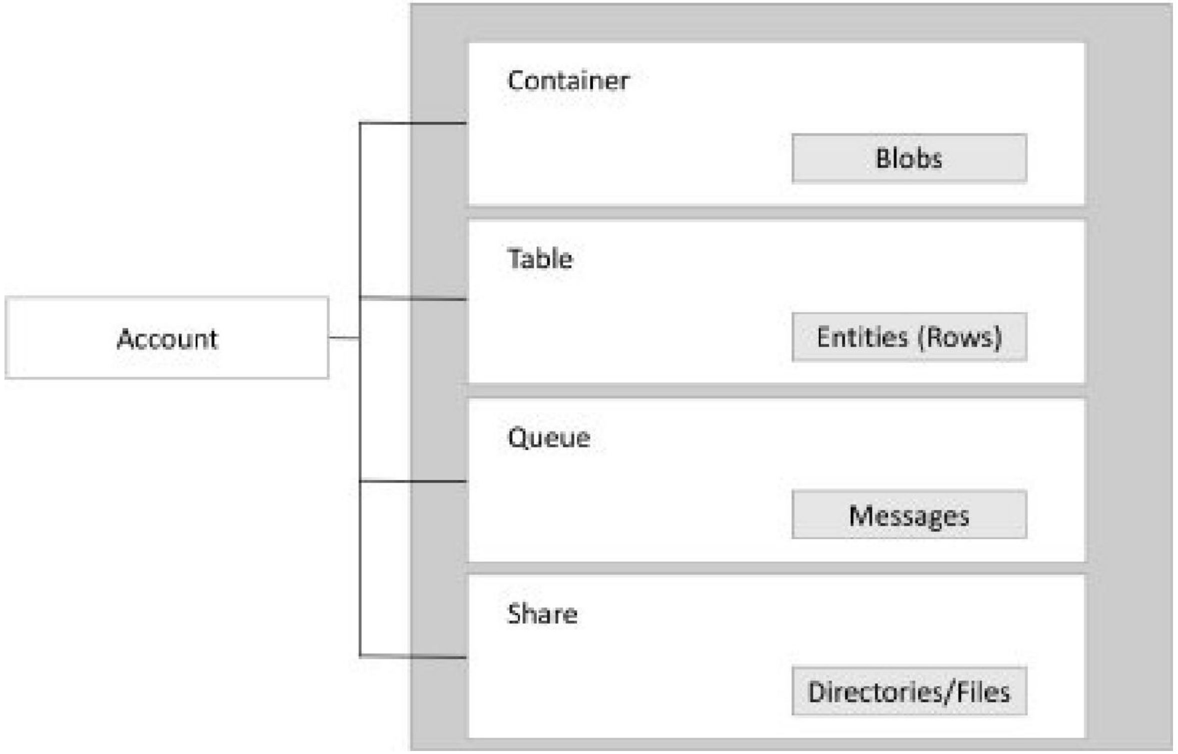
Object Stores

- **AMAZON'S SIMPLE STORAGE SERVICE (S3)** was historically its first cloud service.
- Amazon Web Services (AWS), from Amazon.com, has a suite of cloud service products that have become very popular and are almost looked up to as a de facto standard for delivering IaaS.
- Amazon S3 is a *highly reliable, highly available, scalable and fast storage in the cloud for storing and retrieving large amounts of data just through simple web services*.


Amazon S3 Access Points
Create Access Points for each application and/or user that requires access to objects in your new or existing bucket
Configure S3 Access Points
Configure permissions per Access Point to limit public access, and restrict access by object prefixes, and object tags
Limit Access to VPC
You can create Access Points that limit all S3 storage access to a Virtual Private Cloud (VPC)
Easily scale your access
Access Points are easy to scale as you build more applications for your large shared data sets

- **Azure Storage offers a suite of services with a similar scope** in terms of models supported to those provided by Amazon and Google.
- Azure provides the user with a unified view of many of its storage types associated with their account.
- The **Azure Blob storage service**, like Amazon's S3, is concerned with **highly reliable storage of unstructured objects, which Microsoft calls blobs**.

- **BLOBS** are (typically large) **UNSTRUCTURED OBJECTS** like *images and media*, and are similar to Amazon S3.
- Applications deal with blobs as a whole, although they might **READ/ WRITE parts of a blob**.
- Blobs can have **OPTIONAL METADATA** associated with them in the **FORM OF KEY-VALUE PAIRS**.
- Blobs are always stored **UNDER CONTAINERS**, which are similar to **AWS buckets**.
- Every storage account must have at **least one container**, and containers can have blobs within them.


Account
Container
Blobs
Table
Entities (Rows)
Queue
Messages
Share
Directories/Files

NoSQL Services

- **AMAZON'S DYNAMODB** is a powerful NoSQL database based on an **extensible key-value model**
- For each row, the **primary key column** is the only required attribute, but any number of additional columns can be **defined, indexed, and made searchable** in various ways, including full-text search via **ELASTICSEARCH**.

- **CLOUD BIGTABLE** Google's highly scalable NoSQL database service, is the same database that powers many core *Google services, including Search, Analytics, Maps, and Gmail*.
- Bigtable maps **TWO ARBITRARY STRINGS** (**row key and column key**) and a **TIMESTAMP** (**permitting versioning and garbage collection) to an associated arbitrary byte array**.
- It is designed to handle such **large and sparse datasets** in a manner that is efficient in space used and that supports massive workloads, while providing low latency and high bandwidth.

- The **AZURE TABLE STORAGE SERVICE** is a simple NoSQL key-value store, designed to support the highly reliable storage of any large number of key-value pairs.
- It is similar to Amazon DynamoDB.
- Its **query capabilities are limited**, but it can support many queries at modest cost.

- **AZURE HDINSIGHT** provides an implementation of the **Hadoop storage service** hosted on Azure cloud computers, with implementations of popular big data tools, including Spark
- **HBASE NOSQL DATABASE**, and the **HIVE SQL DATABASE** implemented to run efficiently and scalably on top of that Hadoop fabric

Relational Databases

- **Amazon's Relational Database Service (RDS)** allows you to set up a conventional relational database (e.g., **MySQL or Postgres**) on Amazon computers, thus permitting MySQL and Postgres applications to be ported to Amazon without change.
- The **MySQLcompatible Amazon Aurora** service provides *higher scalability, performance, and resilience than an RDS MySQL instance*

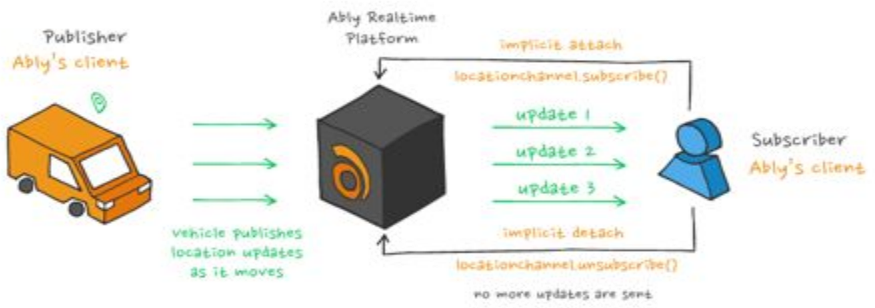
- **Google's Cloud SQL** relational database service has similar capabilities to those provided by **such services in Amazon and Azure.**
- **Azure's SQL Database** provides a relational database service similar to Amazon RDS.
- It is based on their **mature SQL Server technology** and is highly available and scalable

Warehouse Analytics

- **Amazon Redshift** is a data warehouse system, designed to support high-performance execution of **analytic and reporting workloads against large datasets.**
- For massive data analytics, Google provides the **BigQuery petascale data warehouse.**
- **BigQuery** is fully distributed and replicated, so durability is not an issue.
- It also supports SQL query semantics.
- The **Azure Data Lake** is a full suite of data analytics tools built on the open-source **YARN and WebHDFS** platforms

Graphs and More

- Messaging services allow applications to send and receive messages using what are referred to as **PUBLISH/SUBSCRIBE SEMANTICS.**
- **AMAZON'S TITAN** extension to DynamoDB supports graph databases.
- **Elasticsearch** open-source **search and analytics engine**
- **Amazon Kinesis** supports **analysis of stream data.**
- Google supports the open source **graph database Cayley**.
- Its **Cloud Pub/Sub** service provides **messaging** in a similar manner to **Amazon SQS**.

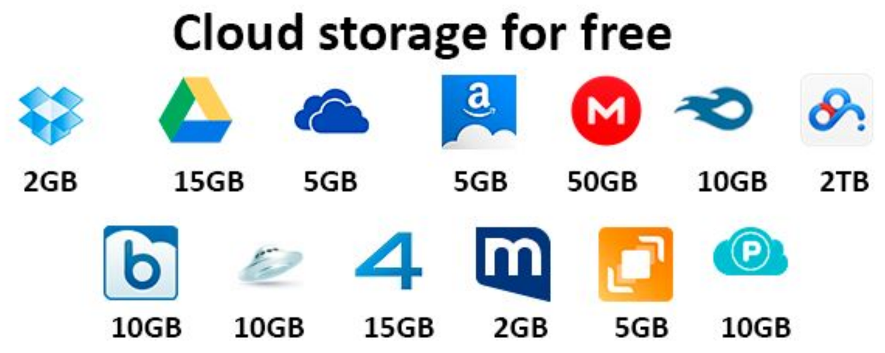


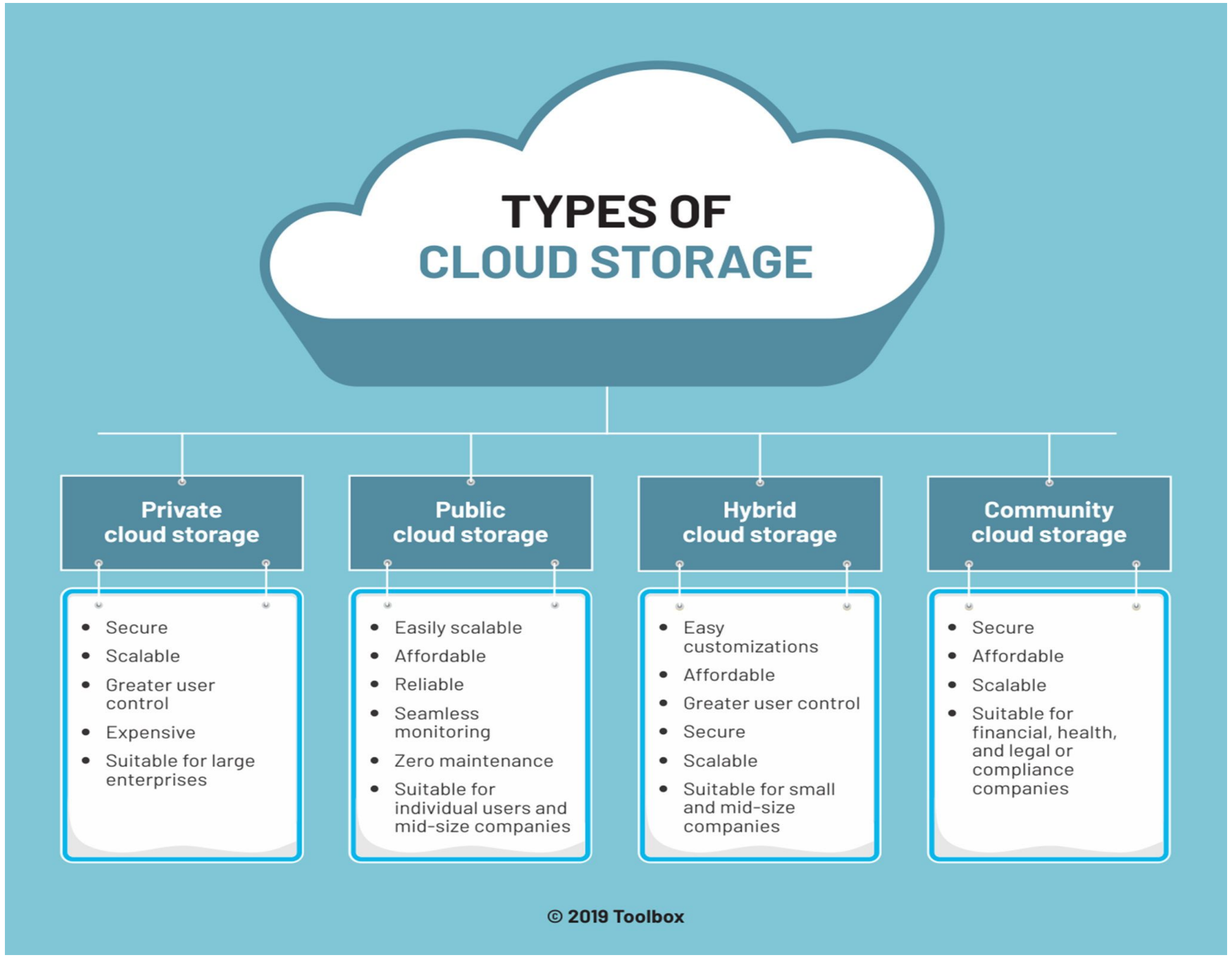
OpenStack Storage Services and Jetstream

- The OpenStack **open source cloud software** supports only a few standard storage services***: object storage, block storage, and file system storage.***
- The OpenStack object storage service is called **Swift**.
- The OpenStack Shared File Systems service, like the **Amazon EFS and Azure File service**, implements a file system model in the cloud environment.

Using Cloud Storage Services

- The services of different cloud providers are often similar in outline, they invariably differ in the details.





- Using Cloud Storage Services
  - Using Amazon Cloud Storage
  - Using Microsoft Azure Storage
  - Using Google Cloud Storage
    - Google Cloud Datastore
    - Google Bigtable
  - Using OpenStack Cloud Storage
  - Transferring and Sharing Data with Globus

Cloud hosting Vs Self hosting

- Cloud hosting is when an **organization installs and accesses software on a server over the internet** that is owned and managed by a third party organization.
- Self hosting is when an organization **installs and accesses software from their own server**. Self hosting is also often referred to as on premises or local hosting.

Two Access Methods: Portals and APIs

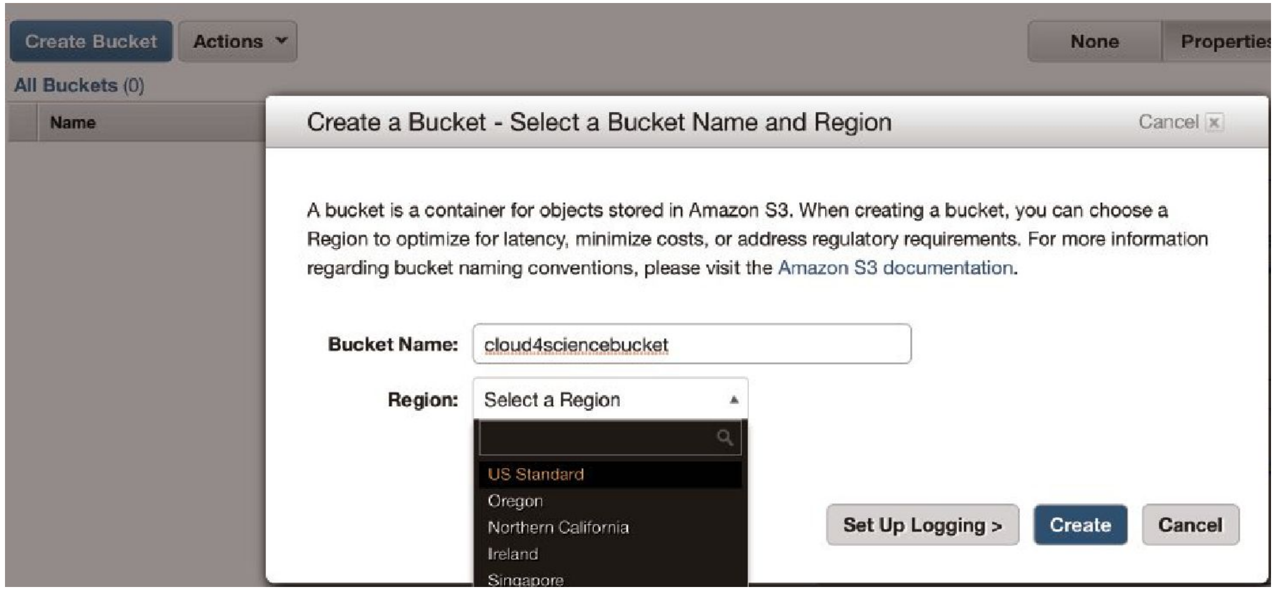
- Cloud providers make this possible by providing **REST APIs** that programmers can use to access their services programmatically.
- For programming convenience, you will usually access these APIs via software development kits (SDKs), which give programmers language-specific functions for interacting with cloud services.
- Application Programming Interface (API) portals enable companies to easily share information and data across real-time, distributed cloud and mobile applications

- **Representational State Transfer** (REST) application programming interface (API) that permits requests to be transmitted via the secure Hypertext Transfer Protocol (HTTPS) that is used by web browsers.
- **LOCAL AND CLOUD-HOSTED APPLICATIONS**

```
PUT / HTTP/1.1
Host: cloud4sciencebucket.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
<CreateBucketConfiguration
              xmlns="http://s3.amazonaws.com/doc
              /2006-03-01/">
  <LocationConstraint>US
  Standard</LocationConstraint>
  </CreateBucketConfiguration>
```

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

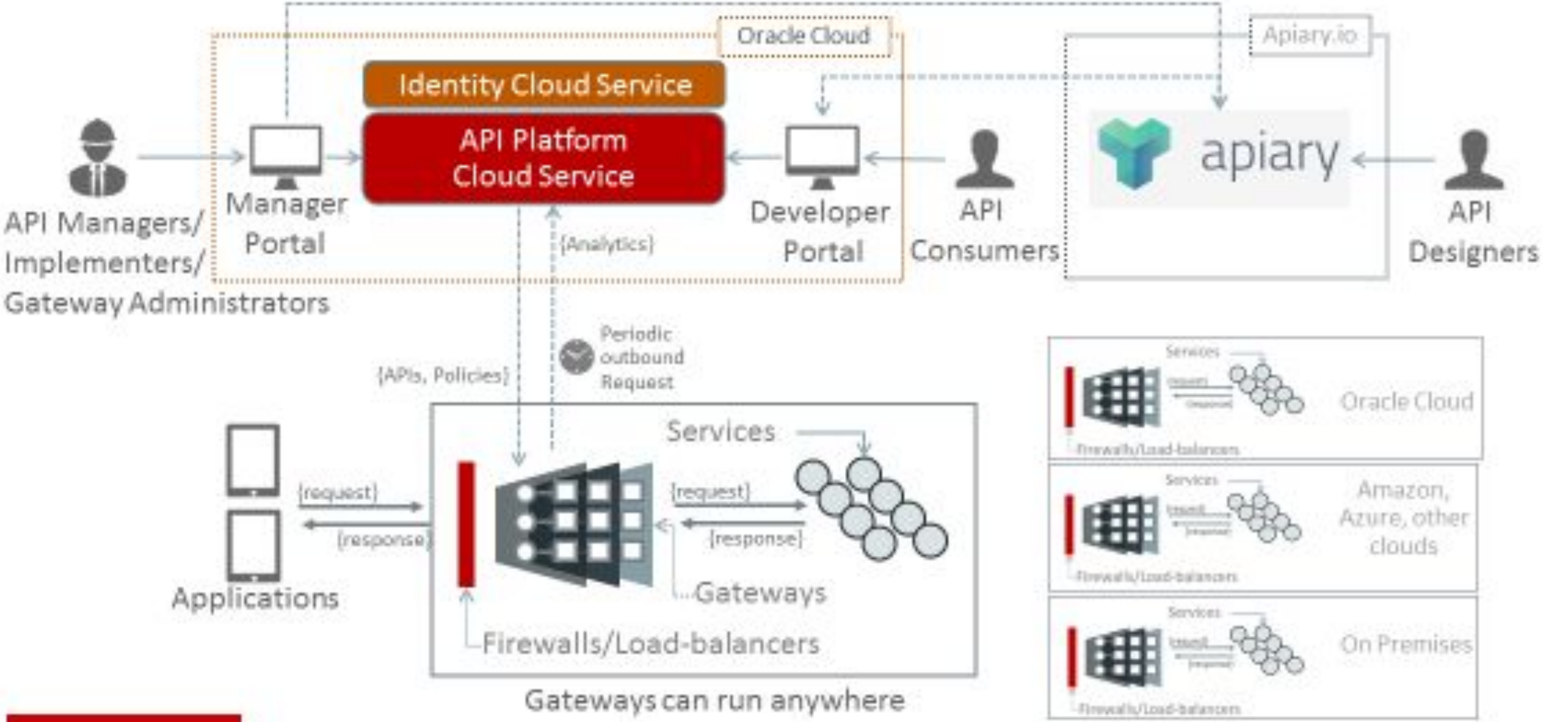
```
import boto3
s3 = boto3.resource('s3')

# Delete the bucket previously created with the
REST API
s3.Bucket('cloud3sciencebucket').delete()

# Create that bucket again, specifying location
bucket = s3.create_bucket(Bucket =
'cloud4sciencebucket',
                      CreateBucketConfiguration
                      ={
                      'LocationConstraint':
                      'us-standard'})

# Upload a file 'test.jpg' into the newly
created bucket
bucket.put_object(Key='test.jpg',
Body=open('test.jpg', 'rb'))
```



Key concepts in above code

- **Boto3** is the name of the Python SDK for AWS. It allows you to directly **create, update, and delete AWS** resources from your Python scripts.
- Common Operations:
    - Creating a Bucket
    - Naming Your Files
    - Creating Bucket and Object Instances
    - Understanding Sub-resources
    - Uploading a File
    - Downloading a File
    - Copying an Object Between Buckets
    - Deleting an Object

Architecture of API Platform Cloud Service



Copyright © 2016, Oracle and/or its affiliates. All rights reserved.

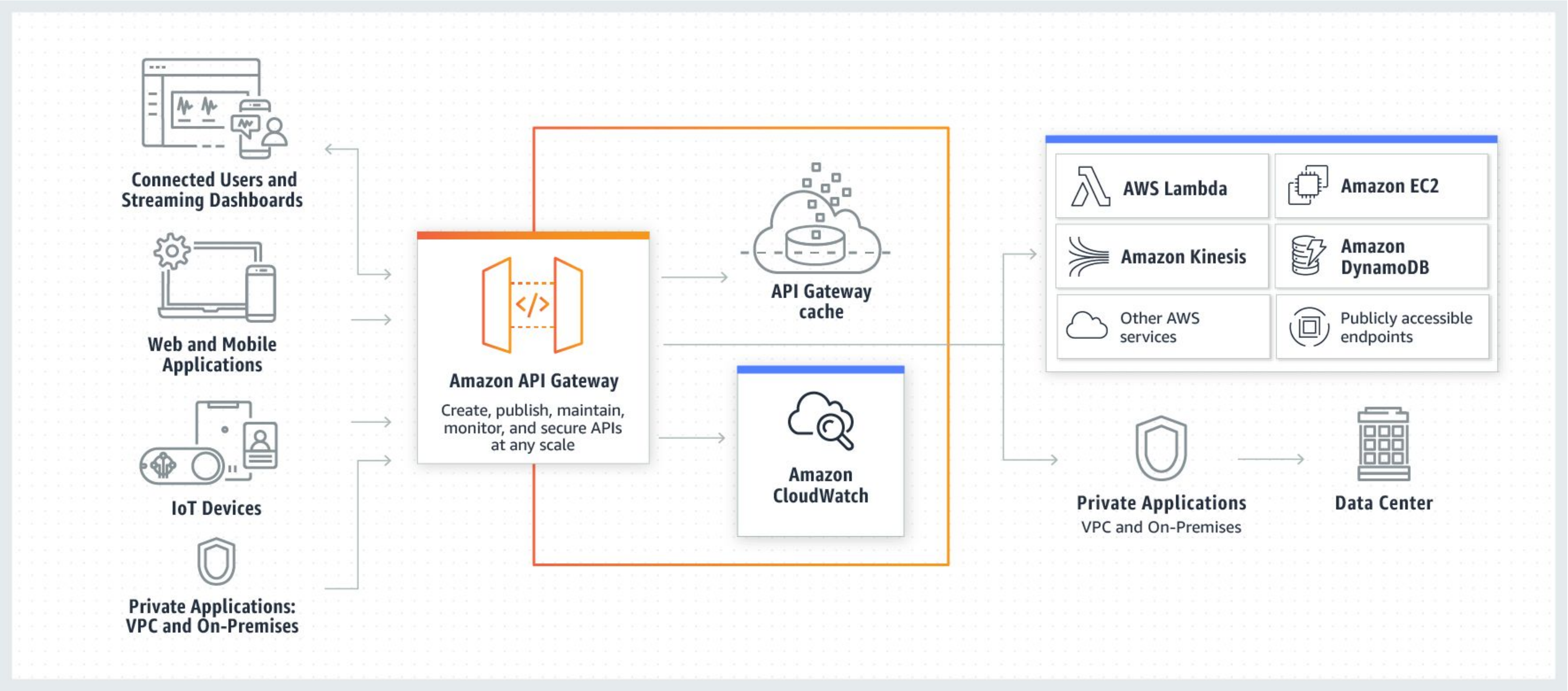
- **MANAGEMENT PORTAL:** APIs are **managed, secured, and published using the Management Portal.** The Management Portal is hosted on the Oracle Cloud, managed by Oracle, and users granted API Manager privileges have access.

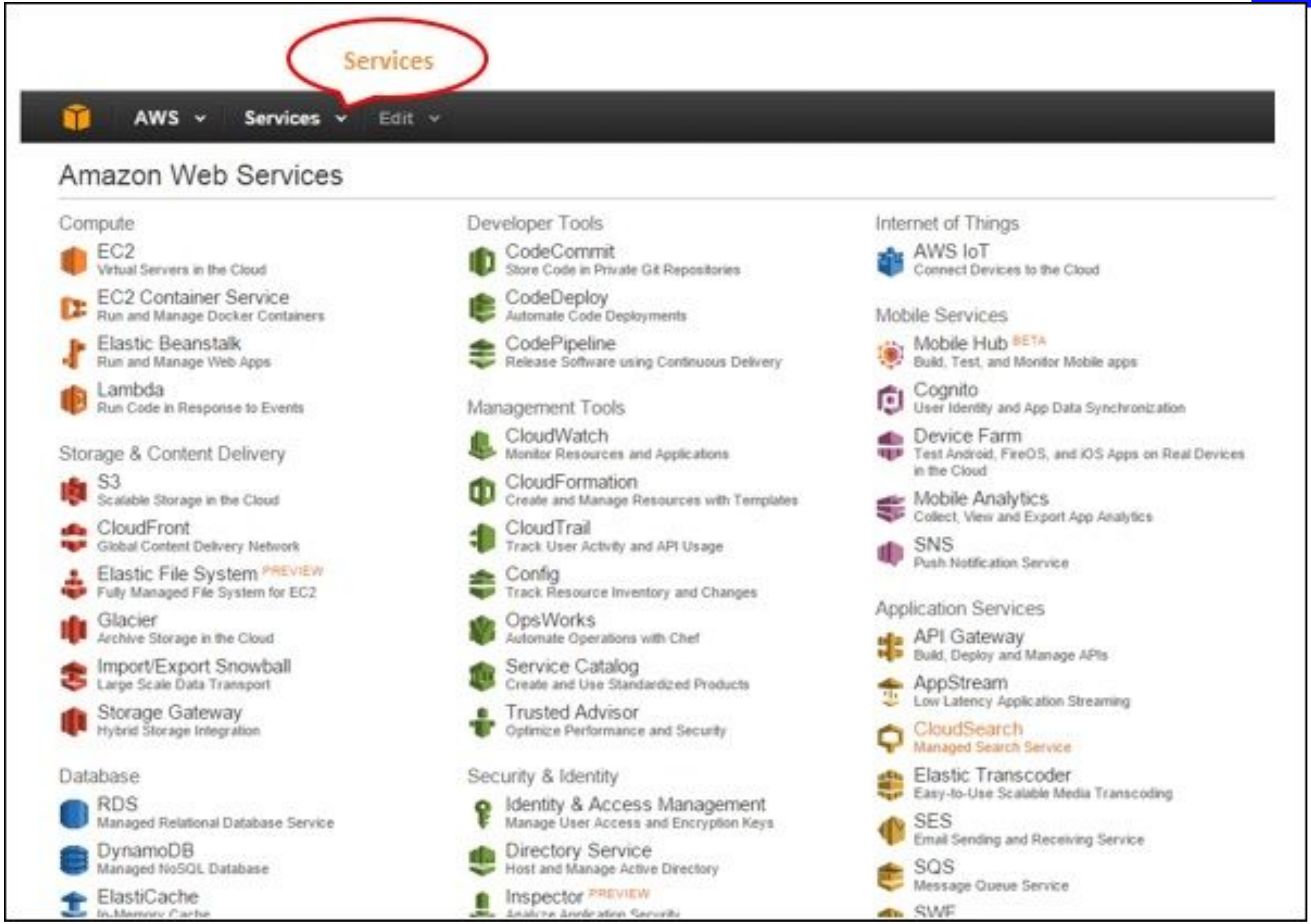
- **GATEWAYS:** API Gateways are the runtime components that enforce all policies, but also help in **collecting data for analytics**. The gateways can be deployed anywhere – on premise, on Oracle Cloud or to any third party cloud providers.

- **DEVELOPER PORTAL:** After an API is **published, Application Developers use the Developer Portal to discover, register, and consume APIs**. The Developer Portal can be customized to run either on the Oracle Cloud or directly in the customer environment on premises.

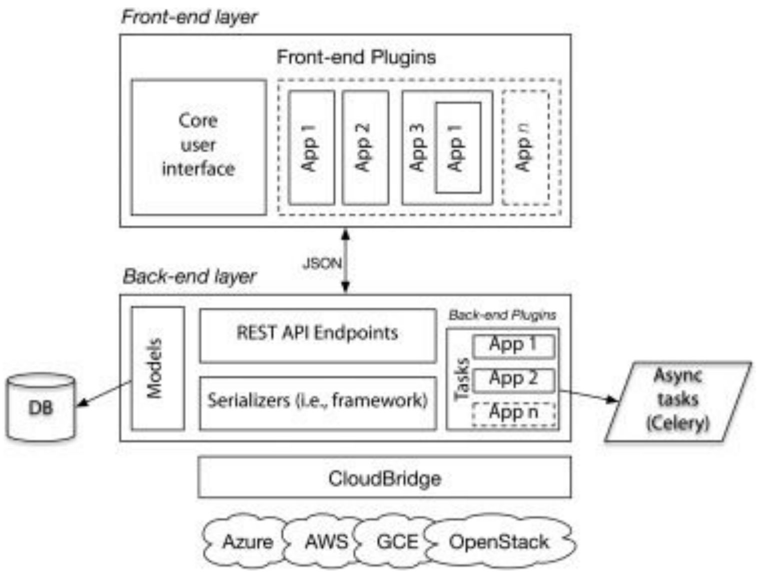
- **ORACLE APIARY:** Apiary provides the world's first platform, API Flow, specifically designed to help companies accelerate and control the design, development, and documentation of their APIs and microservices.
- APIs make up **the new language for businesses to communicate with each other**.
- As they increase in importance, more responsibility lies on those who build and manage the APIs.
- Apiary solves the **fundamental task of improving API development**, but for many companies, meeting those increasing expectations means not only working harder but also streamlining the business process of how work gets done.

Scenario Based Diagrams




Services
AWS
Services
Edit
Amazon Web Services
Compute
EC2
Virtual Servers in the Cloud
EC2 Container Service
Run and Manage Docker Containers
Elastic Beanstalk
Run and Manage Web Apps
Lambda
Run Code in Response to Events
Storage & Content Delivery
S3
Scalable Storage in the Cloud
CloudFront
Global Content Delivery Network
Elastic File System PREVIEW
Fully Managed File System for EC2
Glacier
Archive Storage in the Cloud
Import/Export Snowball
Large Scale Data Transport
Storage Gateway
Hybrid Storage Integration
Database
RDS
Managed Relational Database Service
DynamoDB
Managed NoSQL Database
ElastiCache
Developer Tools
CodeCommit
Store Code in Private Git Repositories
CodeDeploy
Automate Code Deployments
CodePipeline
Release Software using Continuous Delivery
Management Tools
CloudWatch
Monitor Resources and Applications
CloudFormation
Create and Manage Resources with Templates
CloudTrail
Track User Activity and API Usage
Config
Track Resource Inventory and Changes
OpsWorks
Automate Operations with Chef
Service Catalog
Create and Use Standardized Products
Trusted Advisor
Optimize Performance and Security
Security & Identity
Identity & Access Management
Manage User Access and Encryption Keys
Directory Service
Host and Manage Active Directory
Inspector PREVIEW
Internet of Things
AWS IoT
Connect Devices to the Cloud
Mobile Services
Mobile Hub BETA
Build, Test, and Monitor Mobile apps
Cognito
User Identity and App Data Synchronization
Device Farm
Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
Mobile Analytics
Collect, View and Export App Analytics
SNS
Push Notification Service
Application Services
API Gateway
Build, Deploy and Manage APIs
AppStream
Low Latency Application Streaming
CloudSearch
Managed Search Service
Elastic Transcoder
Easy-to-Use Scalable Media Transcoding
SES
Email Sending and Receiving Service
SQS
Message Queue Service

- Each cloud has special features that make it unique, and thus the different cloud provider's REST APIs and SDKs are not identical.
- Two efforts are under way to create a standard Python SDK: CloudBridge and Apache Libcloud libcloud.apache.org.

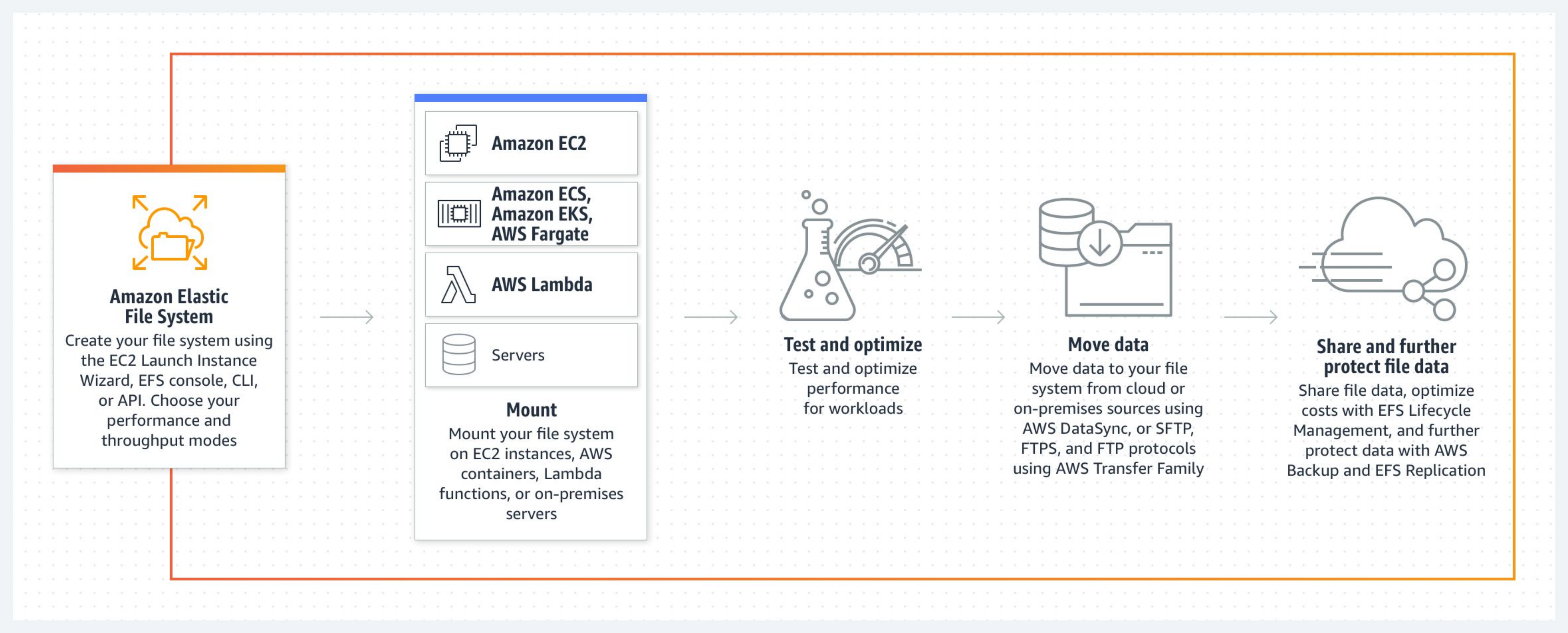


- **BUILDING A DATA SAMPLE COLLECTION IN THE CLOUD**
  - *Have a collection of data samples stored on our personal computer*
  - *For each sample use* ***METADATA****: item number, creation date, experiment id, and a text string comment.*
  - *To enable access to these samples by our collaborators, we want to upload them to cloud storage*
  - *To create a searchable table, also hosted in the cloud, containing the metadata and cloud storage* ***URL*** *for each object*
  - *Each data sample is in a binary file on our personal computer and that the associated metadata are contained in a comma separated value (CSV) file, with one line per item, also on our personal computer.*
  - *Each line in this csv file has the following format:* ***item id, experiment id, date, filename, comment string***

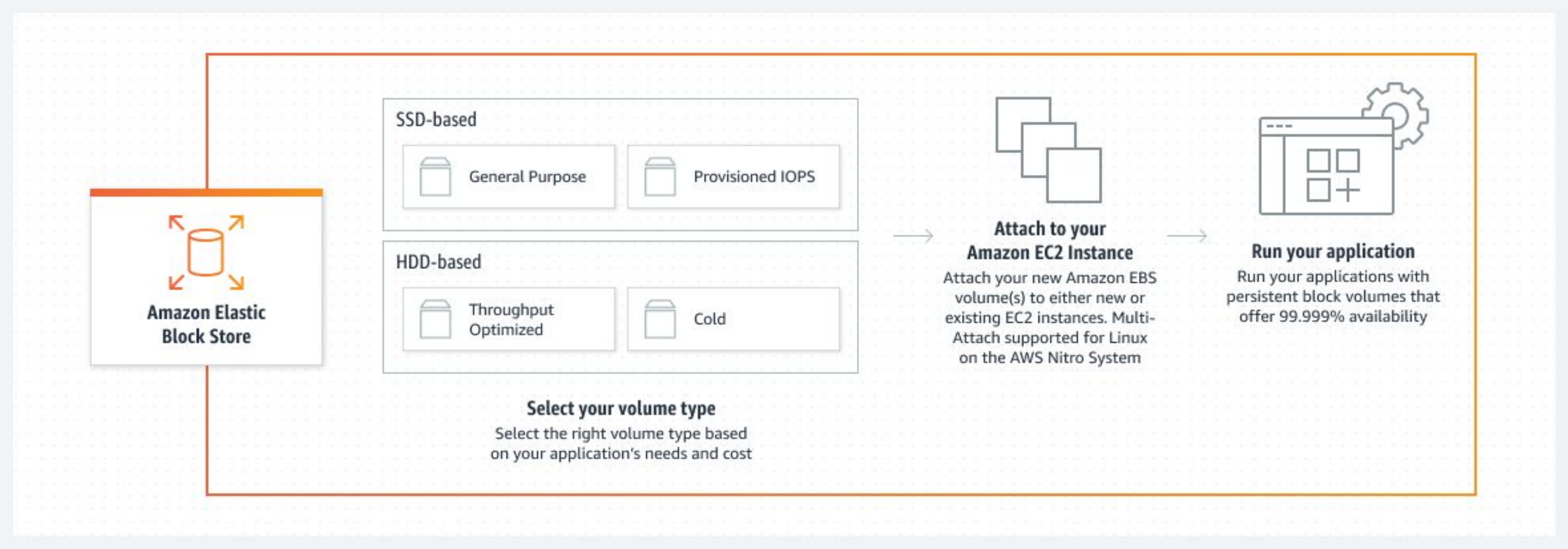
Amazon S3

Analytics data
Log files
Application data
Videos and pictures
Backup and archival
Move data
Move your data to Amazon S3 from wherever it lives – in the cloud, in applications, or on-premises
Amazon S3
Object storage built to store and retrieve any amount of data from anywhere
Store data
Create bucket, specify the Region, access controls, and management options. Upload any amount of data
Control access to data
Optimize cost with storage classes
Replicate data to any Region
Access from on-premises or VPC
Protect and secure your data
Gain visibility into your storage
Artificial Intelligence (AI)
Advanced analytics
Machine Learning (ML)
Analyze data
Use AWS and 3rd party services to analyze your data to gain insights

Amazon Elastic File System



Amazon Elastic Block Store (EBS)



Configure S3

- **Step 1** – Open the Amazon S3 console using this link – https://console.aws.amazon.com/s3/home
- **Step 2** – Create a Bucket using the following steps.
  - A prompt window will open. Click the Create Bucket button at the bottom of the page.
  - Create a Bucket dialog box will open. Fill the required details and click the Create button.
  - The bucket is created successfully in Amazon S3. The console displays the list of buckets and its properties.
  - Select the Static Website Hosting option.
  - Click the radio button Enable website hosting and fill the required details.

Configure S3

- **Step 3** – Add an Object to a bucket using the following steps.
  - Open the Amazon S3 console using the following link – https://console.aws.amazon.com/s3/home
  - Click the Upload button.
  - Click the Add files option. Select those files which are to be uploaded from the system and then click the Open button.
  - Click the start upload button. The files will get uploaded into the bucket.
  - Note: **To open/download an object** – In the Amazon S3 console, in the Objects & Folders list, right-click on the object to be opened/downloaded. Then, select the required object.

Move S3 Objects

- **step 1** – Open Amazon S3 console.
- **step 2** – Select the files & folders option in the panel. Right-click on the object that is to be moved and click the Cut option.
- **step 3** – Open the location where we want this object.
- **step 4** Right-click on the folder/bucket where the object is to be moved and click the Paste into option.

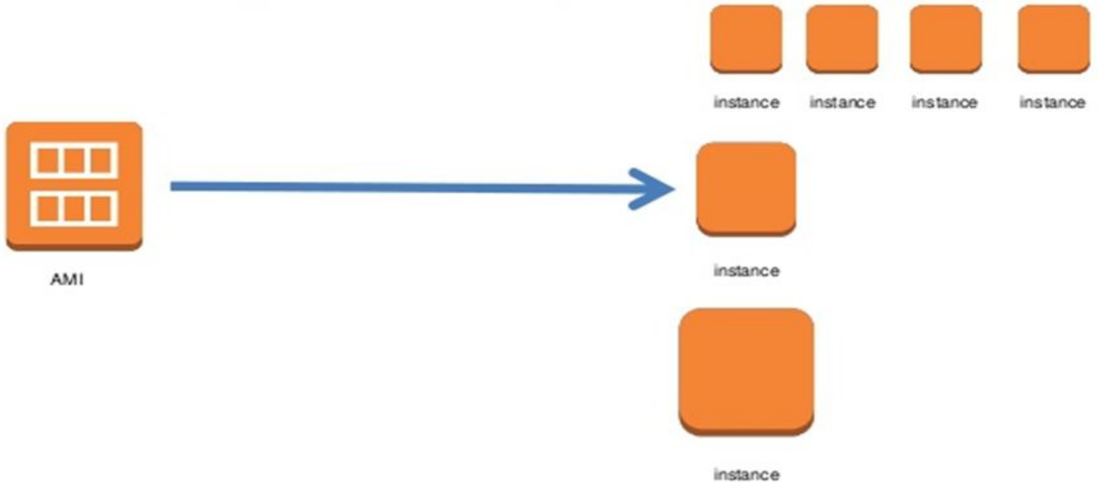
Delete an Object

- **Step 1** – Open Amazon S3.
- **Step 2** – Select the files & folders option in the panel. Right-click on the object that is to be deleted. Select the delete option.
- **Step 3** – A pop-up window will open for confirmation. Click Ok

Empty a Bucket

- **Step 1** – Open Amazon S3 console.
- **Step 2** – Right-click on the bucket that is to be emptied and click the empty bucket option.
- **Step 3** – A confirmation message will appear on the pop-up window.
- **Step 4-** Read it carefully and click the **Empty bucket** button to confirm.

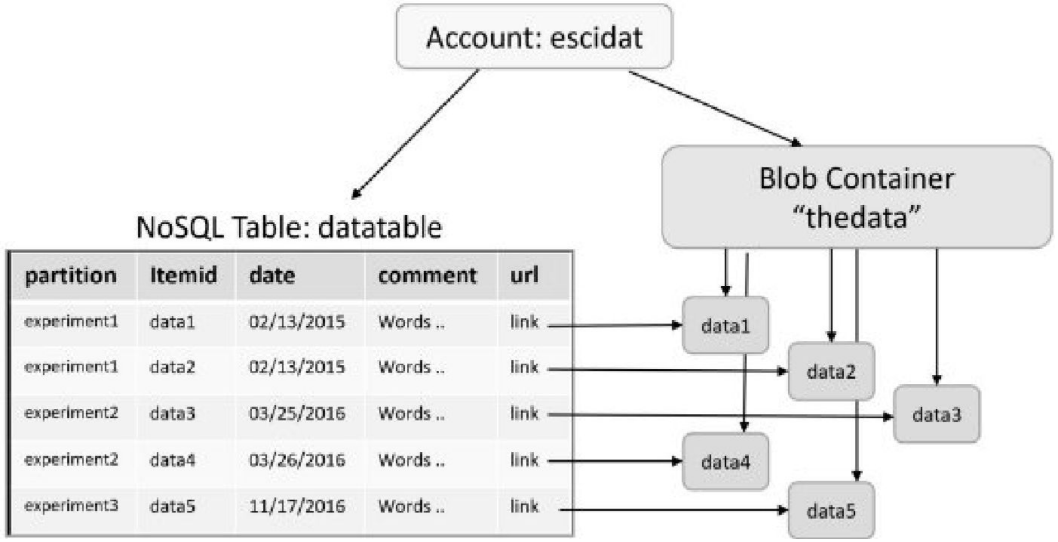
LAUNCH INSTANCE in EC2

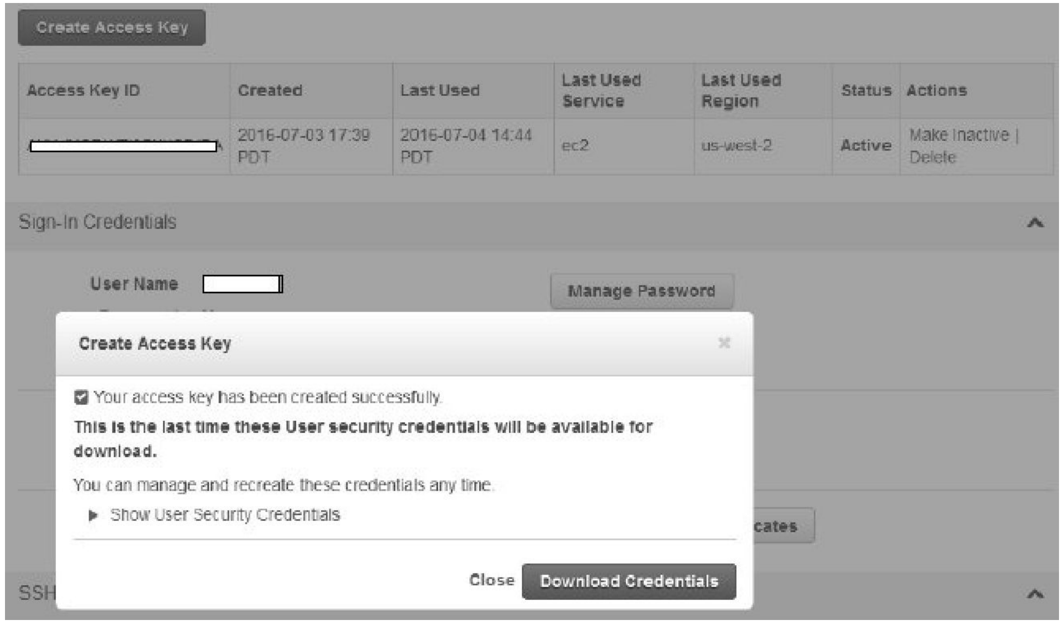


Using Amazon Cloud Storage Services

- **Need:** S3 to store ***the blobs and DynamoDB*** to store the table.
- Amazon key pair, i.e., **ACCESS KEY PLUS SECRET KEY**, which we can obtain from the Amazon **IAM MANAGEMENT** Console.
- Having created a new user, we select the create access key button to create our **security credentials, which we can then download**

- The simple cloud storage involves the upload of a collection of **data blobs to cloud storage and the creation of a NoSQL table containing metadata**
- Downloading **security credentials from the Amazon IAM Management Console**





Identity and Access Management (IAM)

- IAM provides **PERMISSION ACCESS** using the principle of 'least privilege' via IAM Access Analyzer to allow users to have access to only **the required AWS services to perform their job role in an organization or function**.
- IAM policies is used to **GRANT ACCESS** to AWS services to specify what type of access may be granted in the action table which include the following access levels:
- ***Read***
- ***Write***
- ***List***
- ***Permission Management***

Basic of S3

- Amazon Simple Storage Service (S3) is a **cloud object storage and retrieval service.**
- Objects stored in S3 may include file types such as ***csv, mp3 and images*** which are managed using **Application Programming Interface (API) through the internet with Hypertext Transfer Protocol Secure (HTTPS).**

Features	Description
Cors **(cross-origin resource sharing)**	**configure your bucket** to allow cross-origin requests.
Event notification	enable your **bucket to send you notifications of specified bucket events.**
Lifecycle	define **lifecycle rules for objects** in your bucket that have a well-defined lifecycle.
Location	When you create a bucket, you specify the **AWS Region** where you want Amazon S3 to create the bucket. Amazon S3 stores this information **in the location sub resource and provides an API for you to** retrieve this information.
Logging	Logging enables you to **track requests** for access to your bucket. Each access log record provides details about a single access *request, such as the requester, bucket name, request time, request action, response status, and error code*

Features	Description
Object locking	To use S3 **Object Lock**, you must enable it for a bucket. You can also optionally **configure a default retention mode** and period that applies to new objects that are placed in the bucket.
***Policy* and *ACL* (access control list)**	All your resources (such as buckets and objects) are **private by default.** Amazon S3 supports both bucket policy and **access control list (ACL)** options for you to grant and manage **bucket-level permissions**.
Replication	Replication is the automatic, **asynchronous copying** of objects across buckets in different or the same AWS Regions.
Requestpayment	By default, the **AWS account that creates the bucket** (the bucket owner) pays for downloads from the bucket.

Features	Description
Transfer acceleration	Transfer Acceleration **enables fast, easy, and secure transfers of files over long distances between your client** and an S3 bucket
Versioning	Versioning helps you recover **accidental overwrites and deletes**.
Website	You can configure your bucket for **static website hosting**. Amazon S3 stores this configuration by creating a *website* sub resource.
Tagging	You can add cost allocation tags to your bucket to categorize and track your AWS costs. Amazon S3 provides the *tagging* sub resource to store and manage tags on a bucket. Using tags you apply to your bucket, AWS generates a cost allocation report with usage and costs aggregated by your tags.

Bucket Names Rules

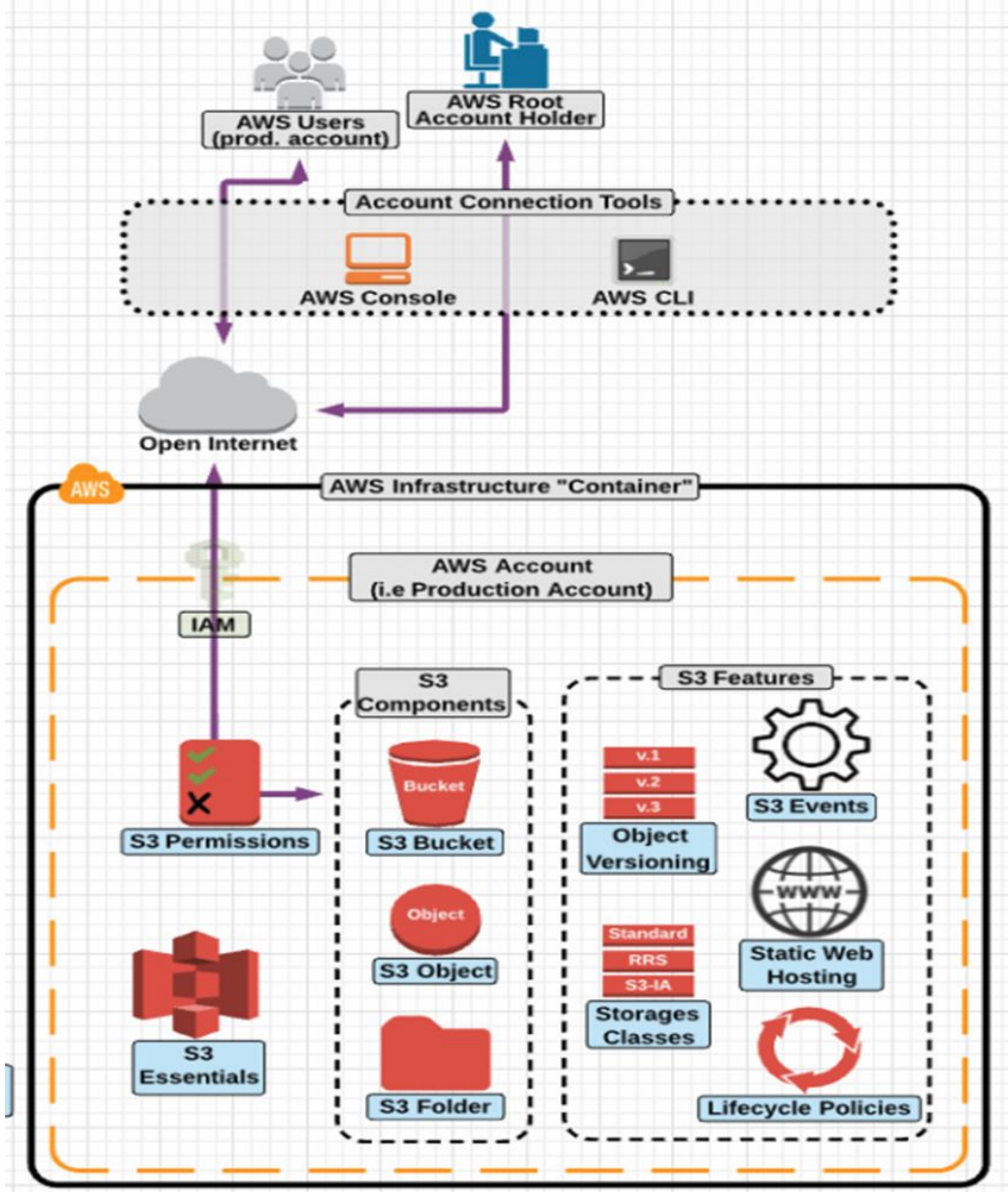
- Bucket names should not contain **UPPER-CASE LETTERS**
- Bucket names **should not contain underscores (_)**
- Bucket names **should not end with a dash**
- Bucket names should be between **3 and 63 characters long**
- Bucket names cannot contain **dashes next to periods** (e.g., "my-.bucket.com" and "my.-bucket" are invalid)

Valid Vs Not valid Bucket Names

- **awsexamplebucket1**
- **log-delivery-march-2020**
- **my-hosted-content**

Before March 1, 2018, buckets created in the US East (N. Virginia) Region could have names that were up to 255 characters long and included uppercase letters and underscores. Beginning March 1, 2018, new buckets in US East (N. Virginia) must conform to the same rules applied in all other Regions.

- Aws_example_bucket (contains underscores)
- AwsExampleBucket (contains uppercase letters)
- aws-example-bucket- (ends with a hyphen)



https://techvinixblog.wordpress.com/2018/07/02/amazon-simple-storage-services3-and-amazon-glacier-storage/

Usage of S3

STORAGE MANAGEMENT AND MONITORING	• **ALL OBJECTS** are stored in S3 • With **S3 VERSIONING** you may store multiple versions of the same within a single bucket. • With **CROSS-REGION REPLICATION**, objects can be replicated into one or more destination S3 buckets hosted in other S3 regions within your AWS account. • Storage can be monitored by applying **TAGS TO S3** to help monitor costs and inventory via the **AWS COST ALLOCATION REPORT, AWS CLOUDTRAIL** for changes to object activities and monitor S3 operational metrics via **AWS CLOUDWATCH**
STORAGE ANALYTICS AND INSIGHTS	• **ACCOUNT SNAPSHOTS** provides an overview of the object storage details and navigating to ***view S3 Storage*** Lens dashboard provides granular details for changing metrics overtime that can be exported via **csv report on a daily basis.** • **S3 Storage Class Analysis** can ***analyze all of the objects*** within a S3 bucket to understand when to transition objects to a different storage class for less frequently used items to a class that is lower cost

Usage of S3

S3 STORAGE CLASSES	**S3 STANDARD**	• The storage class is designed for frequently accessed objects with ***high durability, performance, availability and low latency*** for general purpose uses cases such as big data analytics and static websites
	S3 INTELLIGENT-TIERING	• Intelligent tiering optimizes your storage cost to ***monitor access patterns at the object level.*** • After 30 days of no activity, objects are moved from frequent access tier to infrequent access on an object by object basis to implement cost savings on objects stored in S3. • If the objects are accessed frequently they are moved back to the frequent access tier.
	S3 STANDARD-1A	• S3 Standard-1A is for **less frequently accessed** data that could be accessed quickly when it is required and stored across three availability zones. • The use case is for data accessed greater than 30 days
	S3 ONE ZONE-1A	• S3 One Zone-1A is designed for **infrequently accessed data** that can be accessed quickly but is less durable because it is only stored in a single availability zone

Usage of S3

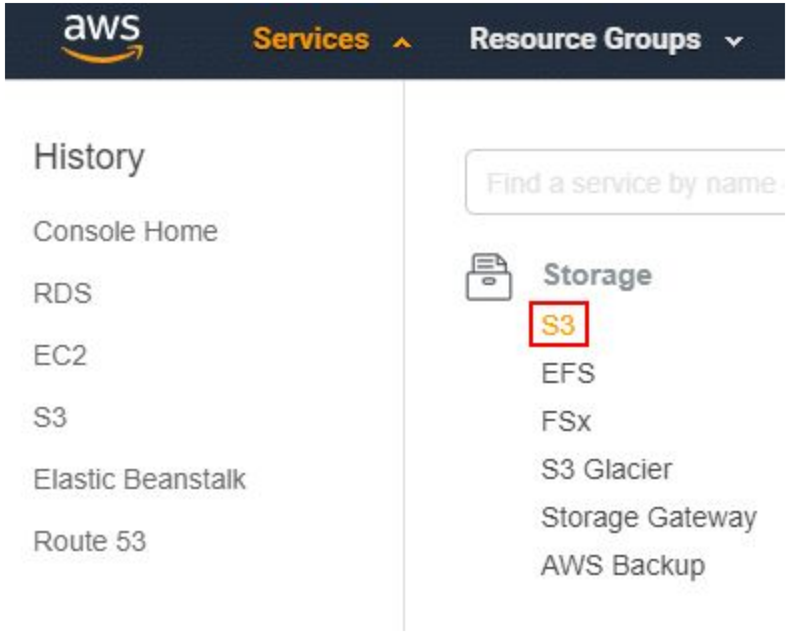
S 3 S T O R A G E C L A S S E S	**S3 Glacier**	• S3 Glacier Storage Classes are for long-term **archival storage** • *Amazon S3 Glacier Instant Retrieval* • *Amazon S3 Glacier Flexible Retrieval (Formerly S3 Glacier)* • *Amazon S3 Glacier Deep Archive* • *Amazon S3 on Outposts*
	Access management and security	• IAM policy which provides granular access to users and groups of users • **ACCESS CONTROL LISTS (ACL**) which provides object access to authorized users • Bucket policy can control anonymous HTTP/HTTPS, encryption and IP address range to access a single S3 bucket
	AWS security	• **Public access** • **Objects can be public** • **Buckets and objects are not public**

Usage of S3

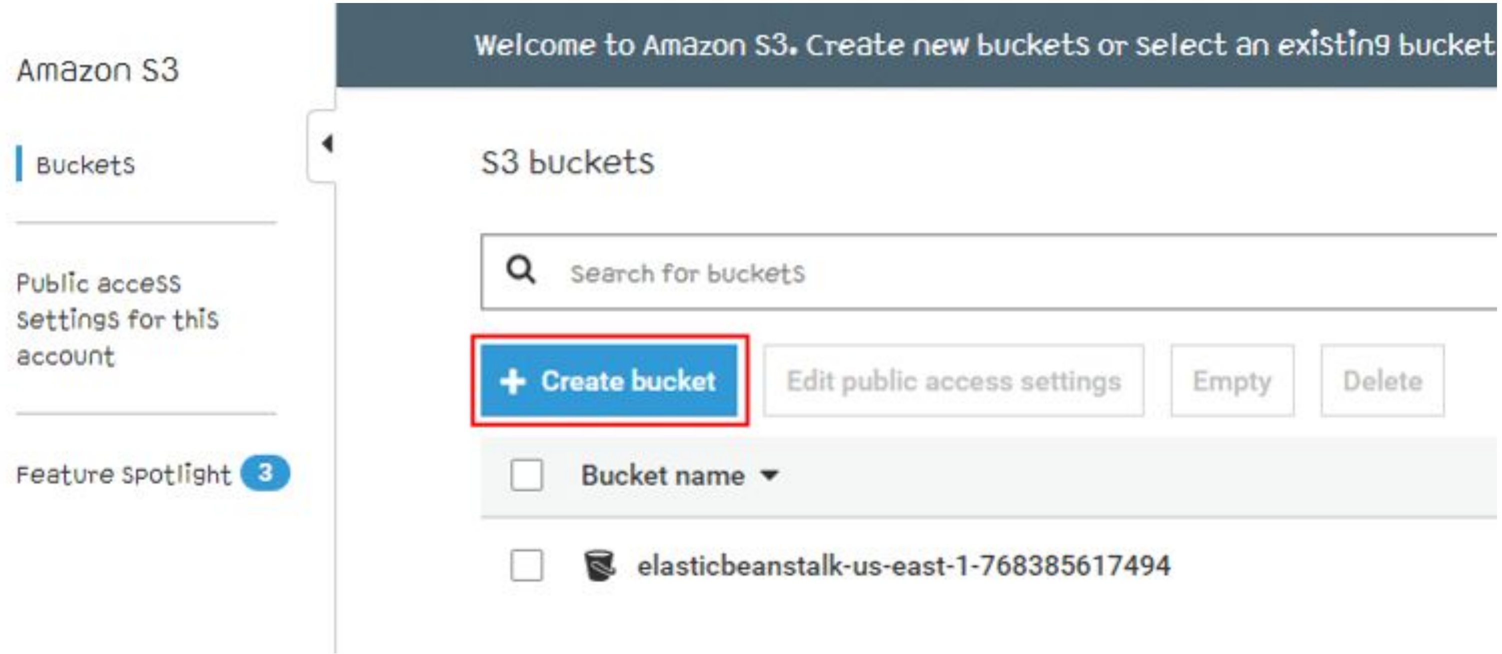
S3 STORAGE CLASSES	**Data processing**	• **S3 Object Lambda** allows you to use your ***customized code to modify the data using AWS Lambda Functions*** to process the output of a returned GET request e.g. redact information
	Querying in one place	• Amazon S3 has an in-built feature to have the ability to query data from **AMAZON ANALYTIC SERVICES** using Amazon Athena and Amazon Redshift Spectrum where SQL queries are made from S3.This removes need to move data onto an analytics platform
	Data transfer	• **Hybrid** cloud storage via **AWS Storage Gateway** • **Off-line** data transfer via **AWS Snow Family** • **Online** data transfer via **AWS DataSync**
	Performance	• Amazon S3 has high performance on cloud storage whereby 5,500 requests can be performed to retrieve data.

General setup for Amazon S3

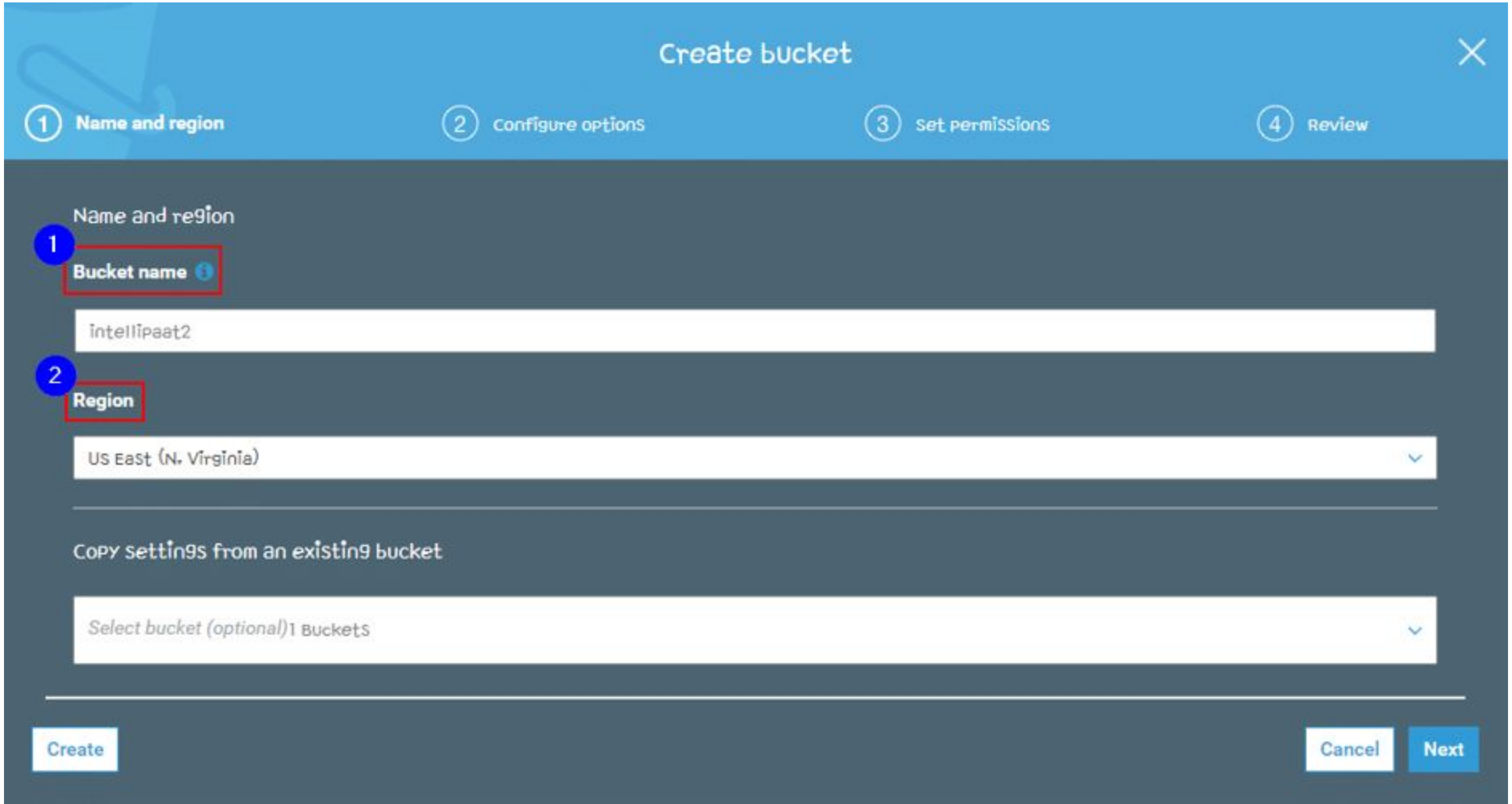
- Step 1: Login to the AWS Management Console
- Step 2: Select S3 from the Services section



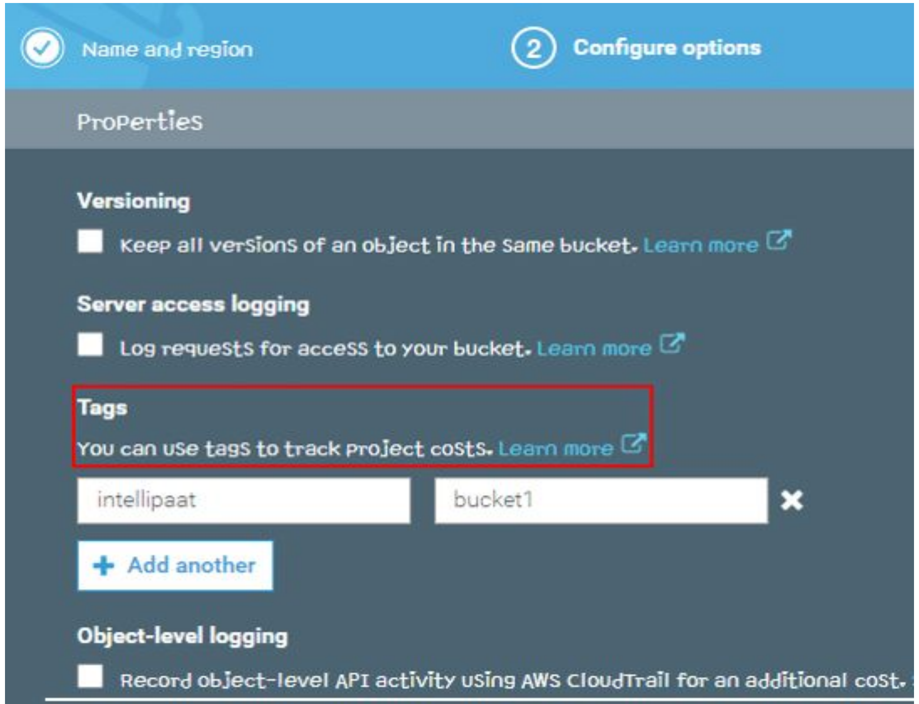
- Step 3: Click on the Create bucket button to start with creating an AWS S3 bucket



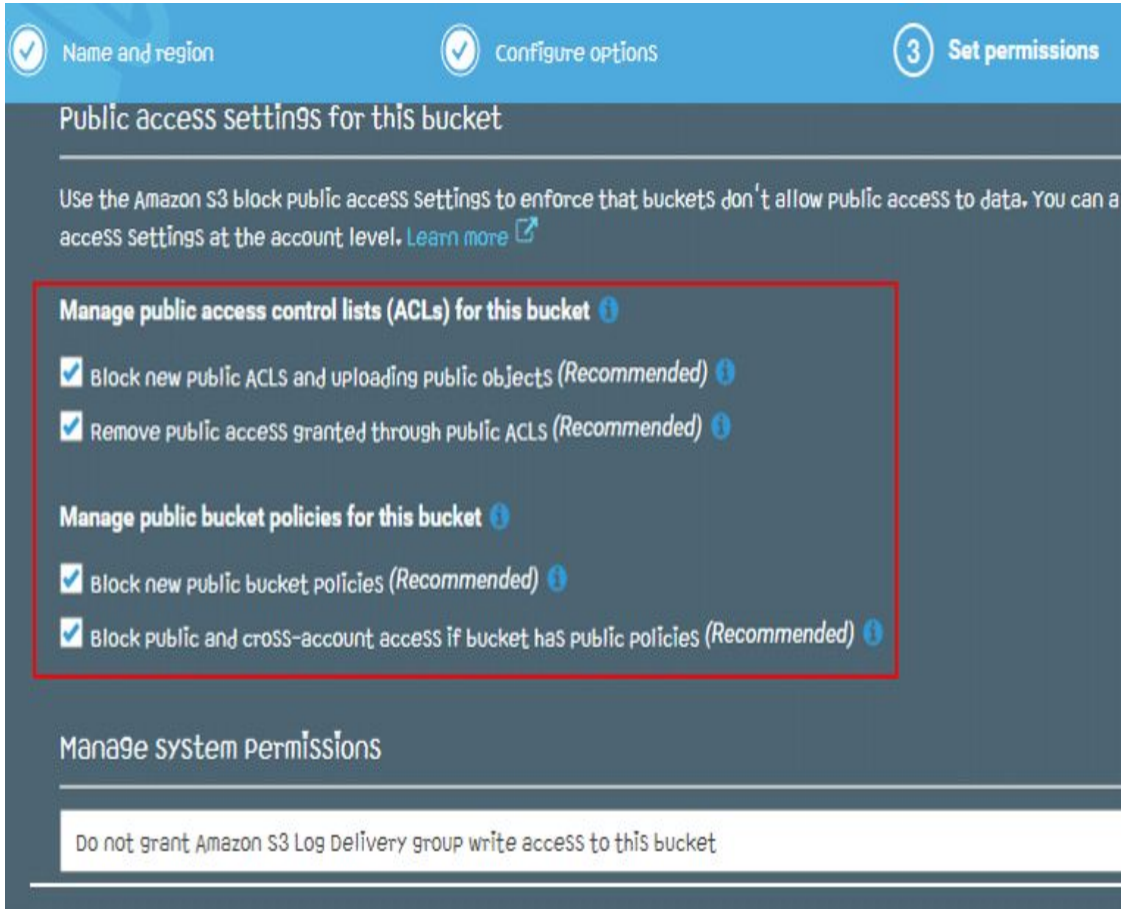
- Step 4: Now, provide a **unique Bucket name** and select the Region in which the bucket should exist. After providing the details click on Next

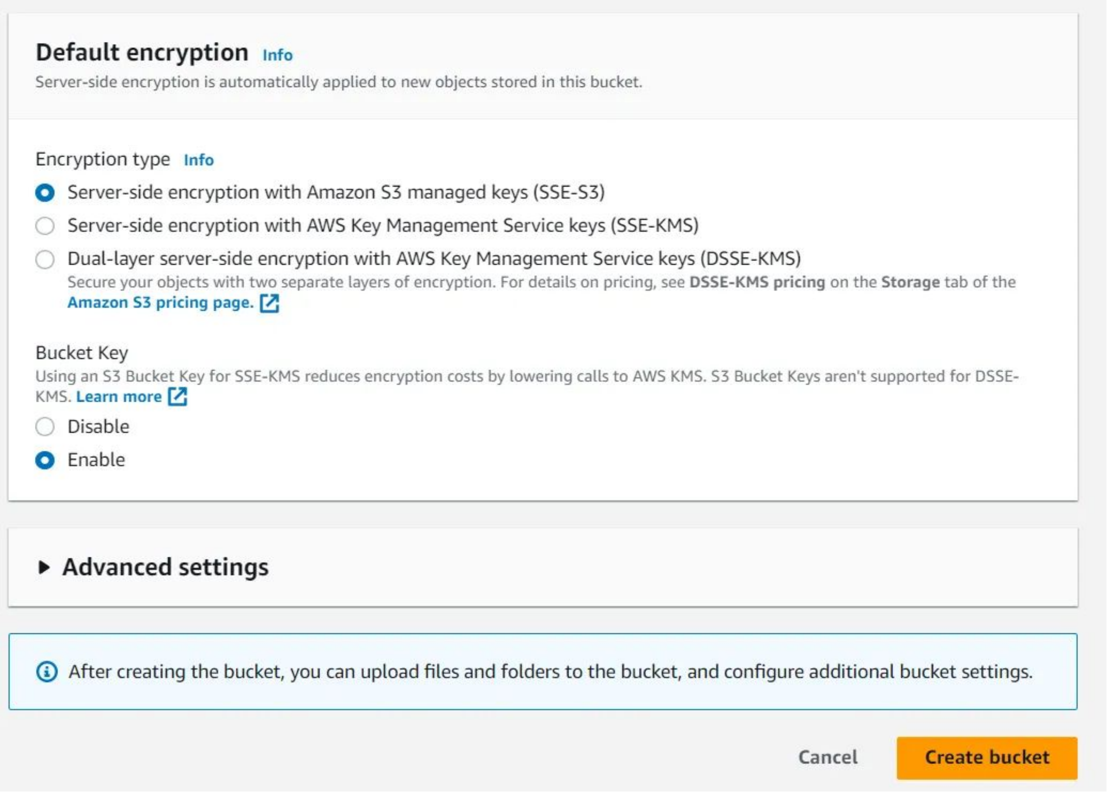


- Step 5: Next is Configure options, and there is no need to provide any details here. You can just click on Next
- But if you want to track costs for this bucket, provide a **tag** to identify it
- Also, you can choose Versioning. Check the Versioning section of this blog to learn what versioning provides

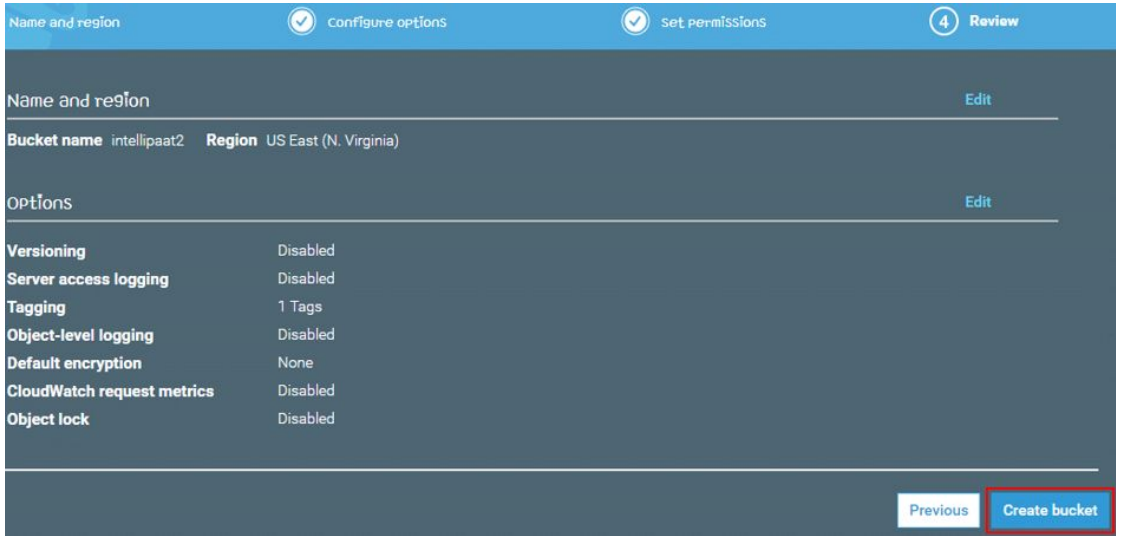


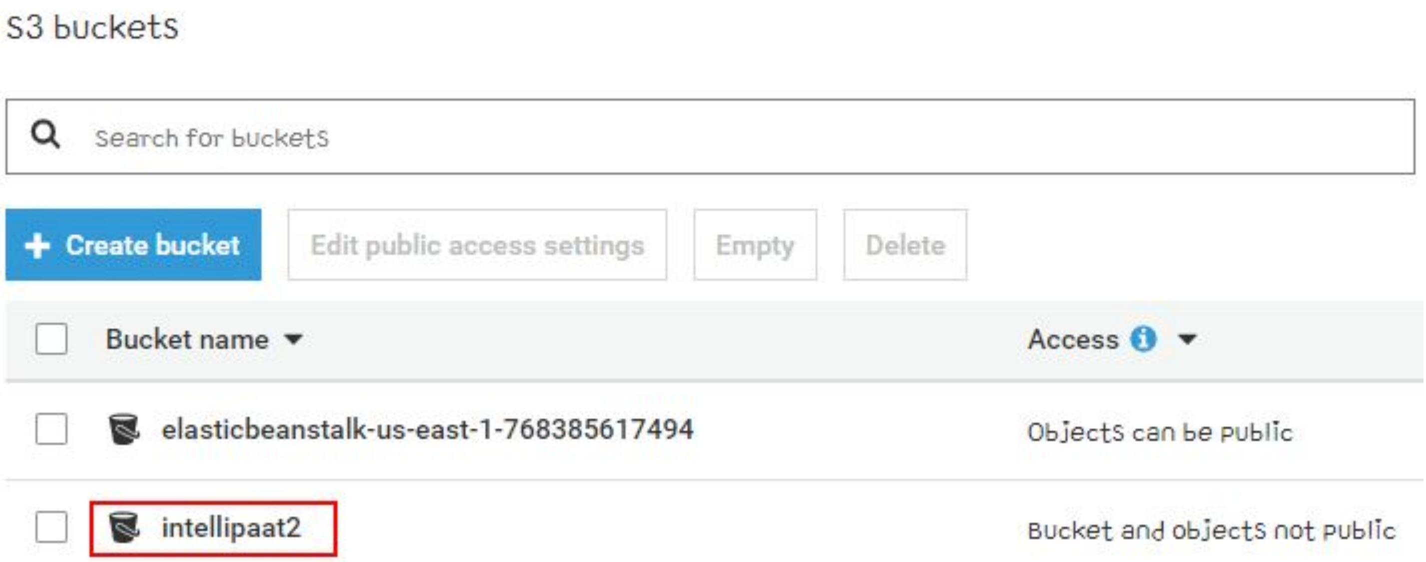
- Step 6: In permissions, keep all checkboxes ticked. This makes the **objects in your bucket inaccessible for the public**





- (Windows instances) For Key pair type, **choose RSA. ED25519 keys are not supported for Windows instances**.
- For Private key file format, choose the format in which to save the private key.
- To save the private key in a format that can be used with OpenSSH, choose pem. To save the private key in a format that can be used with PuTTY, choose ppk.
- Step 7: **Review once and click on** Create bucket





https://intellipaat.com/blog/what-is-amazon-s3/

Create a user

Add users

Access key — Programmatic access field" (essential).

existing policies directly

AdministratorAccess" policy.

CSV file of your user's credentials.

Create a bucket

Create a policy and add it to your user

A policy can be a set of settings or a JSON file attached to an AWS object (**user, resource, group, roles**), and it controls what aspects of the object you can use.

create a policy that enables us to interact with our bucket programmatically — i.e., through the CLI or in a script

Go to the Policies tab and click "Create a policy."
Click the "JSON" tab

```
{
   "Version": "2012-10-17",
   "Statement": [
      {
         "Sid": "ConsoleAccess",  //statement ID
         "Effect": "Allow", //statement results in an allow or an explicit deny
         "Action": [  //actions that will be allowed or denied
            "s3:*"  //interaction to our bucket
         ],
         "Resource": [
            "arn:aws:s3:::your-bucket-name",
             "arn:aws:s3:::your-bucket-name/*"
         ]
      }
   ]
}
```

1. Click the "Add permissions" button.
2. Click the "Attach existing policies" tab.
3. Filter them by the policy we just created.
4. Tick the policy, review it and click "Add"

AWS Key ID and Secret Access Key

//aws configure

Check if authentication is working

```
import boto3  # pip install boto3
# Let's use Amazon S3
s3 = boto3.resource("s3")

# Print out bucket names
for bucket in s3.buckets.all():
    print(bucket.name)
```

sample-bucket-1801

```
import boto3
# Create an S3 access object
s3 = boto3.client("s3") //S3 class using the client function
s3.download_file(
    Bucket="sample-bucket-1801", Key="train.csv",
Filename="data/downloaded_from_s3.csv"
) //download a file from an S3 bucket
s3.upload_file(
    Filename="data/downloaded_from_s3.csv",
    Bucket="sample-bucket-1801",
    Key="new_file.csv",
) //function is upload_file
```

- To create the required S3 bucket, upload our blobs to that bucket, and so forth, all from the amazon web portal.
- Amazon Python Boto3 SDK
- Boto3 considers each service to be a resource

import boto3 s3 = boto3.resource('s3', aws_access_key_id='YOUR ACCESS KEY', aws_secret_access_key='your secret key')	• to have a home directory .aws that • contains two protected files: ***config,containing your default*** • ***Amazon region, and credentials, containing your access and secret keys.*** • If we have this directory in place, then the access key and secret key parameters are not needed.

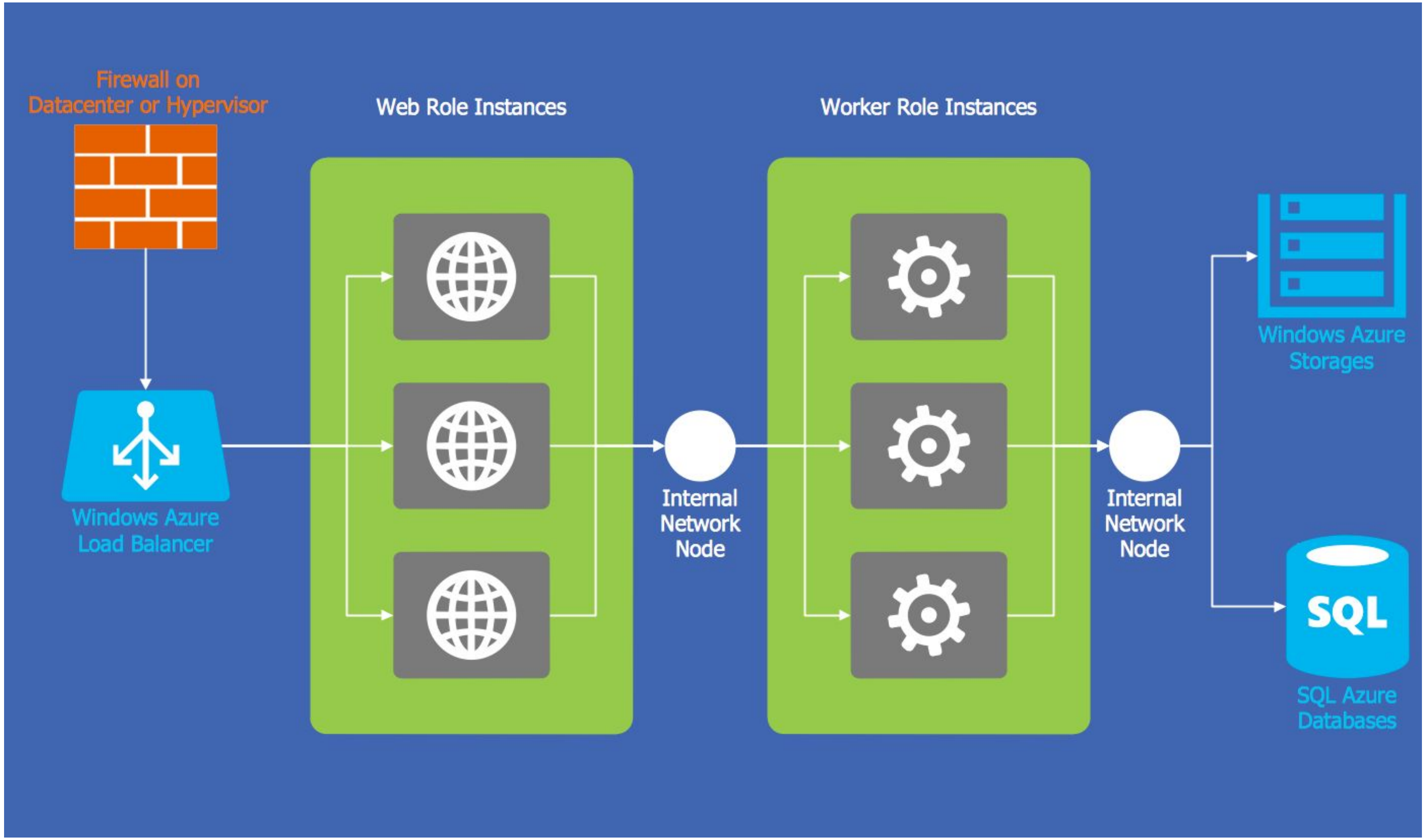
- Created the S3 resource object, we can now create the S3 bucket, datacont, in which we will store our data objects

import boto3 s3 = boto3.resource('s3') s3.create_bucket(Bucket='datacont', CreateBucketConfiguration={ 'LocationConstraint': 'us-west-2'}) # Upload a file, 'test.jpg' into the newly created bucket **s3.Object('datacont', 'test.jpg').put(Body=open('/home/mydata/ test.jpg', 'rb'))**	• DynamoDB table in which we will store metadata and references to S3 objects. • We create this table by defining a special key that is composed of a PartitionKey and a RowKey. • NoSQL systems such as DynamoDB are distributed over multiple storage devices, which enable constructing extremely large tables that can then be accessed in parallel by many servers, each accessing one storage device

```
dyndb = boto3.resource('dynamodb',
region_name='us-west-2' )

# The first time that we define a table, we use
table = dyndb.create_table(
    TableName='DataTable',
    KeySchema=[
        { 'AttributeName': 'PartitionKey',
        'KeyType': 'HASH'},
        { 'AttributeName': 'RowKey', 'KeyType':
        'RANGE' }
    ],
    AttributeDefinitions=[
        { 'AttributeName': 'PartitionKey',
        'AttributeType': 'S' },
```

```
        { 'AttributeName': 'RowKey',
        'AttributeType': 'S' }
    ]
)

# Wait for the table to be created
table.meta.client.get_waiter('table_exists')
                     .wait(TableName='DataTable')

# If the table has been previously defined, use:
# table = dyndb.Table("DataTable")
```

Windows Azure System Design



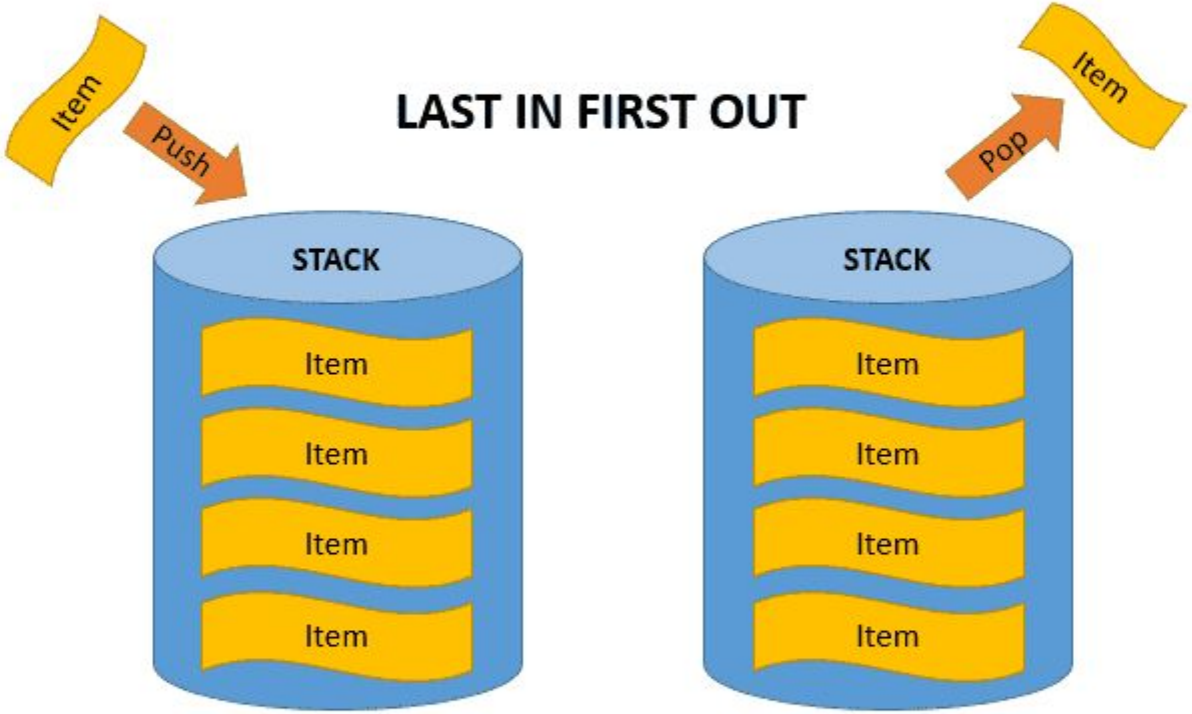
Azure Storage Services

- Accessing services
- Storage account
- Blob storage
- Table storage
- File storage
- Queue storage

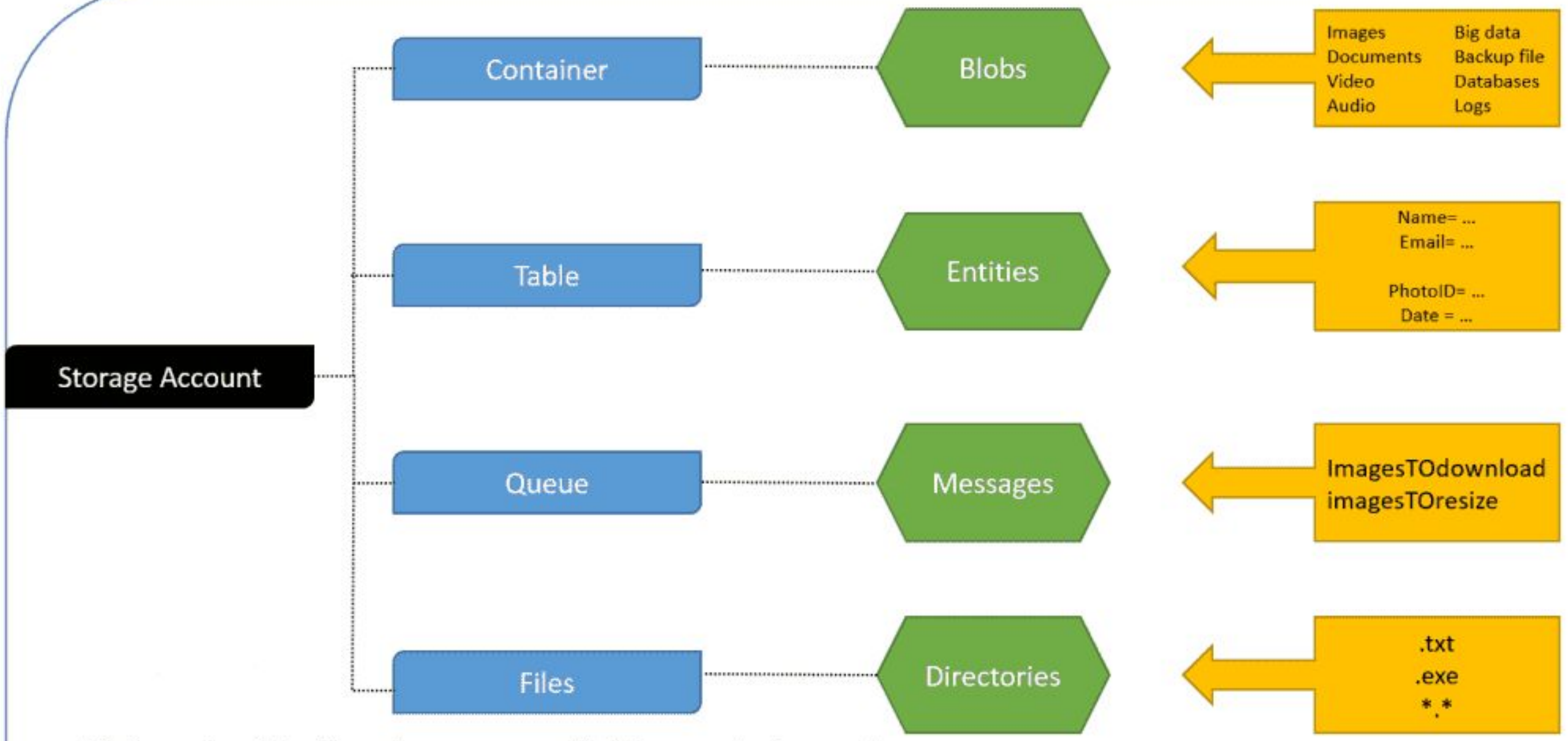
Blob types

- Page blobs are commonly used to store and are designed for random read/write access.
- Block blobs are for everything you can think of as a file.
- Each block in one of these types of blobs can be a different size, up to a maximum of 4 MB, and a block blob can include up to 50,000 blocks
- An append blob is an optimized blob for append operations.
- The main difference between a block blog and an append blog is that when **you modify an append blob**, new **blocks are added to the end block**

Queue data structure

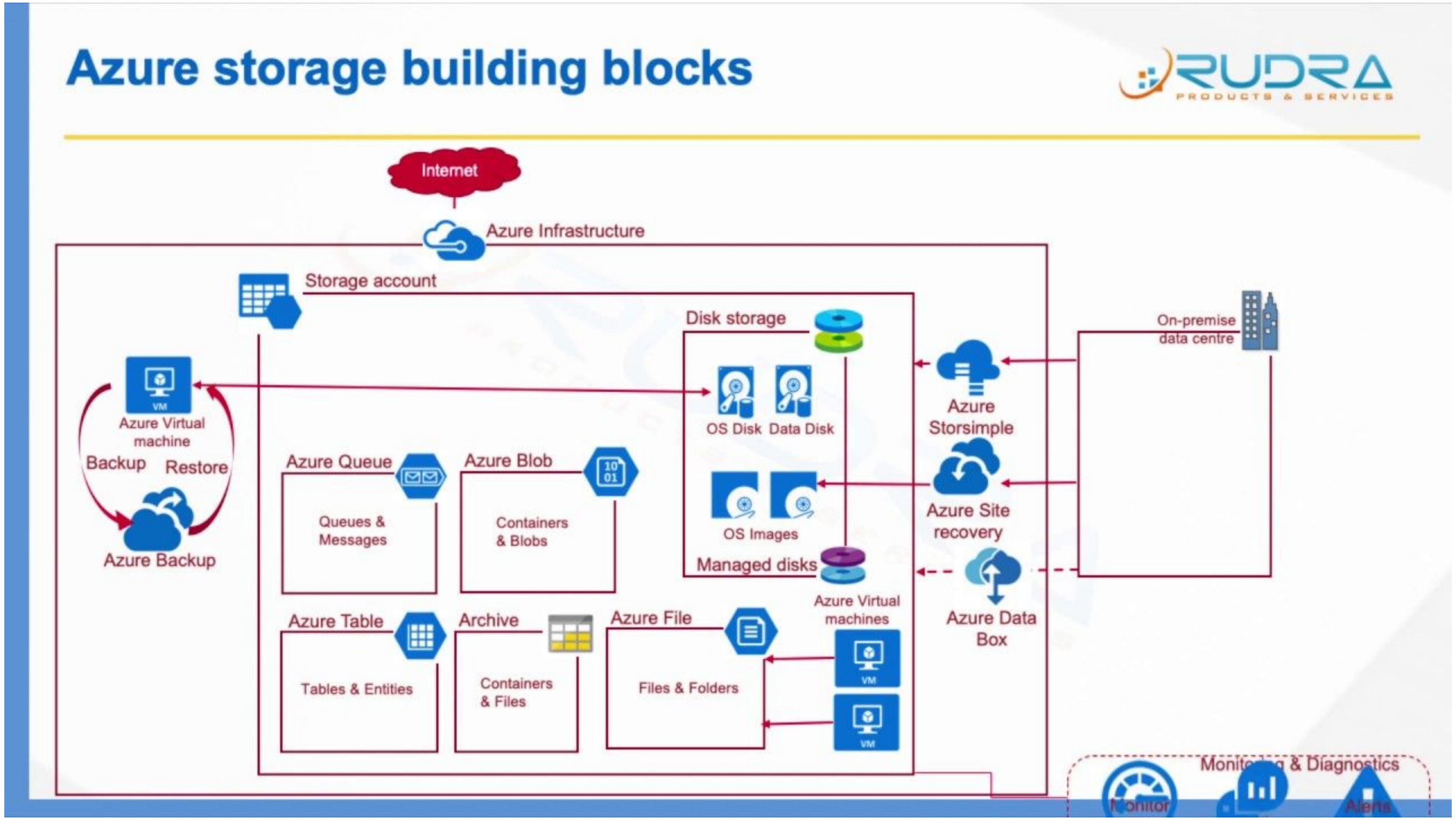


Azure Storage Services – File storage



- *Blob service: http://mystorageaccount.blob.core.windows.net*
- *Table service: http://mystorageaccount.table.core.windows.net*
- *Queue service: http://mystorageaccount.queue.core.windows.net*
- *File service: http://mystorageaccount.file.core.windows.net*

Microsoft Azure Storage



Using Microsoft Azure Storage Services

- Azure account is defined by your **personal ID and a subscription ID.**
- Your personal ID is probably your email address, so that is public
- The **subscription ID** is something to keep secret.
- The differences between Amazon DynamoDB and the Azure Table service are subtle.
- With the Azure Table service, each row has ***the fields PartitionKey, RowKey, comments, date, and URL as before, but this time the RowKey is a unique integer for each row***.
- The PartitionKey is used as **a hash to locate** the row into a specific storage device.
- The RowKey is a **unique global identifier for the row**

- Login and click on storage account in the menu on the left to bring up a panel for storage accounts.
- To add a new account, click on the **+ sign** at the top of the panel.
- You need to supply a ***name and some additional parameters such as LOCATION, DUPLICATION, AND DISTRIBUTION***
- One big difference between S3 and Azure Storage accounts is that each storage account comes with ***two unique access keys, either of which can be used to access and modify the storage account***

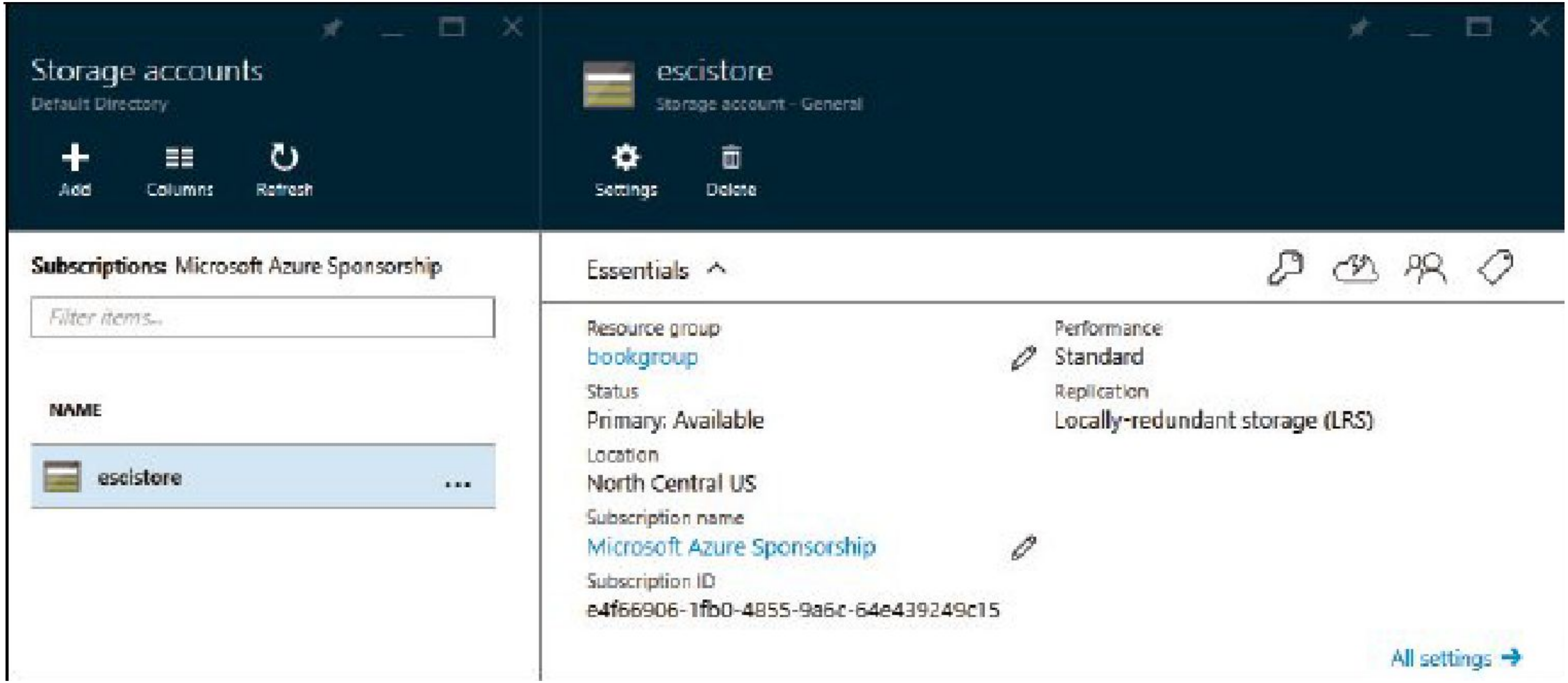
Key Components in WA

Components	Description
FABRIC CONTROLLER	• **Fabrics** are group of machines in Microsoft's datacenter which are aggregated by a switch. • The group of these machines is called **CLUSTER**. • Each cluster is managed and owned by a **fabric controller.**
BLOB	'Blob' expands to **B**inary **L**arge **OB**ject. Blobs include images, text files, videos and audios • **Block blobs** are collection of individual blocks with **unique block ID**. The block blobs allow the users to upload large amount of data. • **Append blobs** are **optimized blocks** that helps in making the operations efficient. • **Page blobs** are compilation of pages. They allow **random read and write operations.** While creating a blob, if the type is not specified they are set to block type by default.
QUEUES	• Queue is a data structure used to store data which follows **First in-First out rule**. • A data item can be inserted from back of the queue while it is retrieved from front. • Azure queues are a very similar concept that is used to store the messages in a queue. • A sender sends the message and a client receives and processes them. • A message has few attributes attached to it, for example expiry time.

Components	Description

Tables	• Storing a table does not mean **RELATIONAL DATABASE** here.
	• Azure Storage can store just a table **without any foreign keys or any other kind of relation.**
	• These tables are highly scalable and ideal for handling large amount of data.
	• Tables can be stored and queried for large amount of data.
	• The relational database can be stored using **SQL Data Services, which is a separate service.**
	    •Tables
	    •Entities
	    •Properties

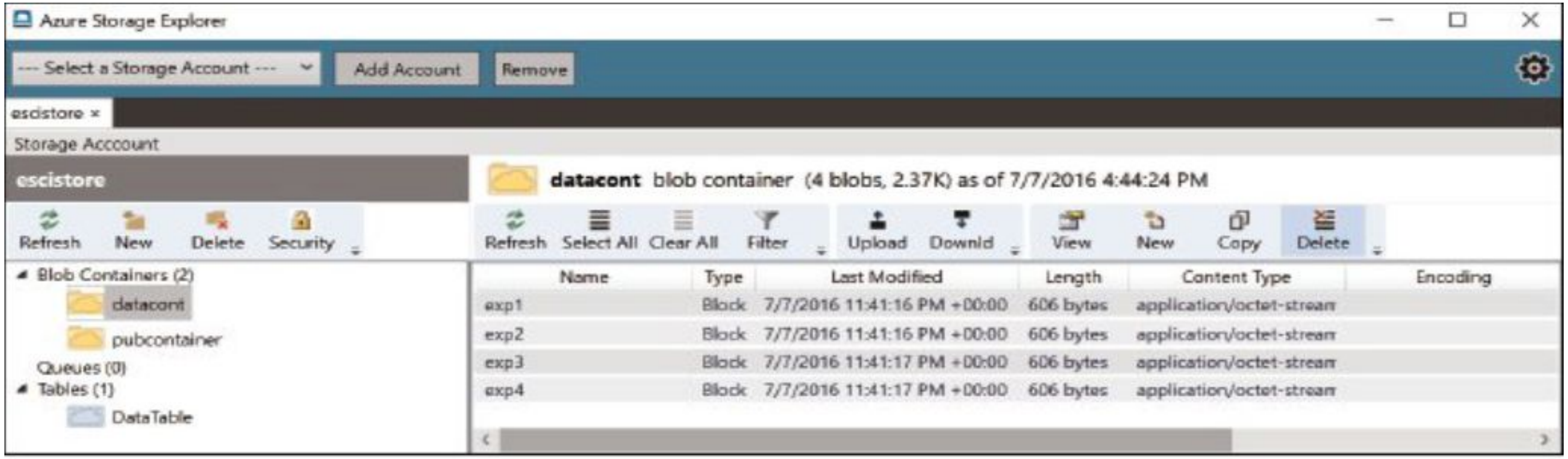
CDN	• Caching is one of the ways for **PERFORMANCE IMPROVEMENT**. • Windows Azure uses caching to increase the speed of cloud services. • **CONTENT DELIVERY NETWORK (CDN)** puts stuff like blobs and other static content in a cache. • The process involves placing the data at strategically chosen locations and caching it. • As a result, it provides maximum bandwidth for its delivery to users.

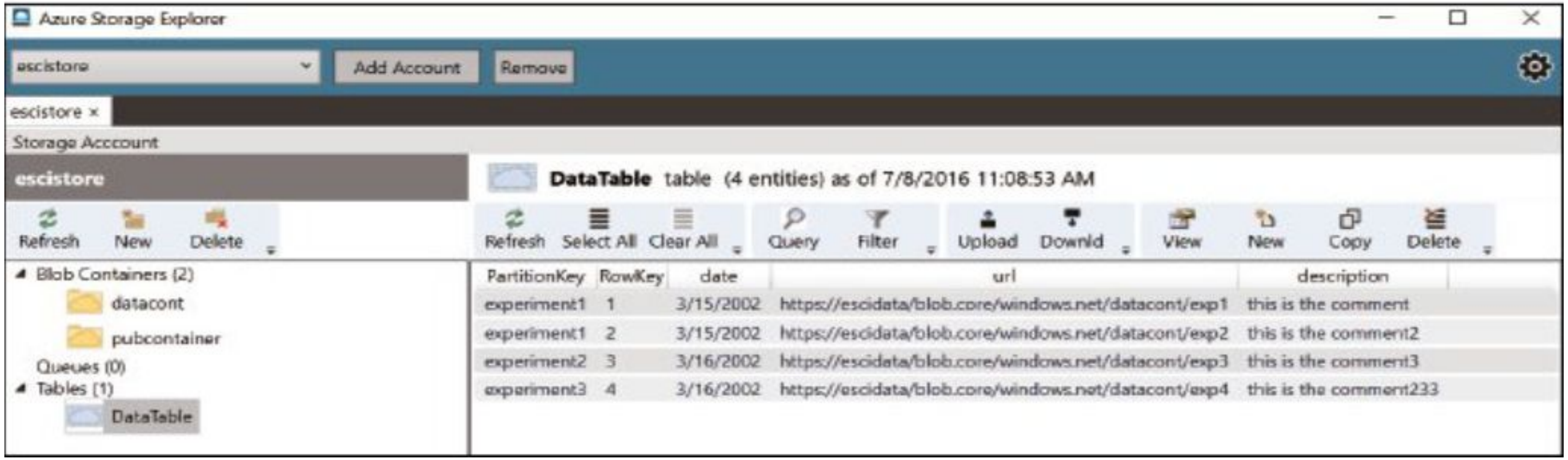

Storage accounts
Default Directory
Add
Columns
Refresh
Subscriptions: Microsoft Azure Sponsorship
Filter items...
NAME
escistore
escistore
Storage account - General
Settings
Delete
Essentials
Resource group
bookgroup
Status
Primary: Available
Location
North Central US
Subscription name
Microsoft Azure Sponsorship
Subscription ID
e4f66906-1fb0-4855-9a6c-64e439249c15
Performance
Standard
Replication
Locally-redundant storage (LRS)
All settings

```
import azure.storage
from azure.storage.table import TableService,
Entity
from azure.storage.blob import BlockBlobService
from azure.storage.blob import PublicAccess
# First, access the blob service
block_blob_service =
BlockBlobService(account_name='escistore',
    account_key='your storage key')
block_blob_service.create_container('datacont',
    public_access=PublicAccess.Container)
# Next, create the table in the same storage
account
table_service =
```

```
                              account_key='your account
                              key')
if table_service.create_table('DataTable'):
    print("Table created")
else:
    print("Table already there")
```

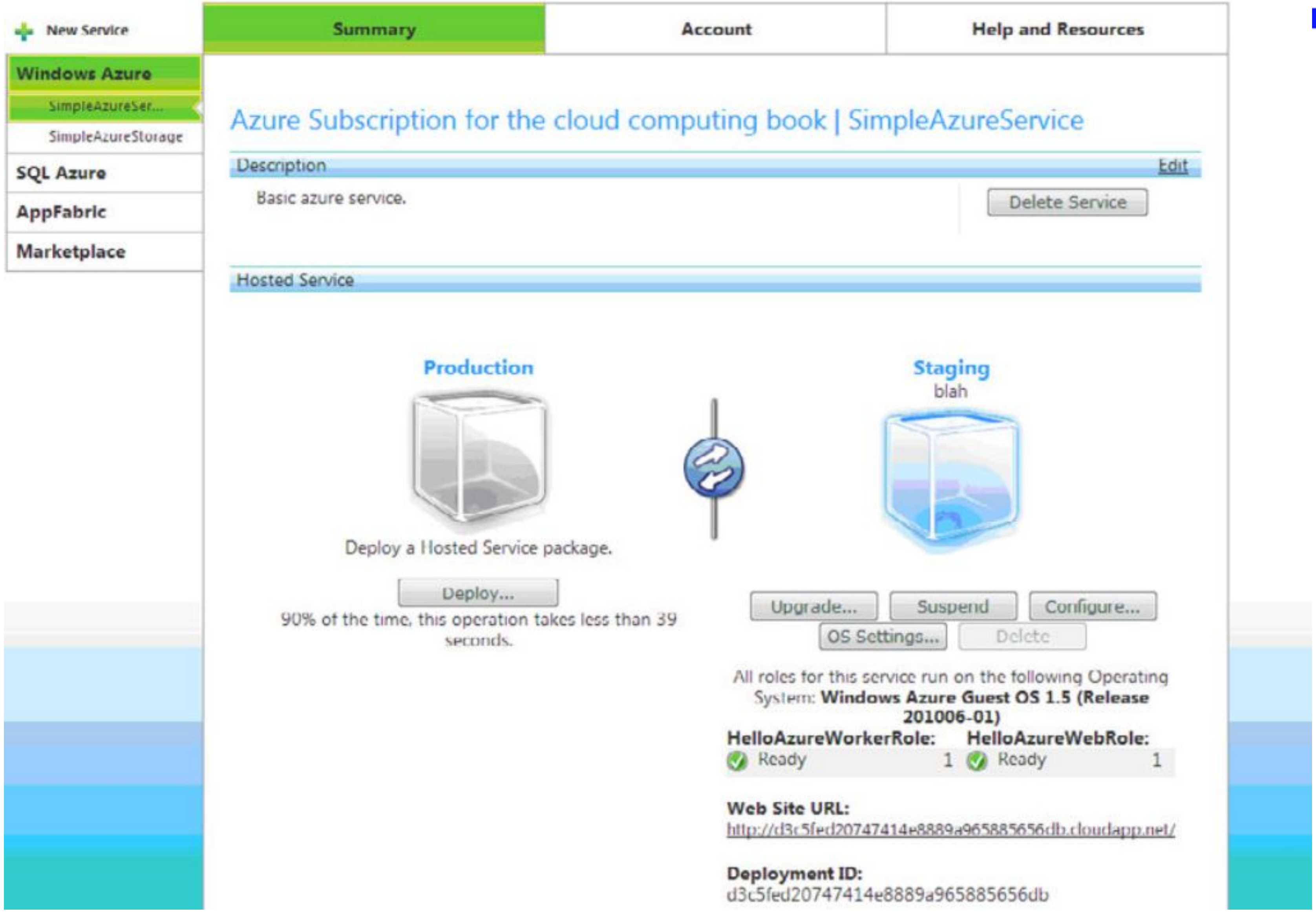
```
import csv
with open('\path-to-your-data\experiments.csv',
'rb') as csvfile:
    csvf = csv.reader(csvfile, delimiter=',',
    quotechar='|')
    for item in csvf:
      print(item)
      block_blob_service.create_blob_from_path(
           'datacont', item[3],
           "\path-to-your-
           files\datafiles\\"+item[3]
           )
    url="https://escistore.blob.core.windows.net/d
    atacont/'+item[3]
    metadata_item = {'PartitionKey':item[0],
    'RowKey':item[1],
         'description' : item[4], 'date' : item[2],
         'url':url}
    table_service.insert_entity('DataTable',
    metadata_item)
```


Azure Storage Explorer
--- Select a Storage Account ---
Add Account
Remove
escistore ×
Storage Account
escistore
Refresh New Delete Security
Blob Containers (2)
datacont
pubcontainer
Queues (0)
Tables (1)
DataTable
datacont blob container (4 blobs, 2.37K) as of 7/7/2016 4:44:24 PM
Refresh Select All Clear All Filter Upload Downld View New Copy Delete
Name Type Last Modified Length Content Type Encoding
exp1 Block 7/7/2016 11:41:16 PM +00:00 606 bytes application/octet-stream
exp2 Block 7/7/2016 11:41:16 PM +00:00 606 bytes application/octet-stream
exp3 Block 7/7/2016 11:41:17 PM +00:00 606 bytes application/octet-stream
exp4 Block 7/7/2016 11:41:17 PM +00:00 606 bytes application/octet-stream


Azure Storage Explorer
escistore
Add Account
Remove
escistore ×
Storage Account
escistore
Refresh New Delete
Blob Containers (2)
datacont
pubcontainer
Queues (0)
Tables (1)
DataTable
DataTable table (4 entities) as of 7/8/2016 11:08:53 AM
Refresh Select All Clear All Query Filter Upload Downld View New Copy Delete
PartitionKey RowKey date url description
experiment1 1 3/15/2002 https://escidata/blob.core/windows.net/datacont/exp1 this is the comment
experiment1 2 3/15/2002 https://escidata/blob.core/windows.net/datacont/exp2 this is the comment2
experiment2 3 3/16/2002 https://escidata/blob.core/windows.net/datacont/exp3 this is the comment3
experiment3 4 3/16/2002 https://escidata/blob.core/windows.net/datacont/exp4 this is the comment233

Key Features	Description
'Replication' dropdown	• **Locally redundant storage:** Copy of the data is created in the same region where storage account is created. • There are **3 COPIES** of each request made against the data that resides on separate domains. • **Zone-redundant storage (available for blobs only):** Copy of the data is created on separate facilities either in the same region or across two regions. • **Three copies** of data are created • **Geo-redundant storage:** Copy is created in a different region which means data is retained even if there is a failure in the complete region. • The numbers of copies of data created are 6 in this case. • **Read-access geo-redundant storage:** This option allows reading of data from a secondary location when data on the primary location is not available. **The number of copies created is 6**

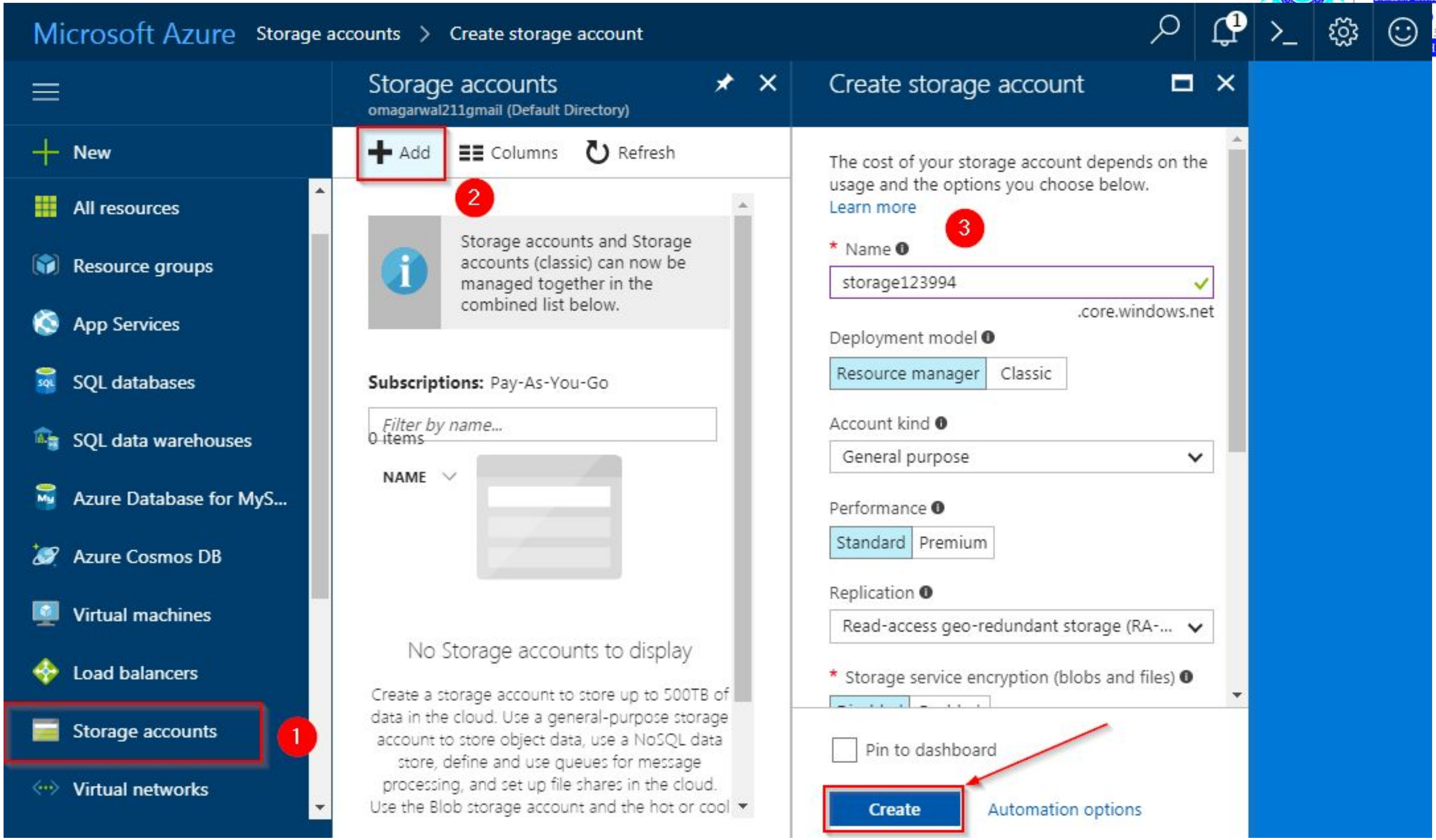
Key Features	Description
Generating an Access Key	• **Azure Storage Service Encryption (SSE)** can automatically encrypt data before it is stored, and it automatically decrypts the data when you retrieve it. • The process is completely transparent to users. • Storage Service Encryption uses **256-bit Advanced Encryption Standard (AES) encryption** • Role-based access control (RBAC) • Cross-origin resource sharing (CORS) • **Auditing access** • **Azure Monitoring**
Managing Data to Azure Storage	Message Processing in Storage Queues 1. *CreateIfNotExist()* 2. *GetQueueReference method()* 3. *ADD A MESSAGE()* 4. *getmessage()* 5. *peekmessage()* 6. *Resultlabel* 7. *OnStart()* 8. *Run()*


New Service
Summary
Account
Help and Resources
Windows Azure
SimpleAzureSer...
SimpleAzureStorage
SQL Azure
AppFabric
Marketplace
Azure Subscription for the cloud computing book | SimpleAzureService
Description
Edit
Basic azure service.
Delete Service
Hosted Service
Production
Deploy a Hosted Service package.
Deploy...
90% of the time, this operation takes less than 39 seconds.
Staging
blah
Upgrade...
Suspend
Configure...
OS Settings...
Delete
All roles for this service run on the following Operating System: Windows Azure Guest OS 1.5 (Release 201006-01)
HelloAzureWorkerRole:
Ready 1
HelloAzureWebRole:
Ready 1
Web Site URL:
http://d3c5fed20747414e8889a965885656db.cloudapp.net/
Deployment ID:
d3c5fed20747414e8889a965885656db

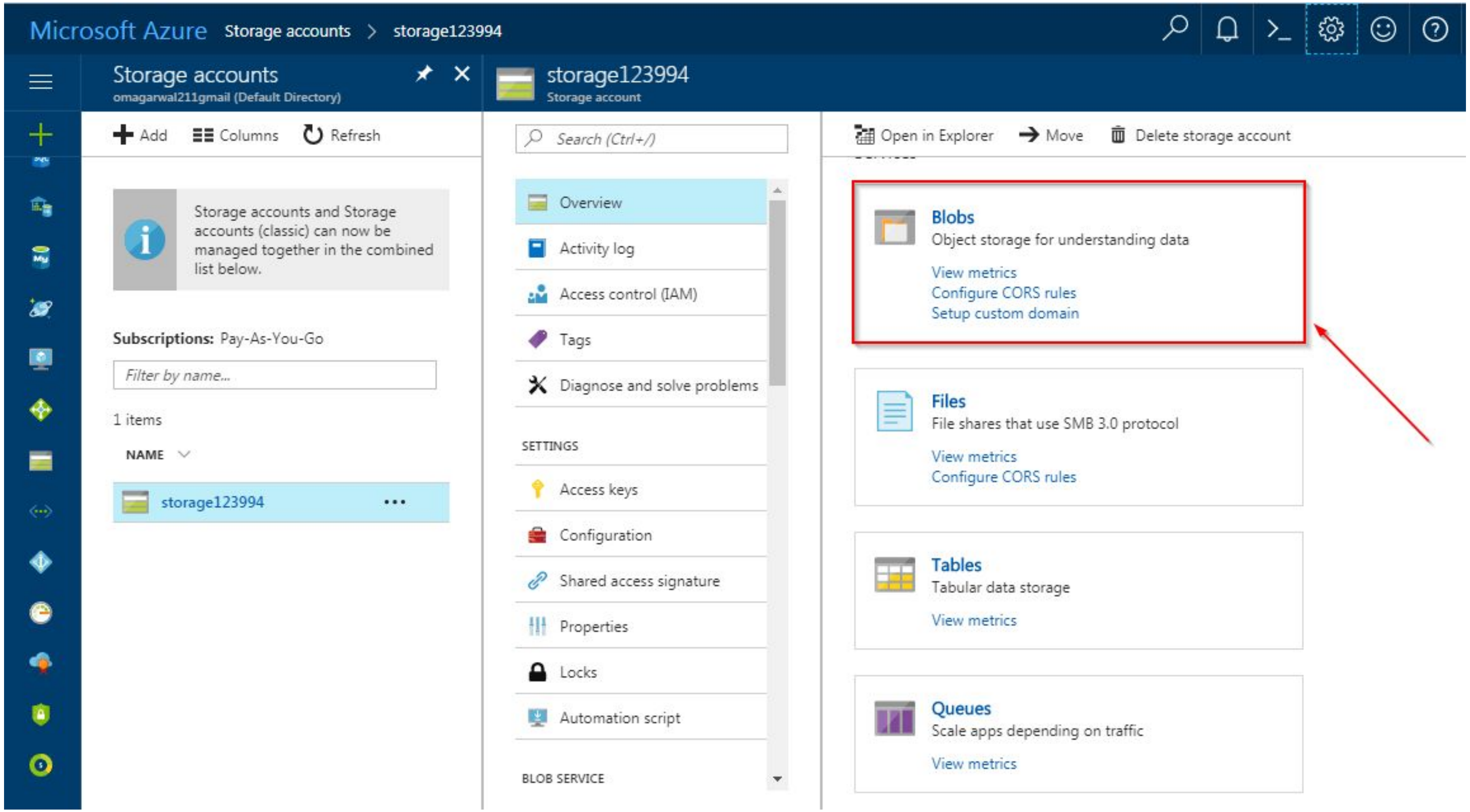
General setup for Microsoft Azure Storage Services(SCENARIO)

- Part 1: setting up a website which will be able to upload files to the blob service.
- Once a file is uploaded, the file's details will also be added to the Azure queue, which will be used to change the background of the webpage when refreshed.
- Step 1: The first step should be creating your **Storage Account**.

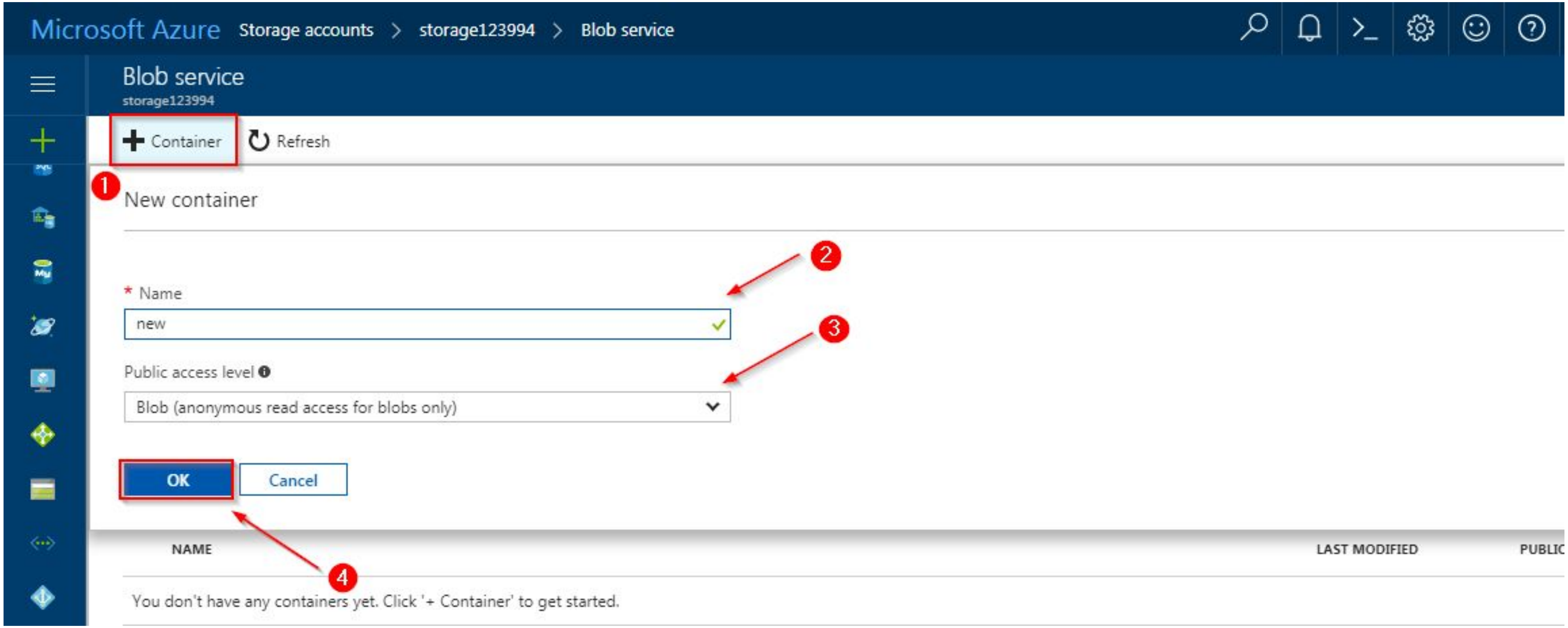
- First, in the left pane click on Storage Accounts
- Then, click on Add
- Finally, enter all the relevant fields and Click on Create


Microsoft Azure
Storage accounts
Create storage account
New
All resources
Resource groups
App Services
SQL databases
SQL data warehouses
Azure Database for MyS...
Azure Cosmos DB
Virtual machines
Load balancers
Storage accounts
Virtual networks
1
Storage accounts
omagarwal211gmail (Default Directory)
Add
Columns
Refresh
2
Storage accounts and Storage accounts (classic) can now be managed together in the combined list below.
Subscriptions: Pay-As-You-Go
Filter by name...
0 items
NAME
No Storage accounts to display
Create a storage account to store up to 500TB of data in the cloud. Use a general-purpose storage account to store object data, use a NoSQL data store, define and use queues for message processing, and set up file shares in the cloud. Use the Blob storage account and the hot or cool
Create storage account
The cost of your storage account depends on the usage and the options you choose below.
Learn more
3
* Name
storage123994
.core.windows.net
Deployment model
Resource manager
Classic
Account kind
General purpose
Performance
Standard
Premium
Replication
Read-access geo-redundant storage (RA-...
* Storage service encryption (blobs and files)
Pin to dashboard
Create
Automation options

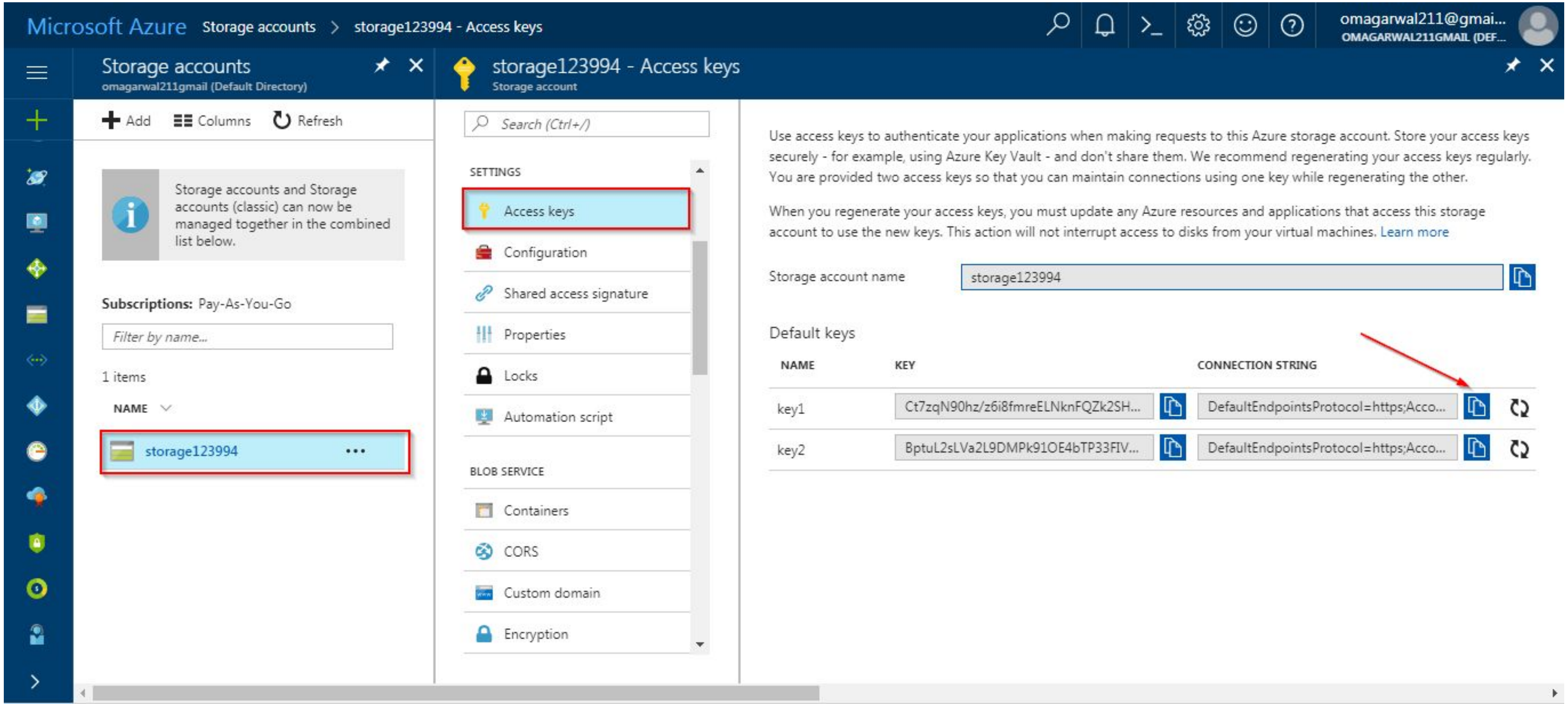
- Step 2:There are four types of storage services in our account, i.e **Blobs, Queues, Files and Tables.**



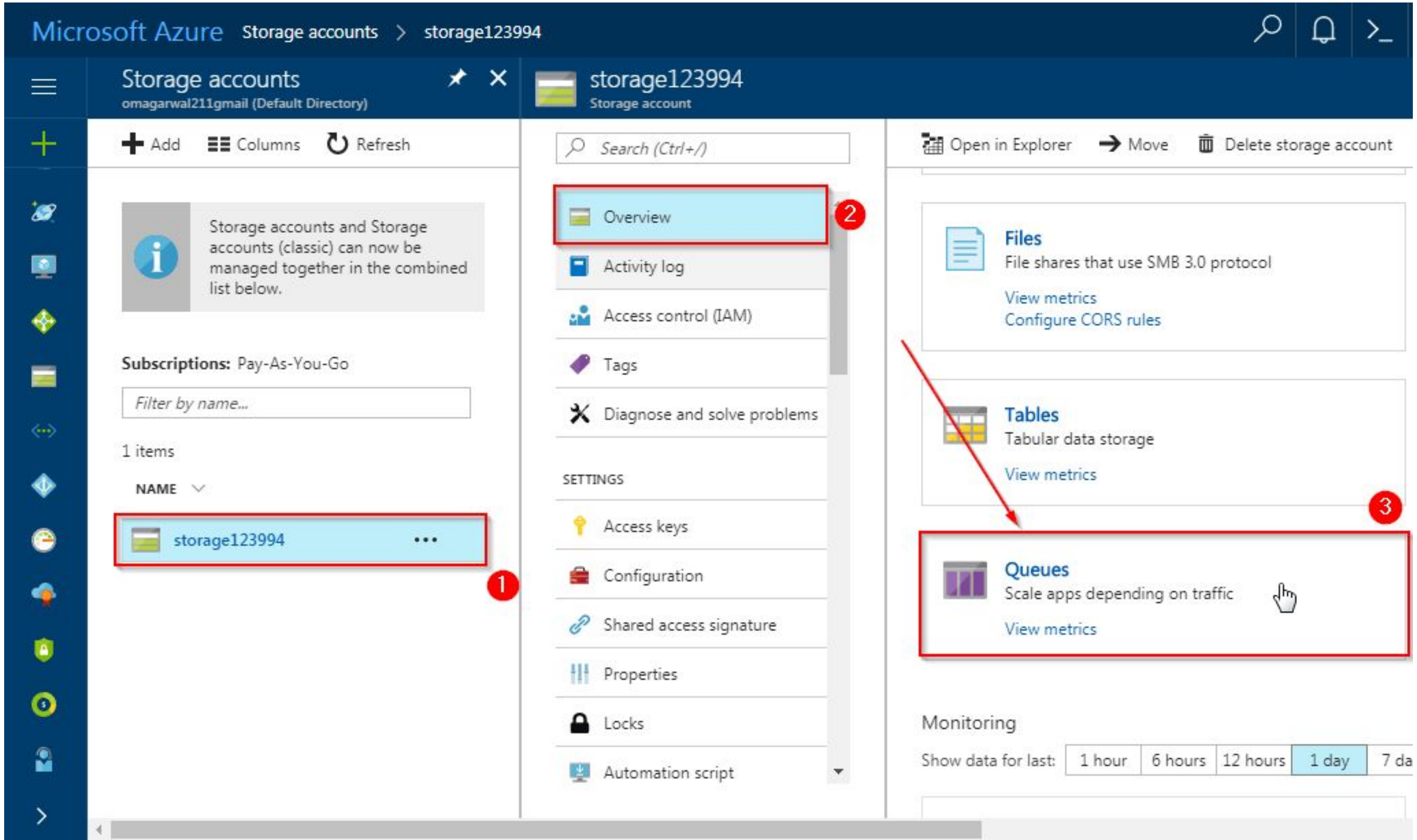
- Step 3: Click on container, **to create a new container**.
  - First, enter the **name of the container**
  - This should be **unique to all the containers** that you will be creating in this particular account.
  - Assign **public access level** to it.
  - Blobs are nothing but files.


Microsoft Azure
Storage accounts > storage123994 > Blob service
Blob service
storage123994
Container
Refresh
1
New container
2
* Name
new
3
Public access level
Blob (anonymous read access for blobs only)
OK
Cancel
4
NAME
LAST MODIFIED
PUBLIC
You don't have any containers yet. Click '+ Container' to get started.

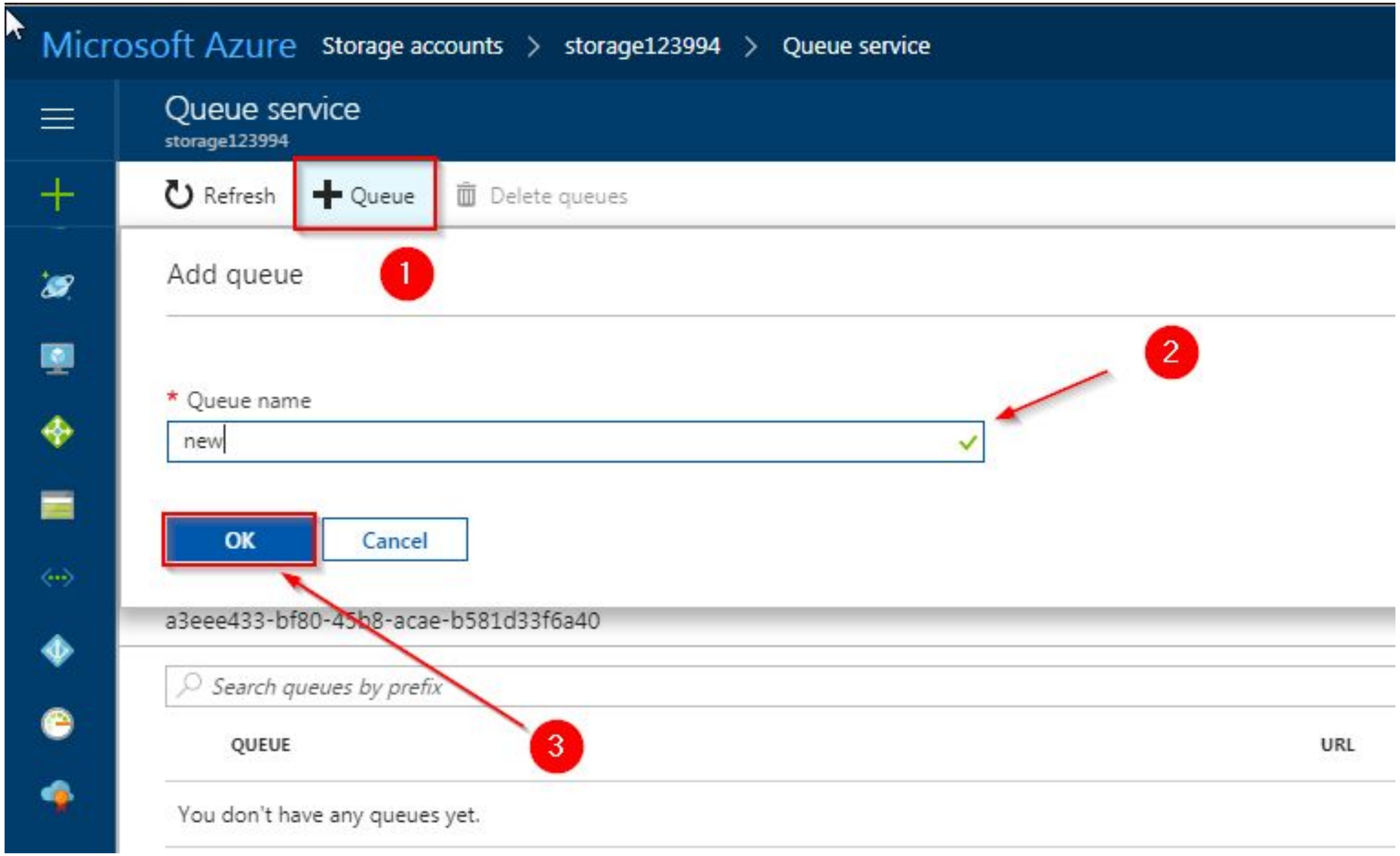
- **Step 4:** Specify the connection string of your storage account in your website's code.
- A connection string authenticates your code to interact with the specified storage account and its services.



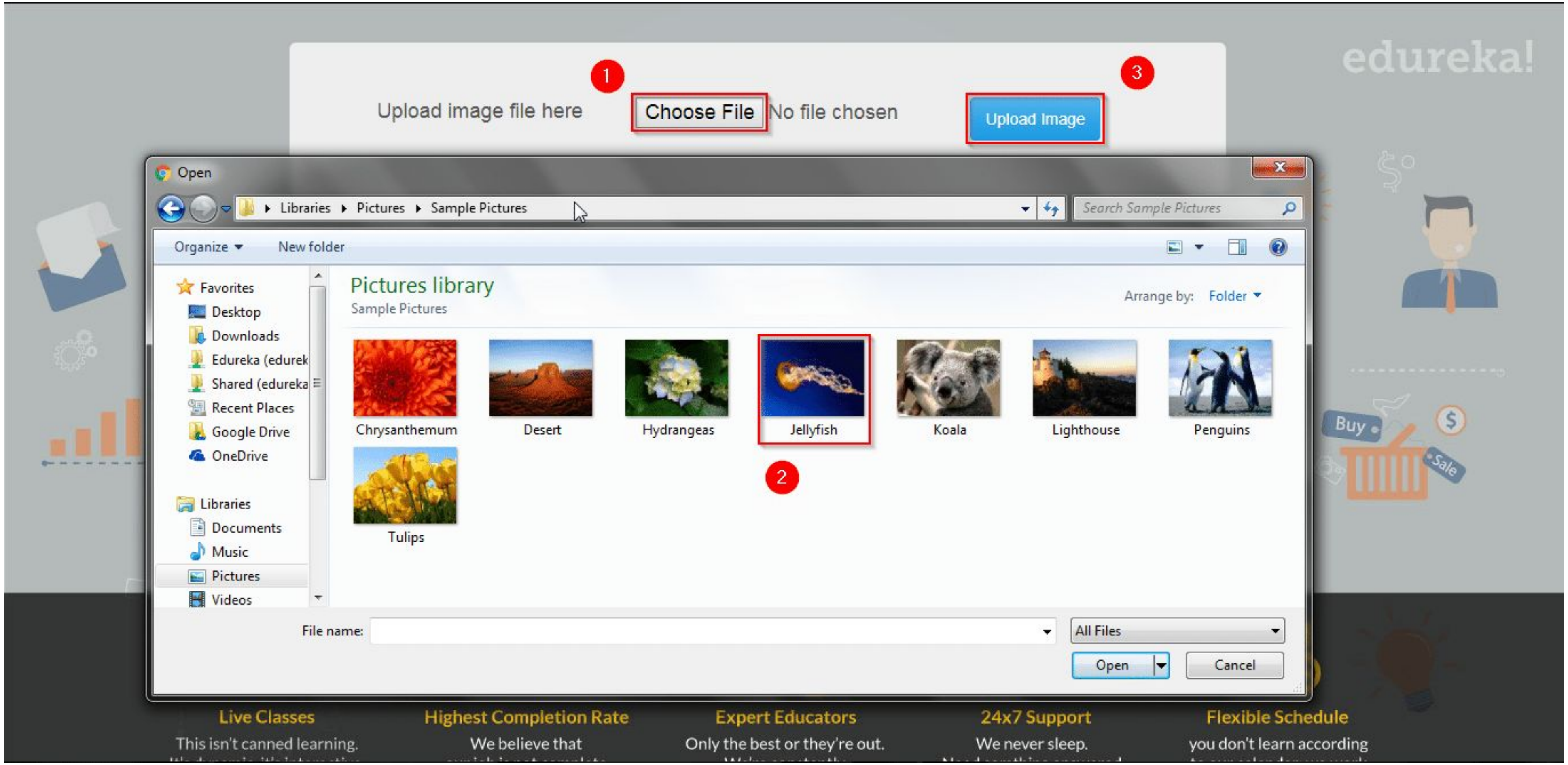
- Step 5: Let's start with the queue now.
- On your storage accounts overview page select queues

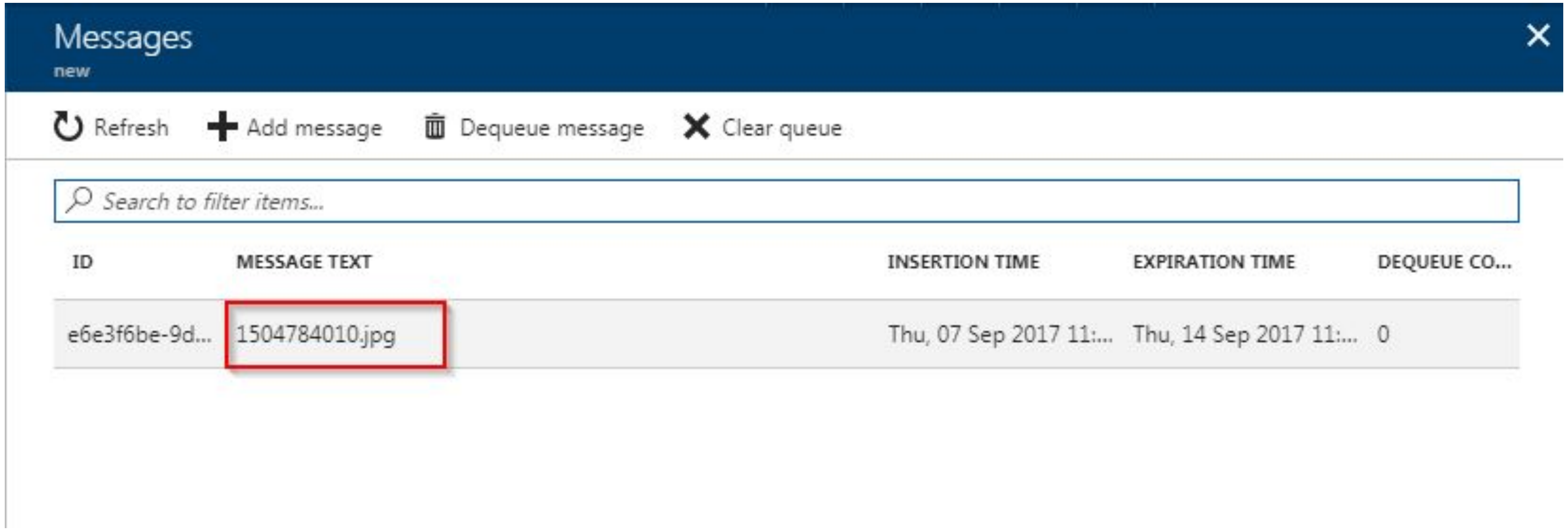


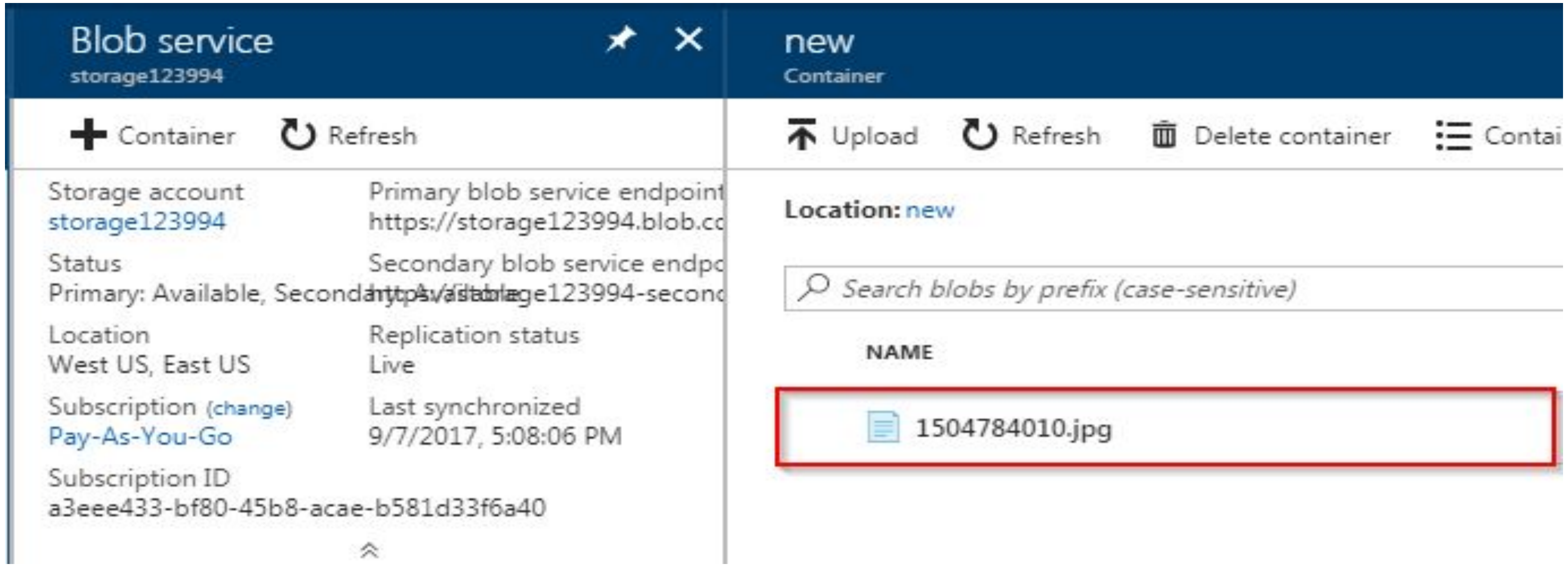
- Step 6: Next, we'll be creating a queue. To do that, Click on Add Queue, give a relevant name to the queue and click on



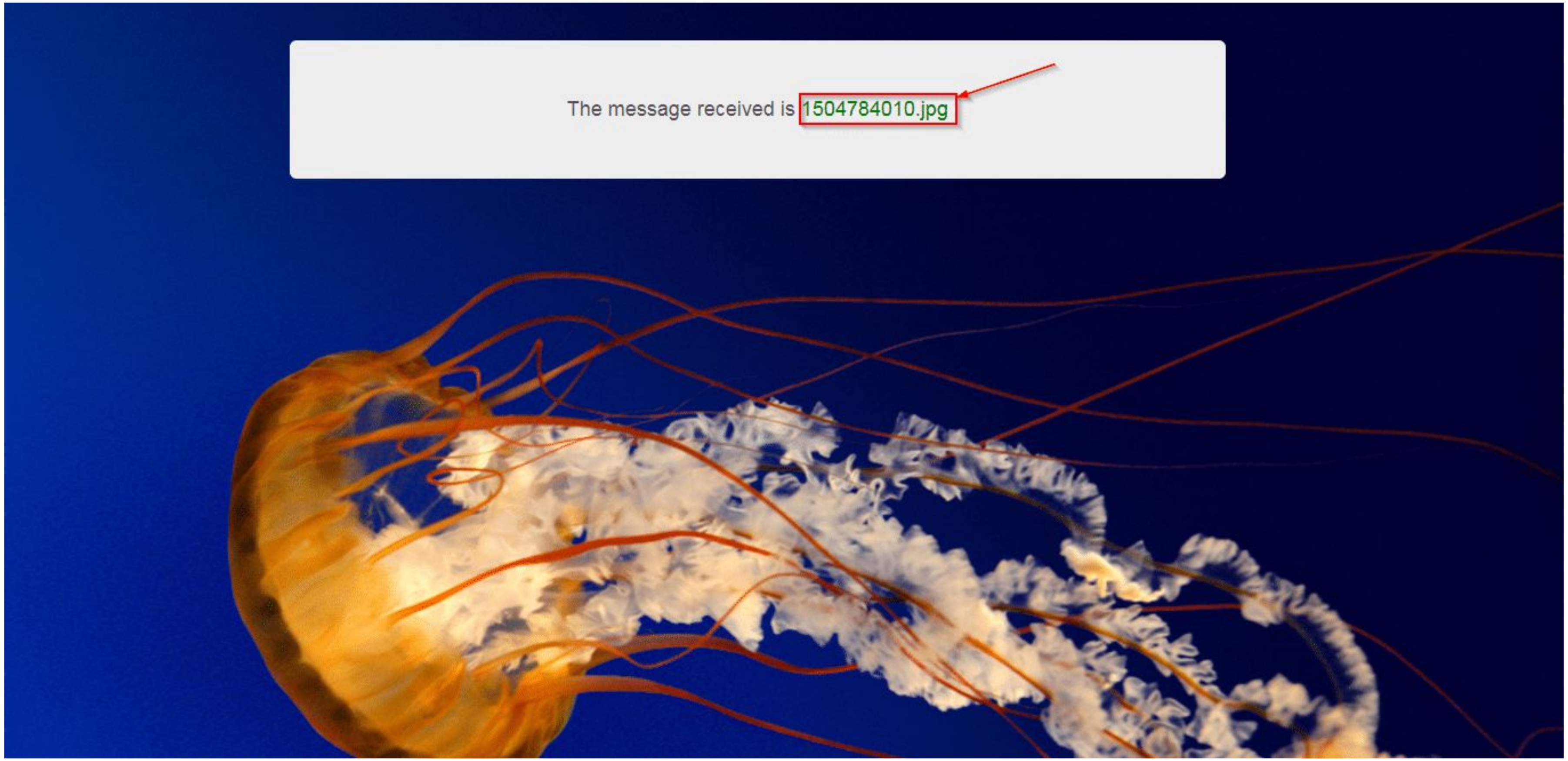
- Step 7: This is the website we have made, select the file that you want to upload, and click on upload.




Messages
new
Refresh
Add message
Dequeue message
Clear queue
Search to filter items...
ID
MESSAGE TEXT
INSERTION TIME
EXPIRATION TIME
DEQUEUE CO...
e6e3f6be-9d...
1504784010.jpg
Thu, 07 Sep 2017 11:...
Thu, 14 Sep 2017 11:...
0


Blob service
storage123994
Container
Refresh
Storage account
storage123994
Primary blob service endpoint
https://storage123994.blob.cc
Status
Primary: Available, Secondary: Available
Secondary blob service endpc
https://storage123994-seconc
Location
West US, East US
Replication status
Live
Subscription (change)
Pay-As-You-Go
Last synchronized
9/7/2017, 5:08:06 PM
Subscription ID
a3eee433-bf80-45b8-acae-b581d33f6a40
new
Container
Upload
Refresh
Delete container
Contai
Location: new
Search blobs by prefix (case-sensitive)
NAME
1504784010.jpg

- Step 8: Let's go to our process page in the website to see if the entry from the queue and blob can be read

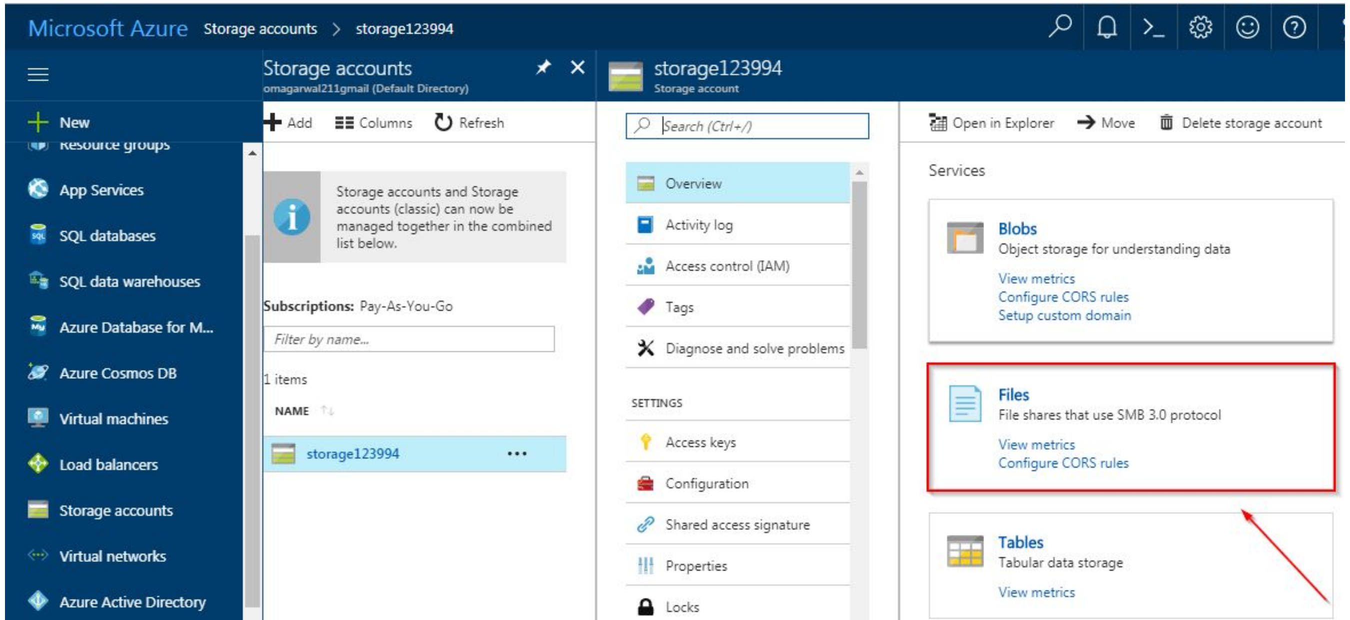


https://www.edureka.co/blog/azure-storage-tutorial/

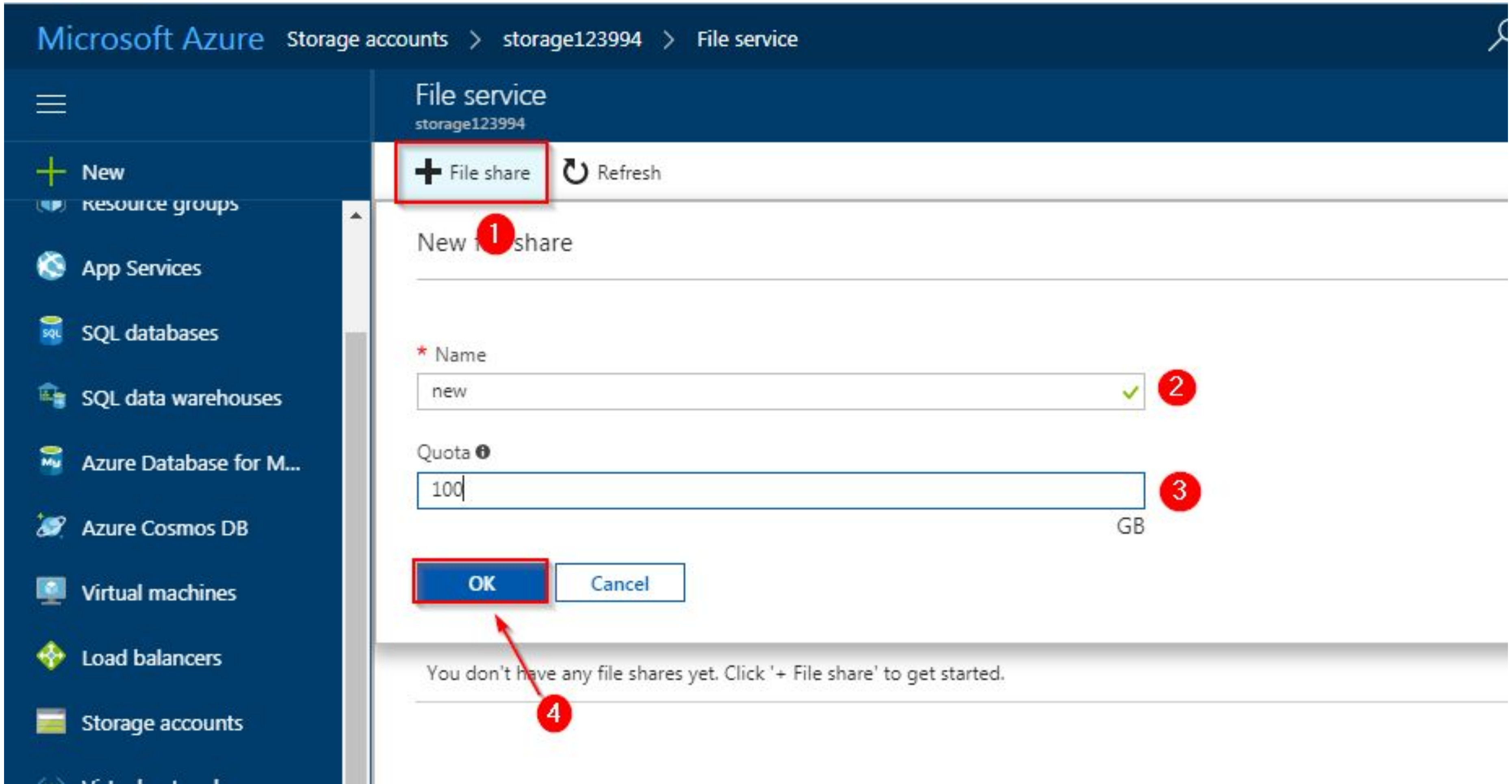
- Part 2: The file service in azure.

(ONLY FOR LEARNING PURPOSE)

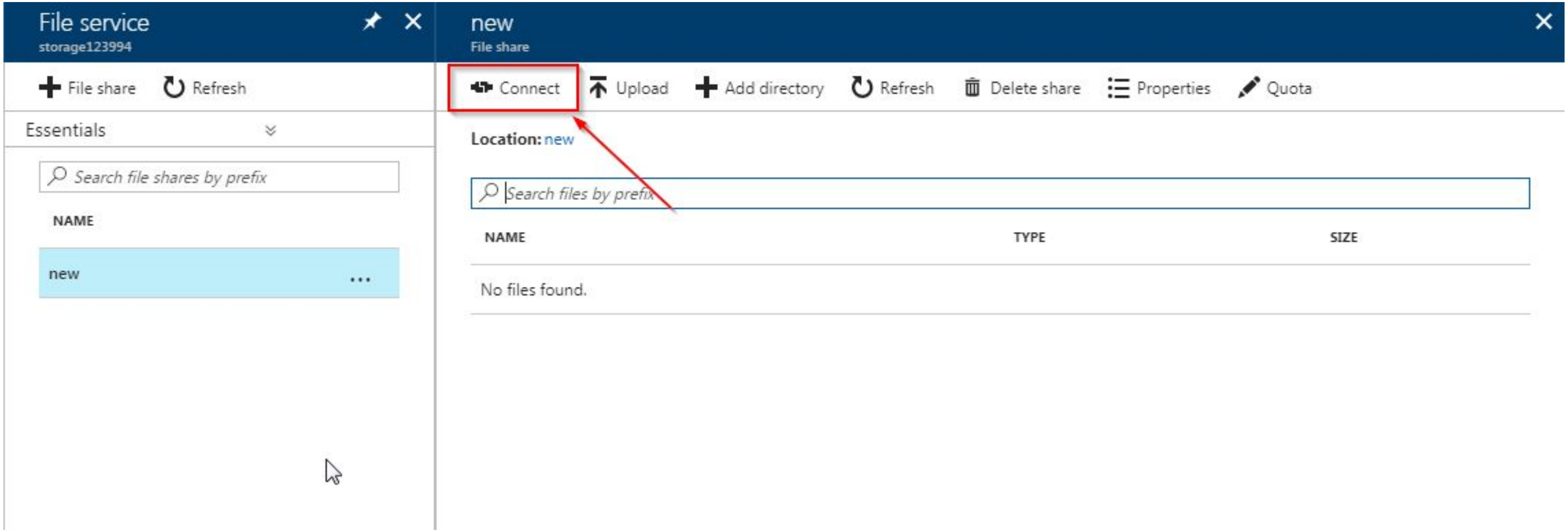
- The File Service in Azure uses the SMB (**Server Message Block)3.0** protocol for file transfers, this service can be attached to your windows OS as if it was an external drive.
- Step 1: Go to your storage account overview page, and select the file service

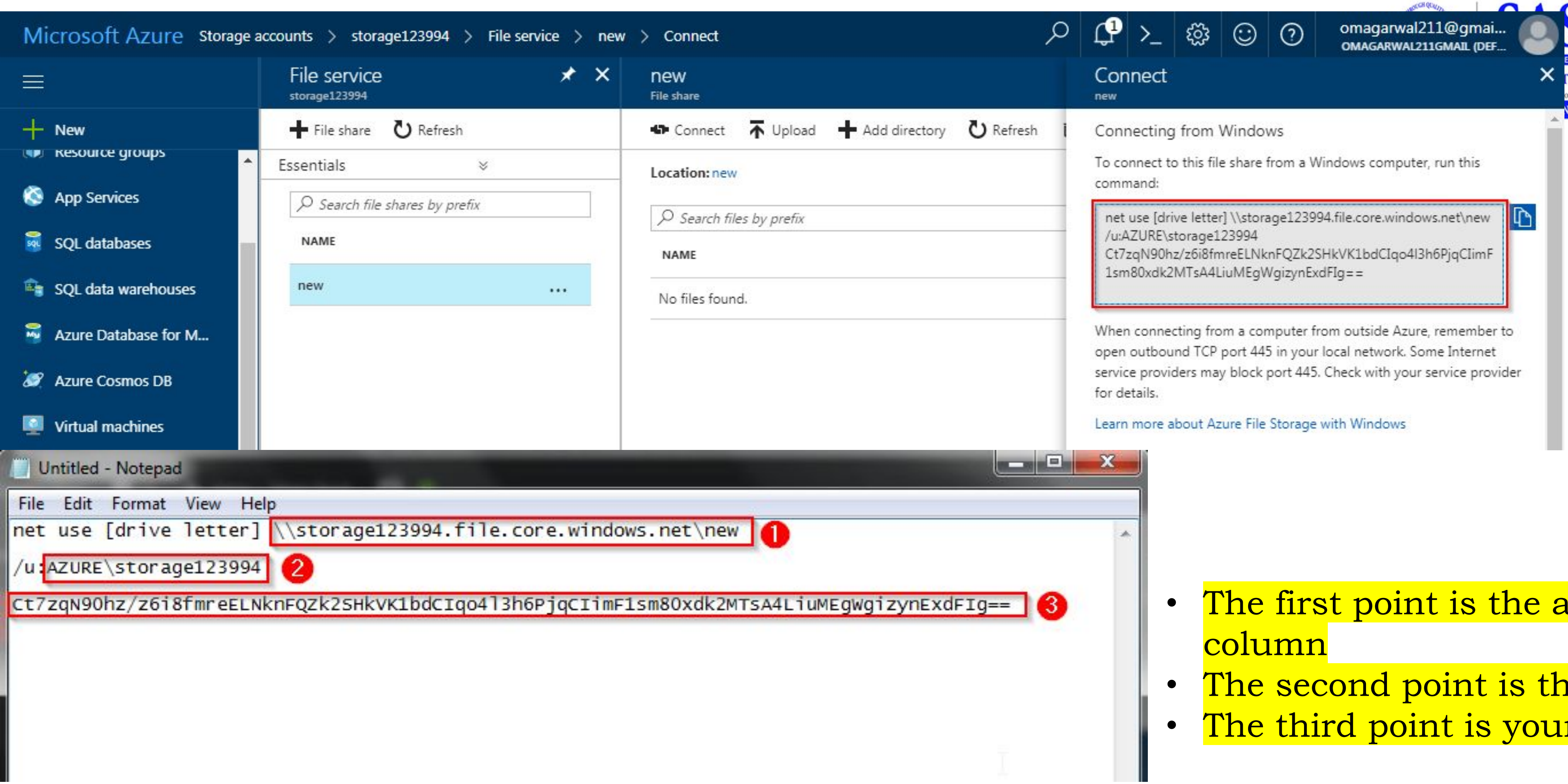


• Step 2: On the next page, enter the name of your file instance, and desired size of your instance. Finally, click on OK.



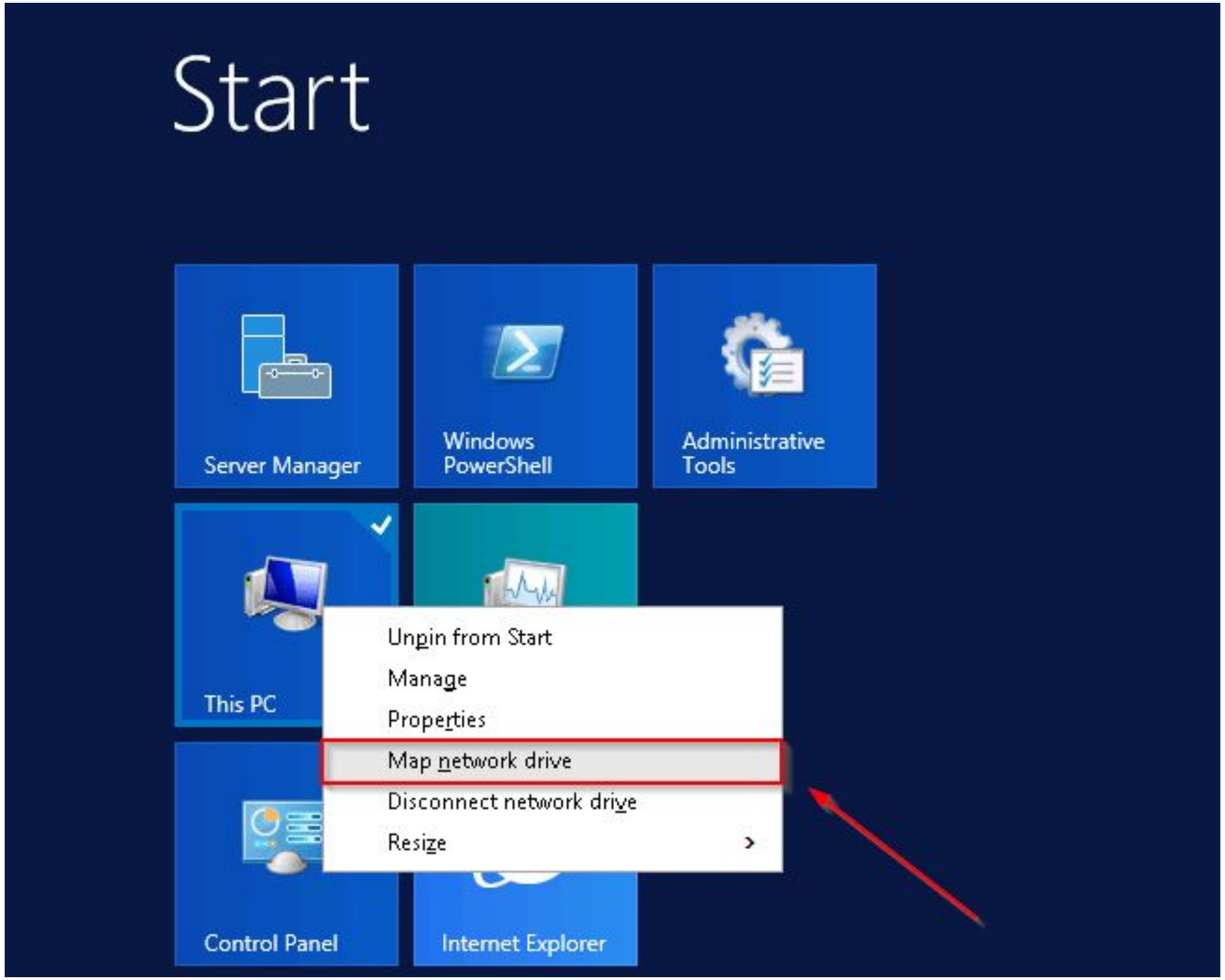
- Step 3: Select your file service, and then click on connect.



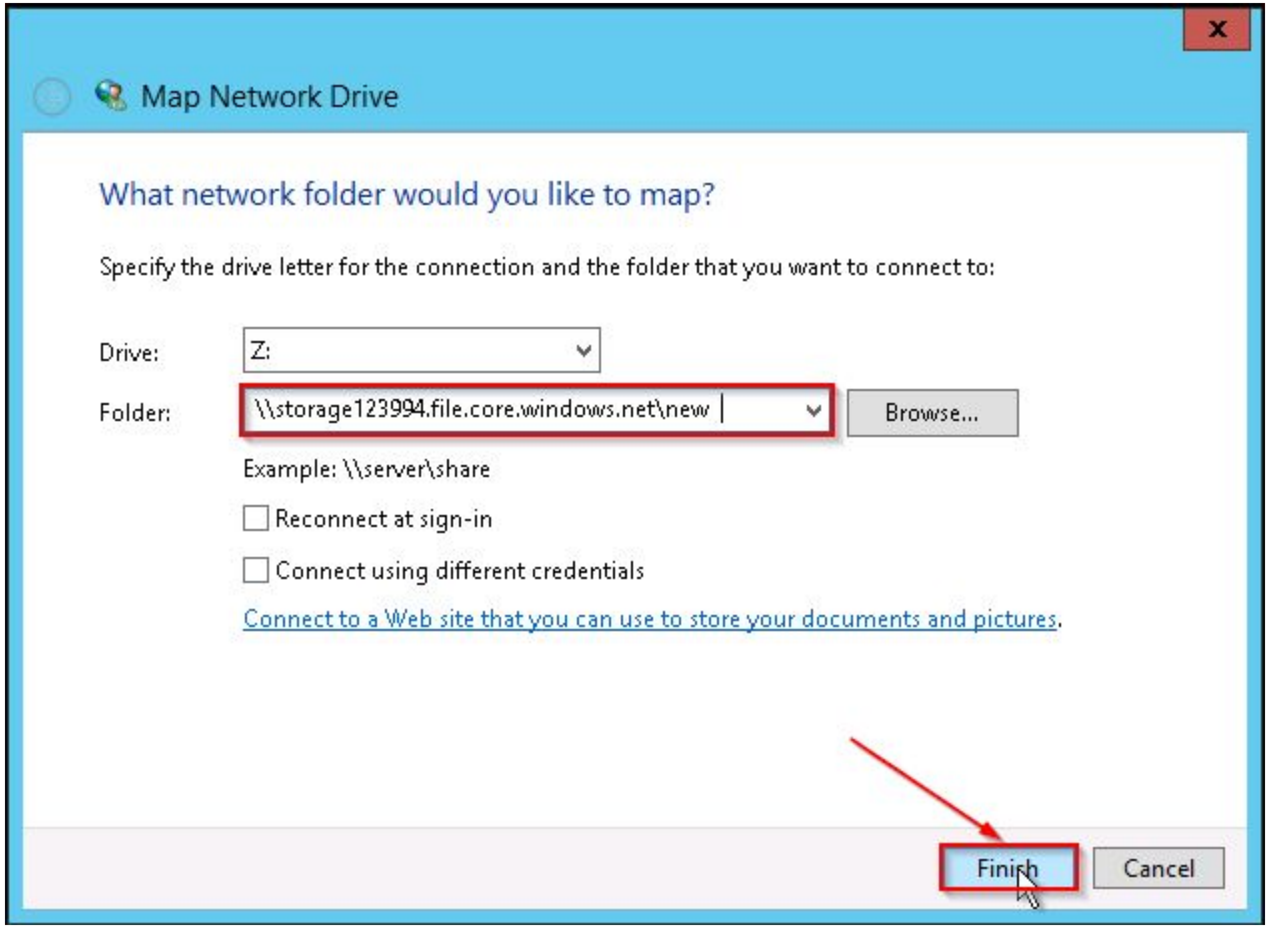


- The first point is the address column
- The second point is the user name
- The third point is your password

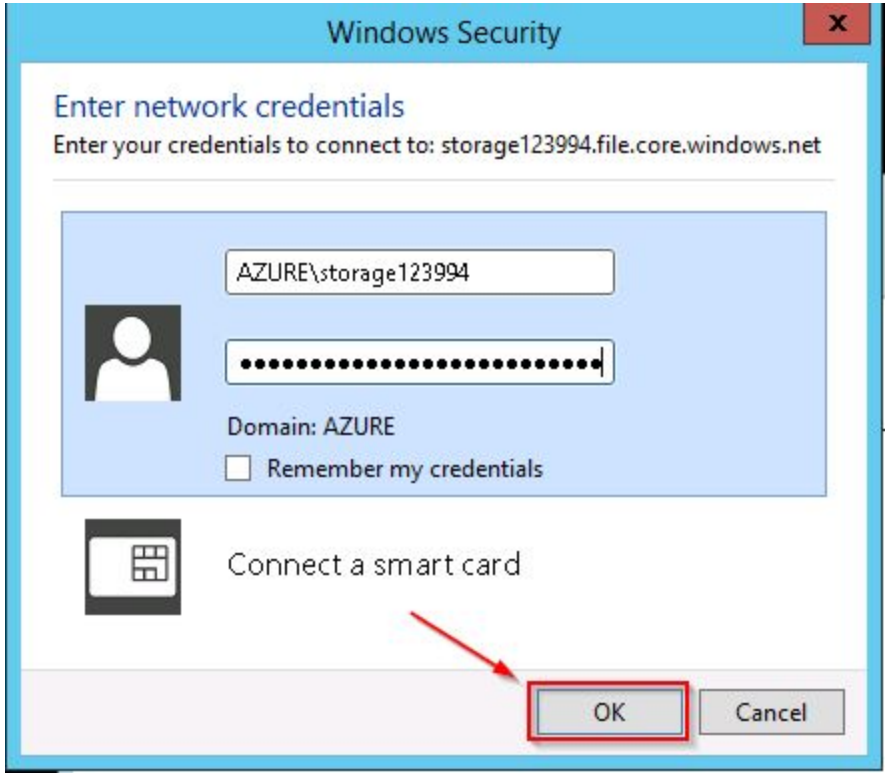
- Step 4: Right Click on your my computer icon, on your desktop and click on Map Network Drive.



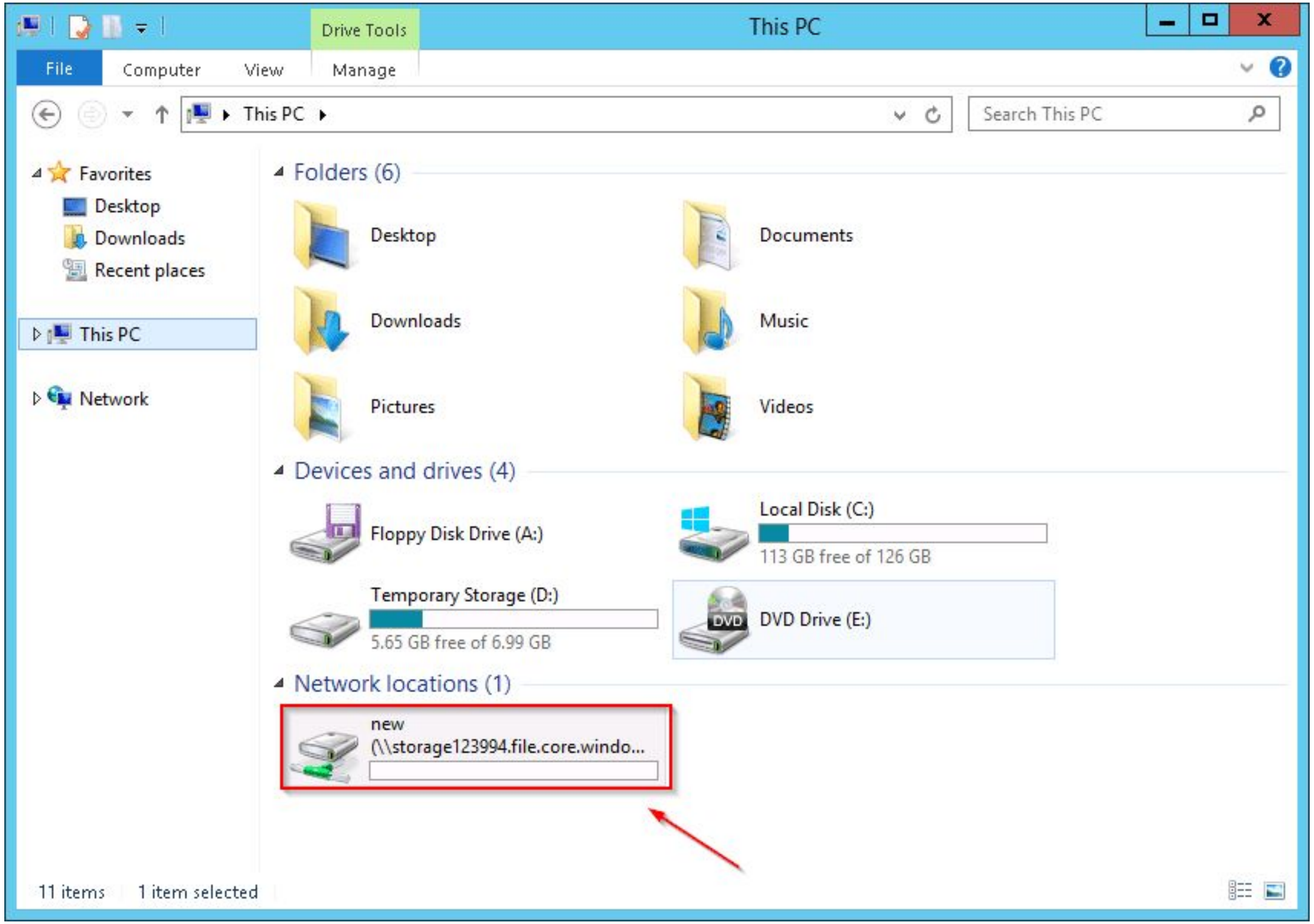
- Step 5: Enter the first point that you copied from your notepad in the folder text box and click on finish.



- Step 6: On the next step, enter the username and password from notepad, and finally click on OK.



- Step 7: Congratulations! Your azure storage drive is ready. You can now use it, like any other drive on your computer

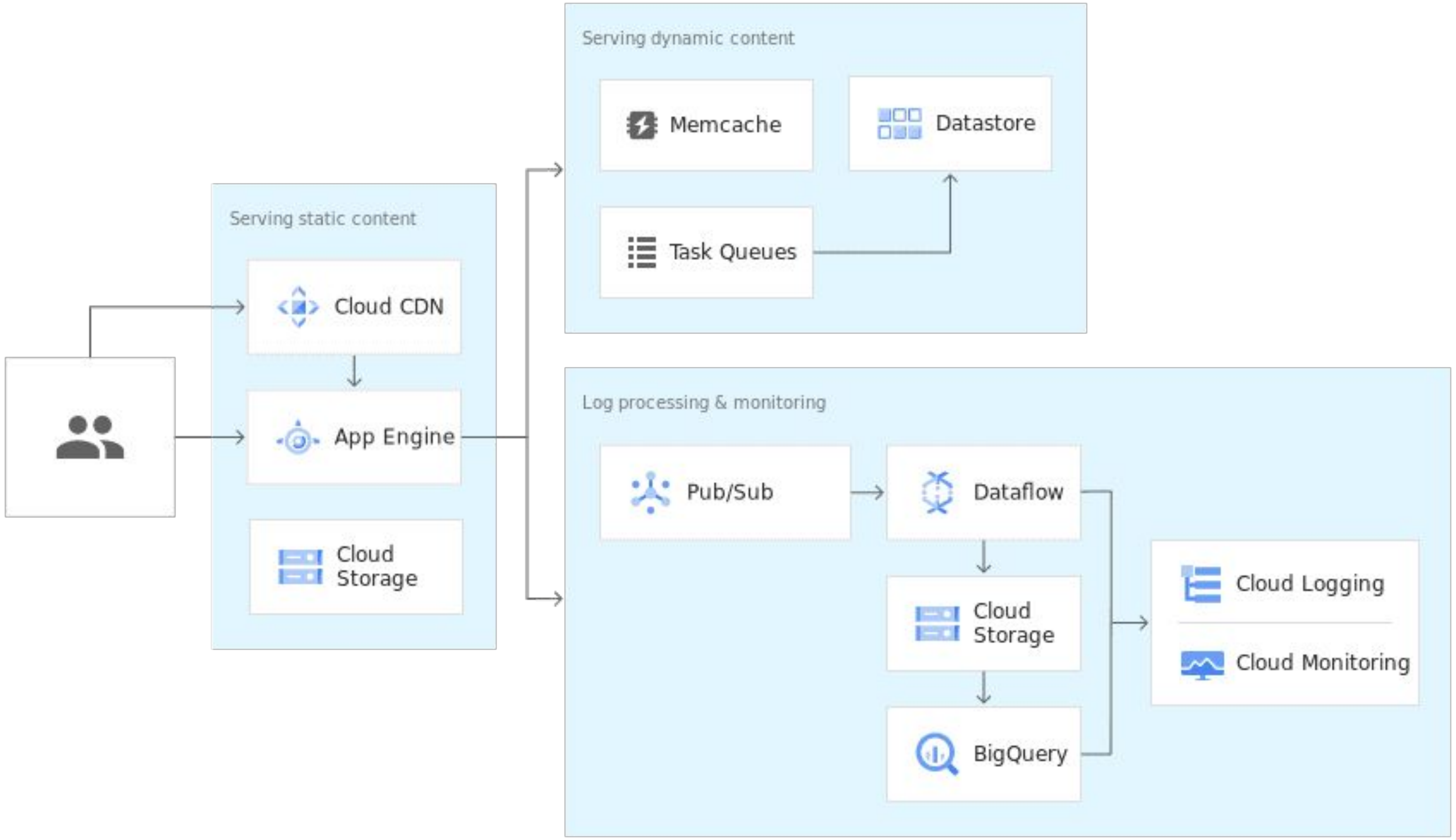


Using Google Cloud Storage Services

- Google AppEngine, Google Cloud, which includes their data storage services, their NoSQL services Cloud Datastore and Bigtable, and various computational services.
- Once you have an account, you can install the Google Cloud SDK which consists of the Google Cloud command-line tool and the gsutil package.
- These tools are available for Linux, Mac OS, and Windows

Google Cloud Platform (GCP) Services

- **Compute**
  - Google Compute Engine
  - Google App Engine
  - Google Kubernetes Engine
  - Google Cloud Container Registry
  - Cloud Functions
- **Networking**
  - Google Virtual Private Cloud (VPC)
  - Google Cloud Load Balancing
  - Content Delivery Network
  - Google Cloud Interconnect
  - Google Cloud DNS
- **Storage and Databases**
  - Google Cloud Storage
  - Cloud SQL
  - Cloud Bigtable
  - Google Cloud Datastore
  - Persistent Disk
- **Big Data**
  - Google BigQuery
  - Google Cloud Dataproc
  - Google Cloud Datalab
  - Google Cloud Pub/Sub
- **Cloud AI**
  - Cloud Machine Learning
  - Vision API
  - Speech API
  - Natural Language API
  - Translation API
  - Jobs API
- **Identity & Security**
  - Cloud Resource Manager
  - Cloud IAM
  - Cloud Security Scanner
  - Cloud Platform Security

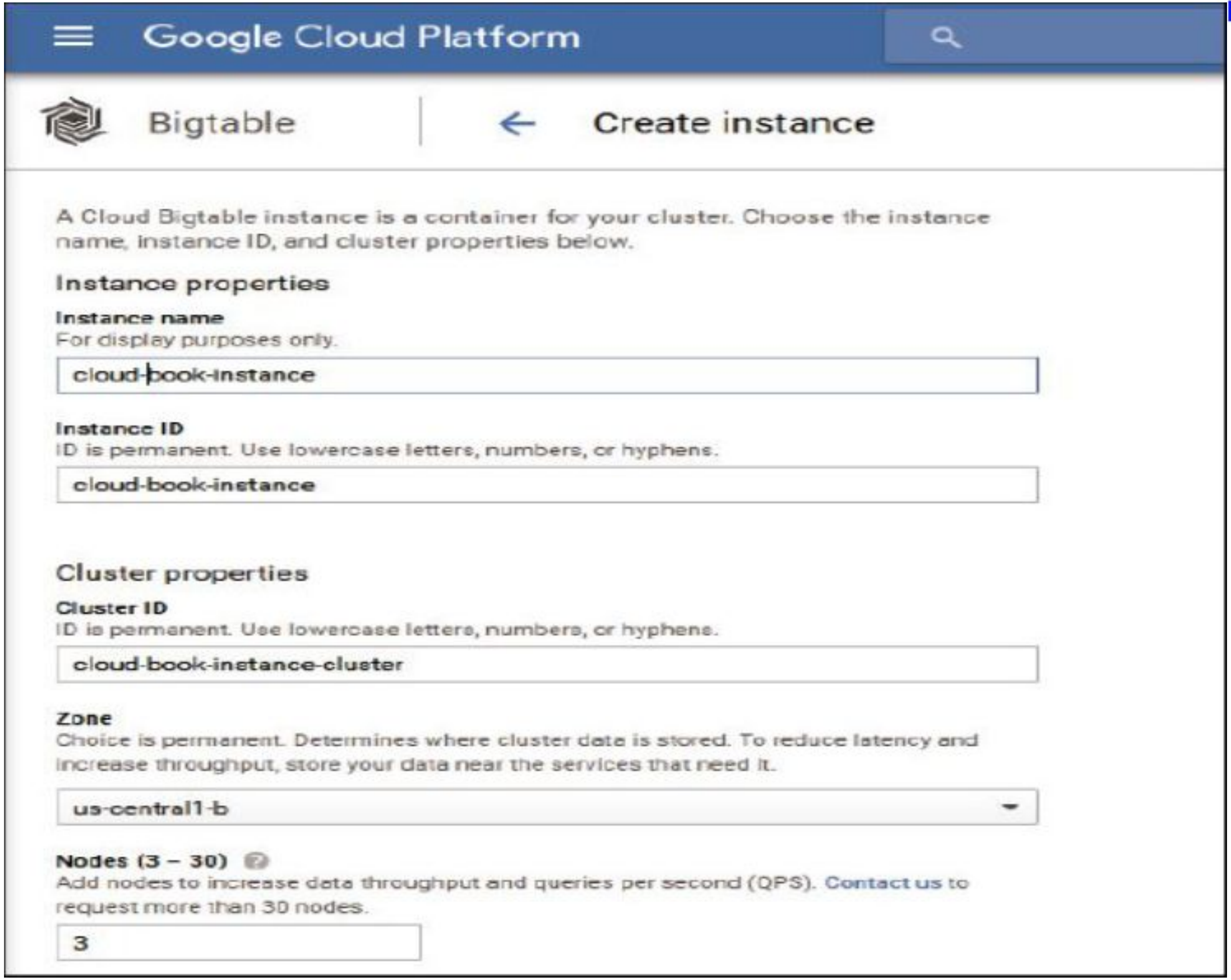


https://cloud.google.com/storage/docs/introduction

```
from gcloud import storage
client = storage.Client()
# Create a bucket with name 'afirstbucket'
bucket = client.create_bucket('afirstbucket')
# Create a blob with name 'my-test-file.txt' and
load some data
blob = bucket.blob('my-test-file.txt')
blob.upload_from_string('this is test content!')
blob.make_public()
```

Google Bigtable

- Bigtable is the predecessor of Apache HBase, the NoSQL store built on the **Hadoop Distributed File System (HDFS).**
- **Bigtable and HBase** are designed for large data collections that must be accessed quickly by major data analytics jobs.


Google Cloud Platform
Bigtable
Create instance
A Cloud Bigtable instance is a container for your cluster. Choose the instance name, instance ID, and cluster properties below.
Instance properties
Instance name
For display purposes only.
cloud-book-instance
Instance ID
ID is permanent. Use lowercase letters, numbers, or hyphens.
cloud-book-instance
Cluster properties
Cluster ID
ID is permanent. Use lowercase letters, numbers, or hyphens.
cloud-book-instance-cluster
Zone
Choice is permanent. Determines where cluster data is stored. To reduce latency and increase throughput, store your data near the services that need it.
us-central1-b
Nodes (3 – 30)
Add nodes to increase data throughput and queries per second (QPS). Contact us to request more than 30 nodes.
3

Bigtable | ← Create an instance

A Cloud Bigtable instance is a container for your clusters. Learn more

Instance name
For display purposes only

my-instance

Instance ID
ID is permanent

my-instance

Storage type
Choice is permanent. Applies to all clusters. Affects cost.

- (●) SSD
  Lower latency and more rows read per second. Typically used for real-time serving use cases, such as ad serving and mobile app recommendations.
- ( ) HDD
  Higher latency for random reads. Good performance on scans and typically used for batch analytics, such as machine learning or data mining.

Clusters

Cluster

Cluster ID
ID is permanent.

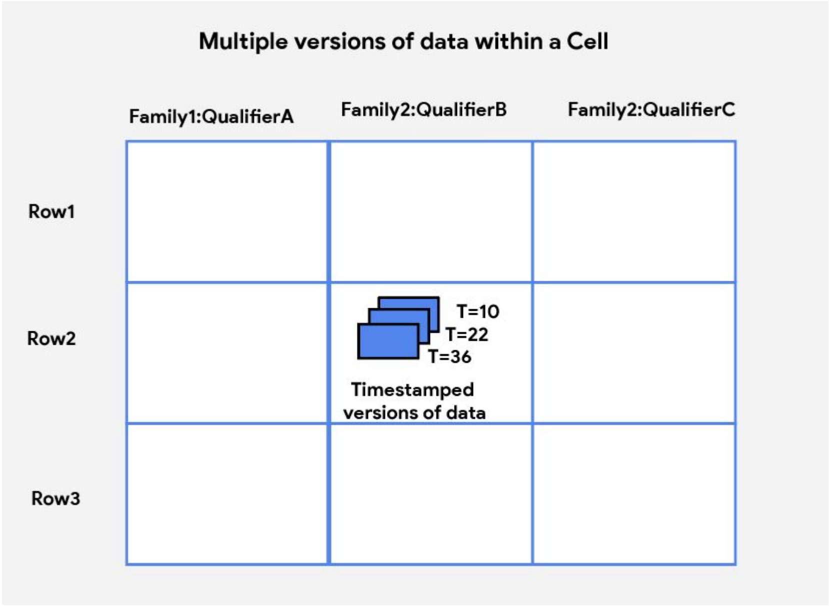
my-instance-c1

Location
Choice is permanent. Determines where cluster data is stored. To reduce latency and increase throughput, store your data near the services that need it.

Region	Zone
us-central1	us-central1-b

Nodes
Add nodes to increase your cluster's capacity for data throughput, storage, and rows read per second. Add enough to keep each cluster's CPU utilization under the recommended threshold for your current number of clusters and app profile routing

- Accessing your instance with CBT(The **CBT** tool is a **command-line interface for performing several different operations on Cloud Bigtable**)
    - *cbt listinstances*
    - *Missing –instance*
- Data structures & schema basics
    - *cbt createtable catalog*
- *Column Family*
- *Rows, Columns & Cells*
- *Cell Versions*



- Garbage Collection
- Querying and accessing Data
- Retrieve Single Entry
- Reading All Rows
- Start & End
- **Prefix**
- **Regex**
- **Count**
- **Schema Design**
- **Avoid Hot Spots**
- **Row Keys optimized for queries**
- **Cleanup**

```
from gcloud import bigtable
clientbt = bigtable.Client(admin=True)
clientbt.start()
instance = clientbt.instance('cloud-book-
instance')
table = instance.table('book-table')
table.create()
# Table has been created
column_family = table.column_family('cf')
column_family.create()

#now insert a row with key 'key1' and columns
'experiment', 'date',
#'link'
row = table.row('key1')
row.set_cell('cf', 'experiment', 'exp1')
row.set_cell('cf', 'date', '6/6/16')
row.set_cell('cf', 'link',
'http://some_location')
row.commit()
```

Google Cloud Datastore

```
from gcloud import datastore
clientds = datastore.Client()
key = clientds.key('blobtable')
```

To add an entity to the table, we write the following.

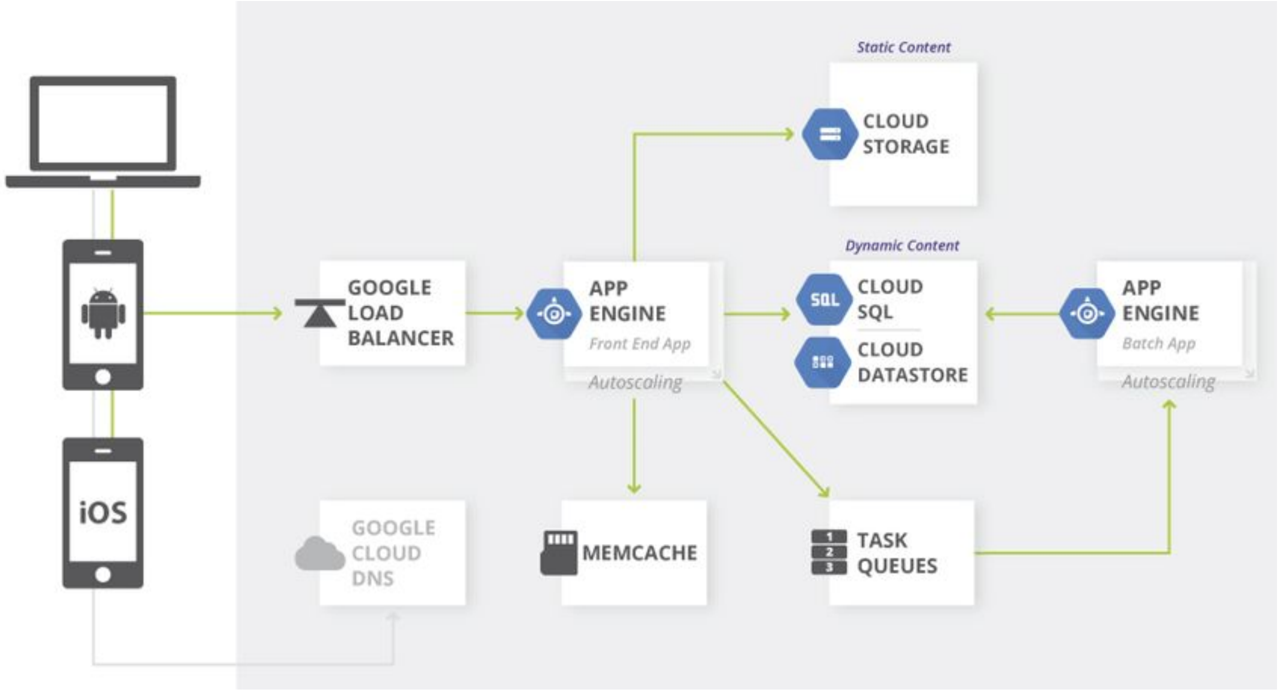
```
entity = datastore.Entity(key=key)
entity['experiment-name'] = 'experiment name'
entity['date'] = 'the date'
entity['description'] = 'the text describing the
experiment'
entity['url'] = 'the url'
clientds.put(entity)
```

```
from gcloud import storage
from gcloud import datastore
import csv

client = storage.Client()
clientds = datastore.Client()
bucket = client.bucket('book-datacont')
key = clientds.key('book-table')
```

```
with open('\path-to-your-data\experiments.csv',
'rb') as csvfile:
    csvf = csv.reader(csvfile, delimiter=',',
    quotechar='|')
    for item in csvf:
        print(item)
        blob = bucket.blob(item[3])
        data = open("\path-to-your-
        data\datafiles\\"+item[3], 'rb')
        blob.upload_from_file(data)
        blob.make_public()
        url = "https://storage.googleapis.com/book-
        datacont/"+item[3]
        entity = datastore.Entity(key=key)
        entity['experiment-name'] = item[0]
        entity['experiment-id'] = item[1]
        entity['date'] = item[2]
        entity['description'] = item[4]
        entity['url'] = url
        clientds.put(entity)
```

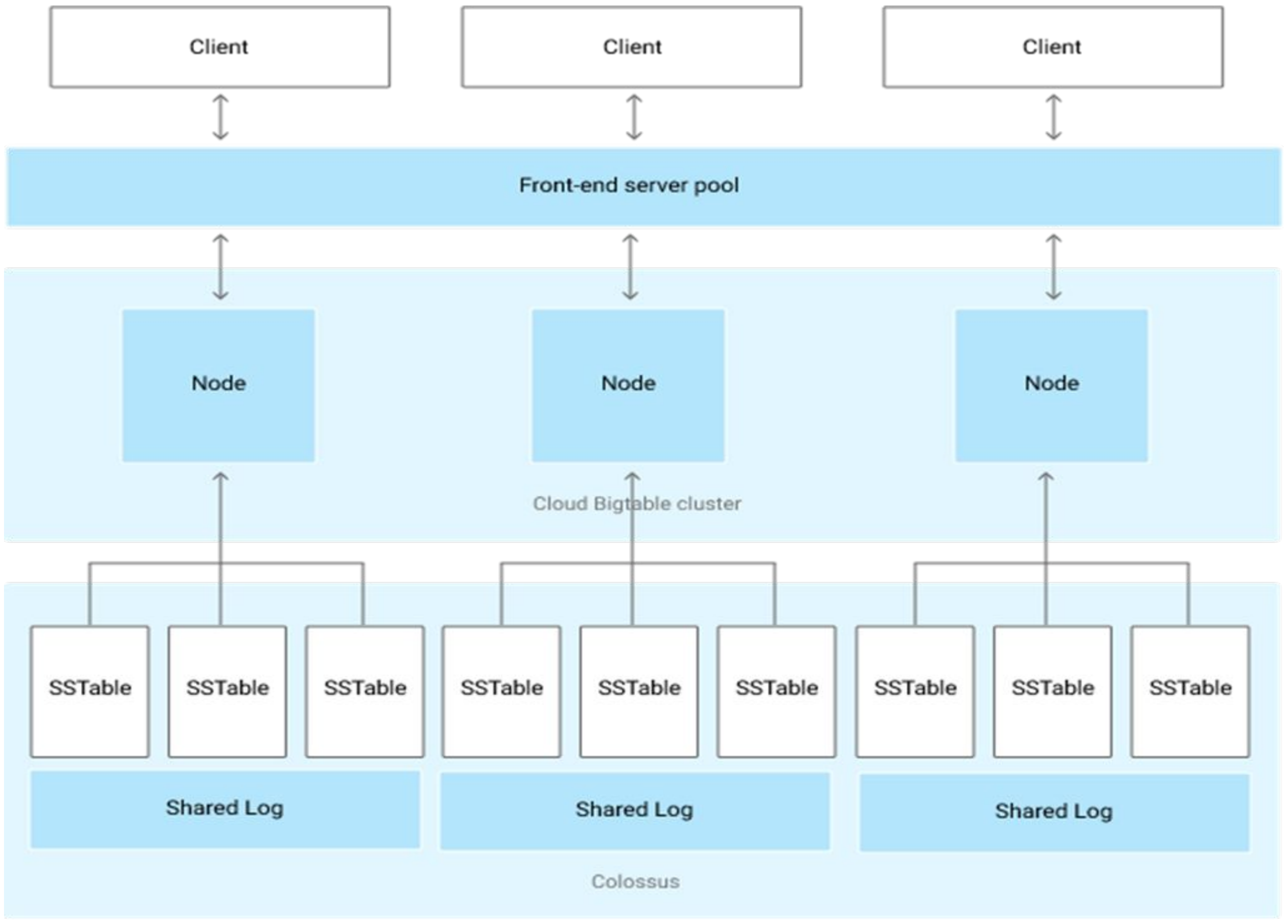
GOOGLE APP ENGINE

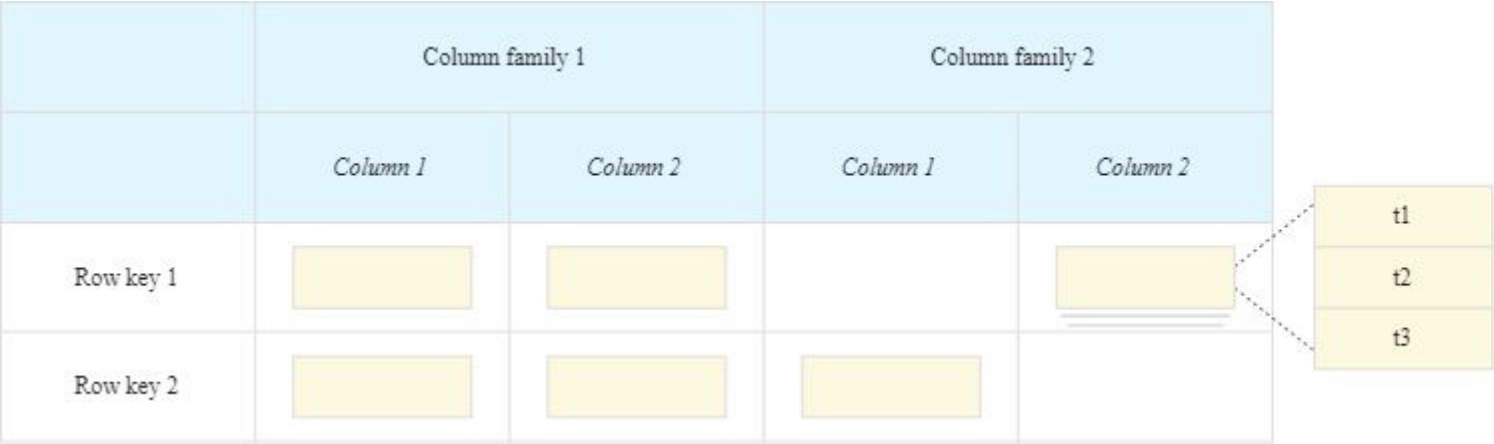
- Google App Engine is a PaaS solution that enables users to host their own applications on the same or similar infrastructure as **Google Docs, Google Maps and other popular Google services**



DEVELOPING AND DEPLOYING ON GOOGLE APP ENGINE

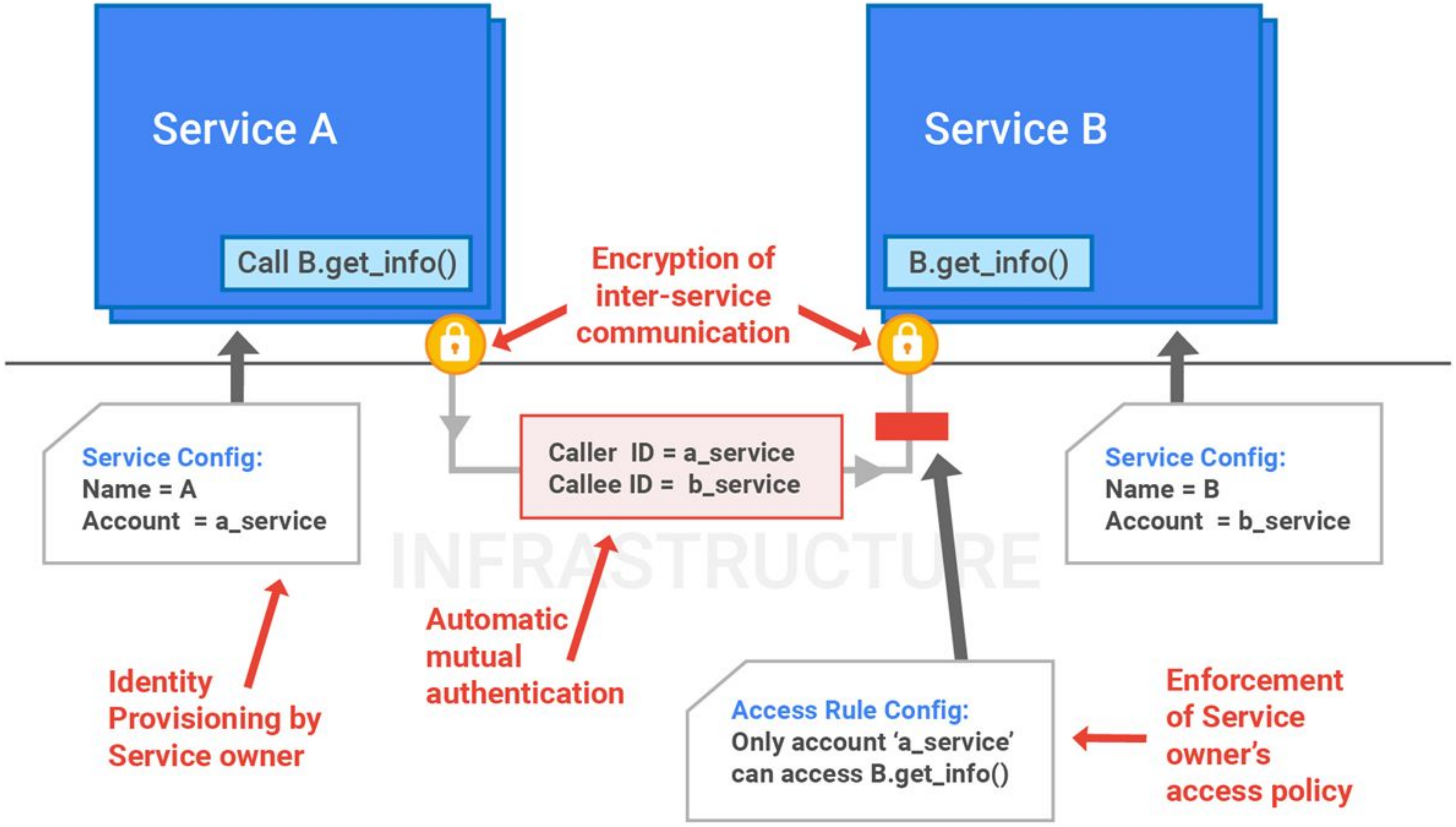
1. Download the SDK (Eclipse plug-in)
2. Create a new “Web Application Project”
3. Configure the application
4. Develop code
5. Test in simulated App Engine environment
6. Deploy to Google App Engine


Client
Client
Client
Front-end server pool
Node
Node
Node
Cloud Bigtable cluster
SSTable
SSTable
SSTable
SSTable
SSTable
SSTable
SSTable
SSTable
SSTable
Shared Log
Shared Log
Shared Log
Colossus

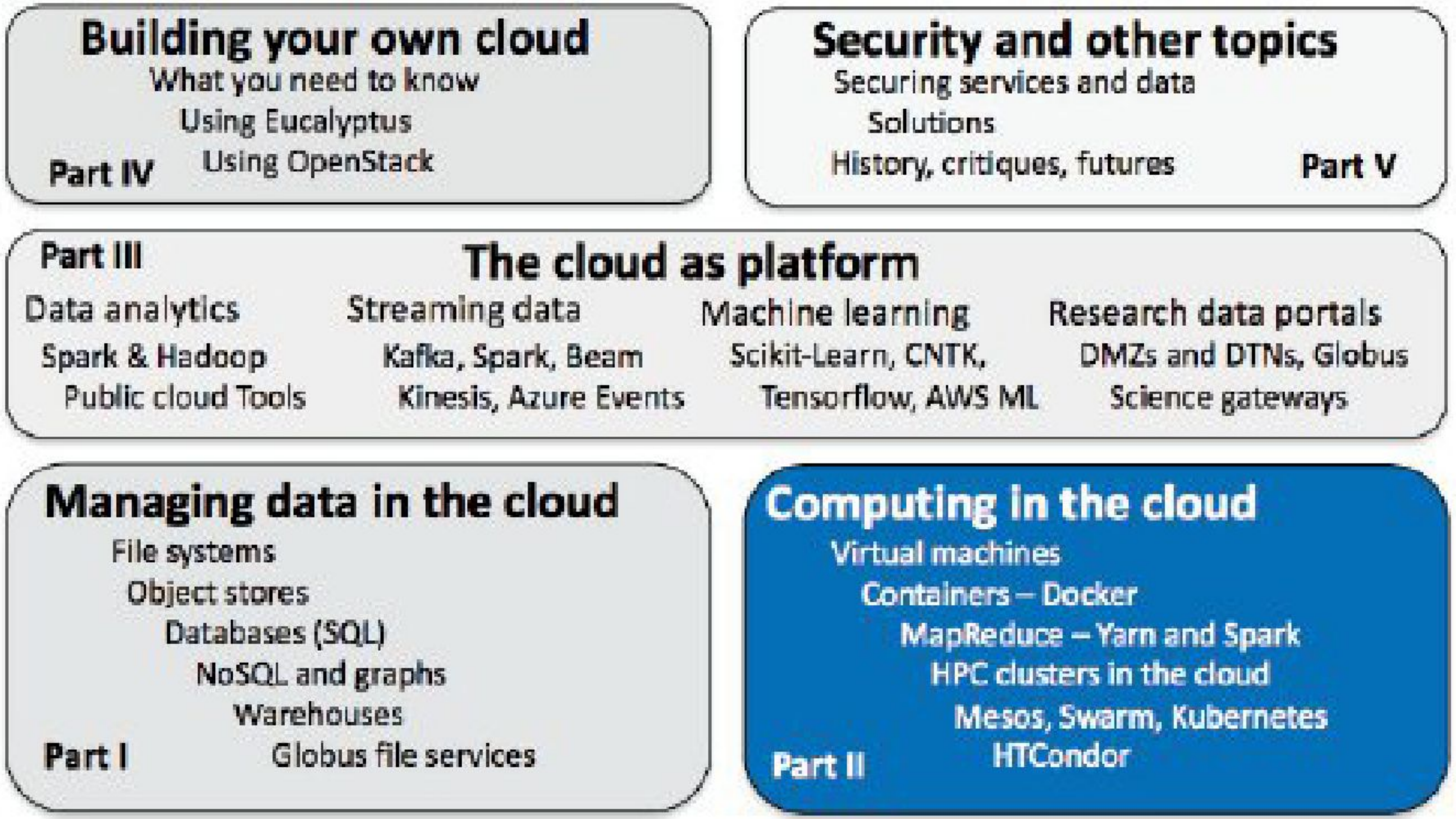


Column family 1
Column family 2
Column 1
Column 2
Column 1
Column 2
Row key 1
Row key 2
t1
t2
t3

Google Data Protocol (GDP)

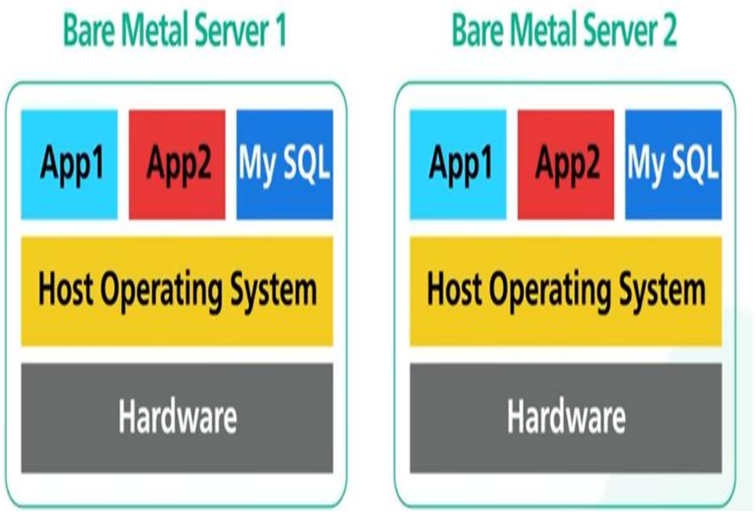


UNIT II

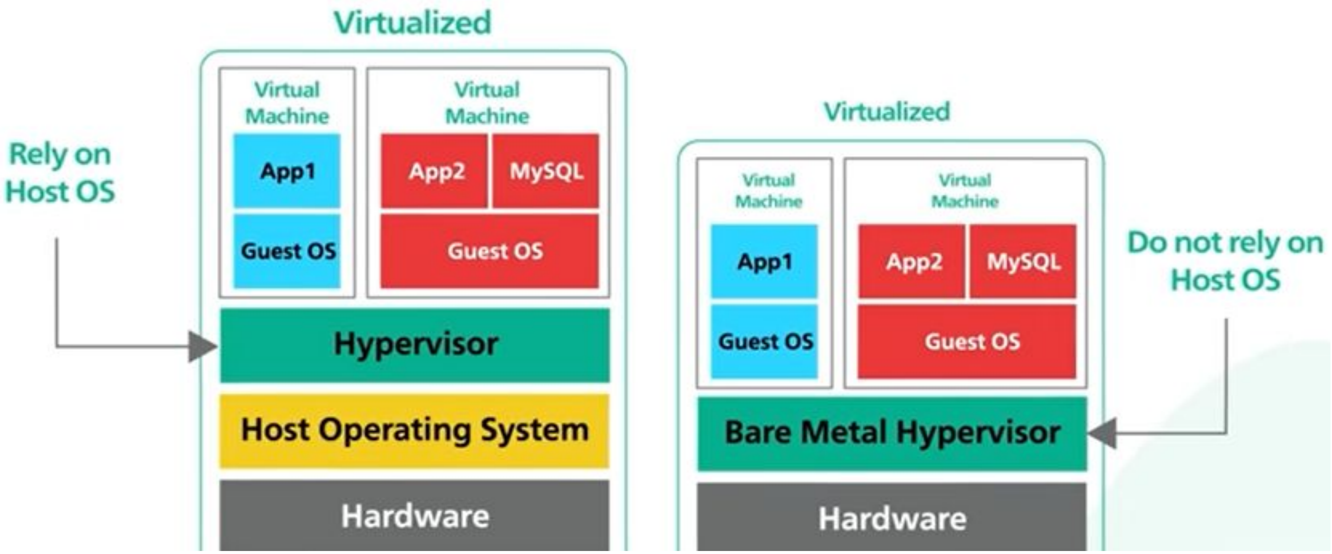

Building your own cloud
What you need to know
Using Eucalyptus
Using OpenStack
Part IV
Security and other topics
Securing services and data
Solutions
History, critiques, futures
Part V
Part III
The cloud as platform
Data analytics
Spark & Hadoop
Public cloud Tools
Streaming data
Kafka, Spark, Beam
Kinesis, Azure Events
Machine learning
Scikit-Learn, CNTK,
Tensorflow, AWS ML
Research data portals
DMZs and DTNs, Globus
Science gateways
Managing data in the cloud
File systems
Object stores
Databases (SQL)
NoSQL and graphs
Warehouses
Globus file services
Part I
Computing in the cloud
Virtual machines
Containers – Docker
MapReduce – Yarn and Spark
HPC clusters in the cloud
Mesos, Swarm, Kubernetes
HTCondor
Part II

Computing in the Cloud

The most basic is known in the cloud industry as **Infrastructure As A Service (Iaas)** because it provides virtualized infrastructure to its users.



Bare Metal H/W

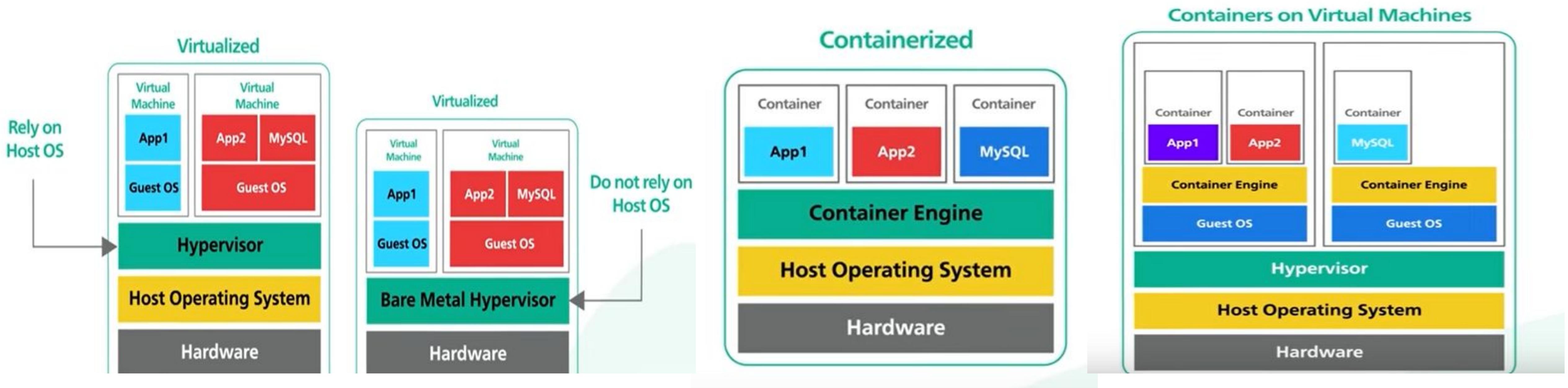


Hosted or Type 2 Hypervisor

Bare Metal or Native Type 1 Hypervisor

Virtual Machines and Containers

- A virtual machine is just the software image of a complete machine that can be loaded onto the server and run like any other program.
- The server in the data center runs a piece of software called a **HYPERVISOR that allocates and manages the server's resources** that are granted to its "guest" virtual machines.

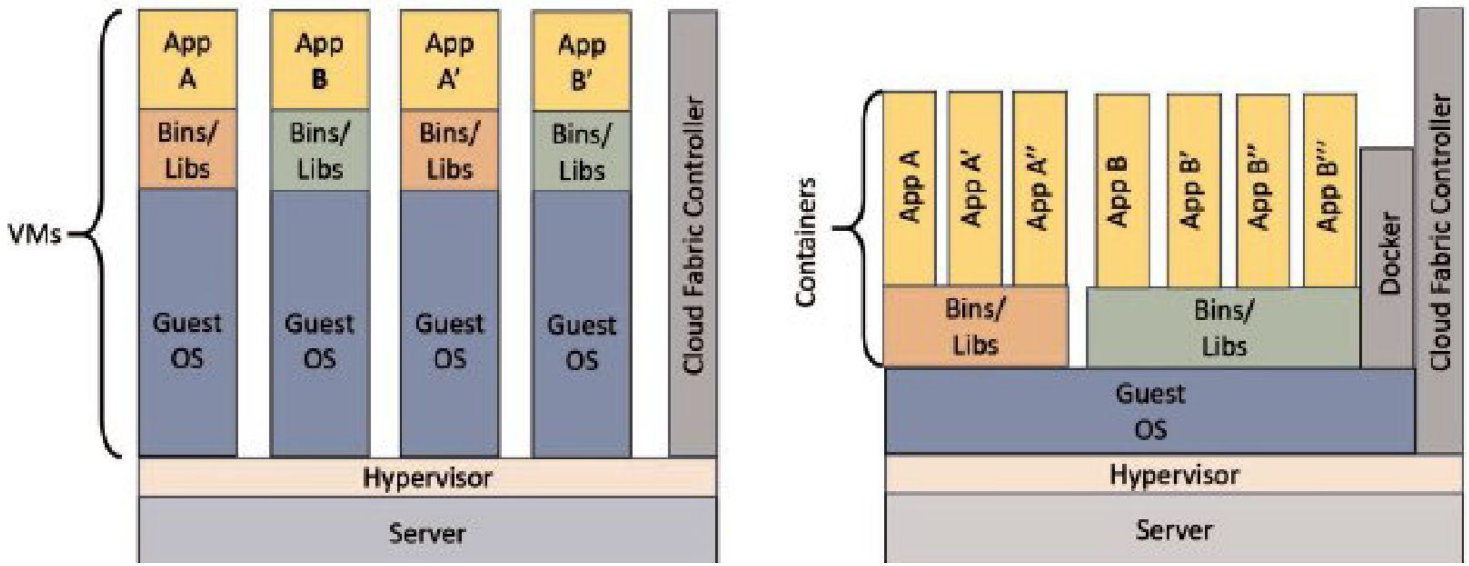


- To the hypervisor, all VMs look the same and can be managed in a uniform way.
- The cloud management system (sometimes called the fabric controller) can select which server to use to run the requested VM instances, and it can monitor the health of each VM
- **NATIVE HYPERVISORS /TYPE 1/BARE METAL (VMWARE.ESXI from vmware, Hyper V from Microsoft, Oracle VM from oracle)**
- **HOSTED HYPERVISORS /TYPE 2/ (VMWARE.GSX from vmware,virtual box from oracle, virtual PC from Microsoft)**
- **HYBRID HYPERVISORS**
- **BARE METAL OR NATIVE HYPERVISORS** run directly on the hardware, providing all the features (e.g., I/O) needed by the guests.
- **HOSTED HYPERVISORS** run on top of an **existing OS** and leverage the features of the underlying OS.
- Virtual machines run on top of the hosted hypervisor, which runs on top of an existing OS.

Containers

- Containers are similar to VMs but are based on a different technology and serve a slightly different purpose.
- Rather than run a full OS, a container is layered on top of the host OS and uses that OS's resources in a clever way.
- Containers allow you to **PACKAGE UP AN APPLICATION** and all of its library dependencies and data into a **single, easy-to-manage unit**.
- When you launch the container, the **application can be configured to start up, go through its initialization, and be running in seconds**

Virtual machines vs. containers on a typical cloud server



Virtual machines Vs Containers

Virtual machines	Containers
VM is piece of software that allows you to install other **software inside of it so you basically control it virtually** as opposed to installing the software directly on the computer.	While a container is a software that allows different **functionalities of an application independently**.
Applications running on VM system can run different OS.	While applications running in a container environment **share a single OS**.
VM virtualizes the computer system	While containers virtualize the **operating system only.**
VM size is very large	While the size of container is very light; i.e. a few megabytes.
VM's are useful when we require all of OS resources to run various application	While containers are useful when we are required to maximize the running applications using minimal servers.
Examples of VM are: KVM, Xen, VMware	While examples of containers are:RancherOS, PhotonOS, **Containers by Docker**

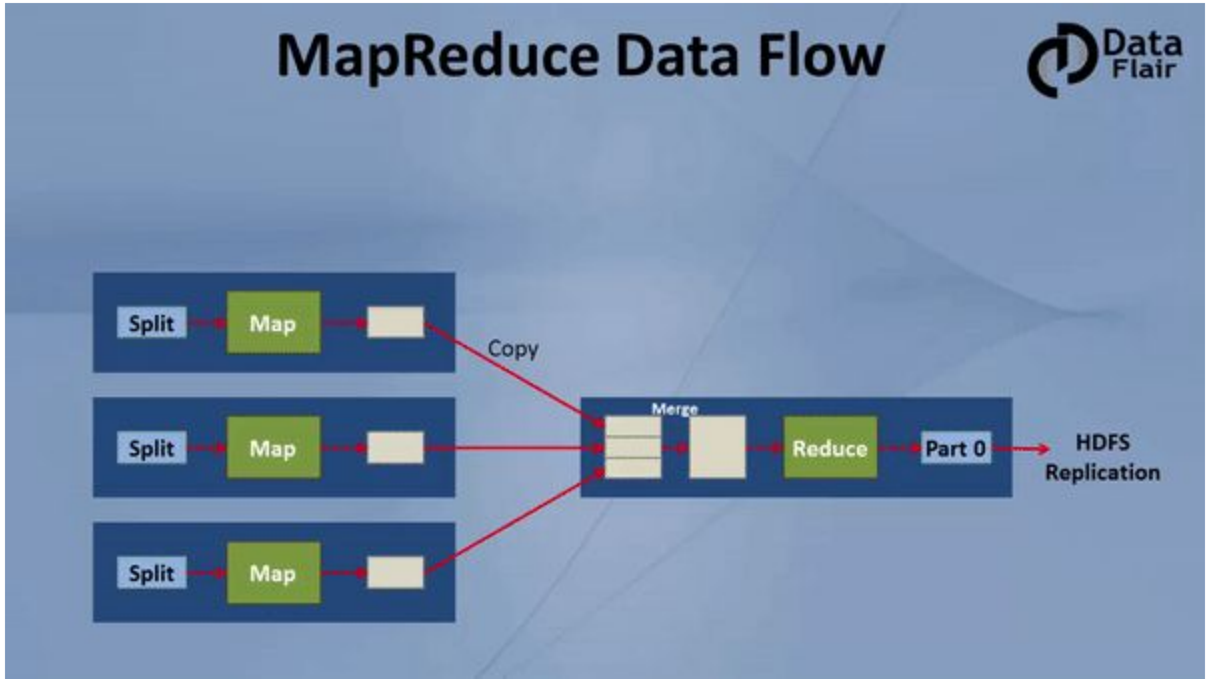
Virtual machines Vs Containers

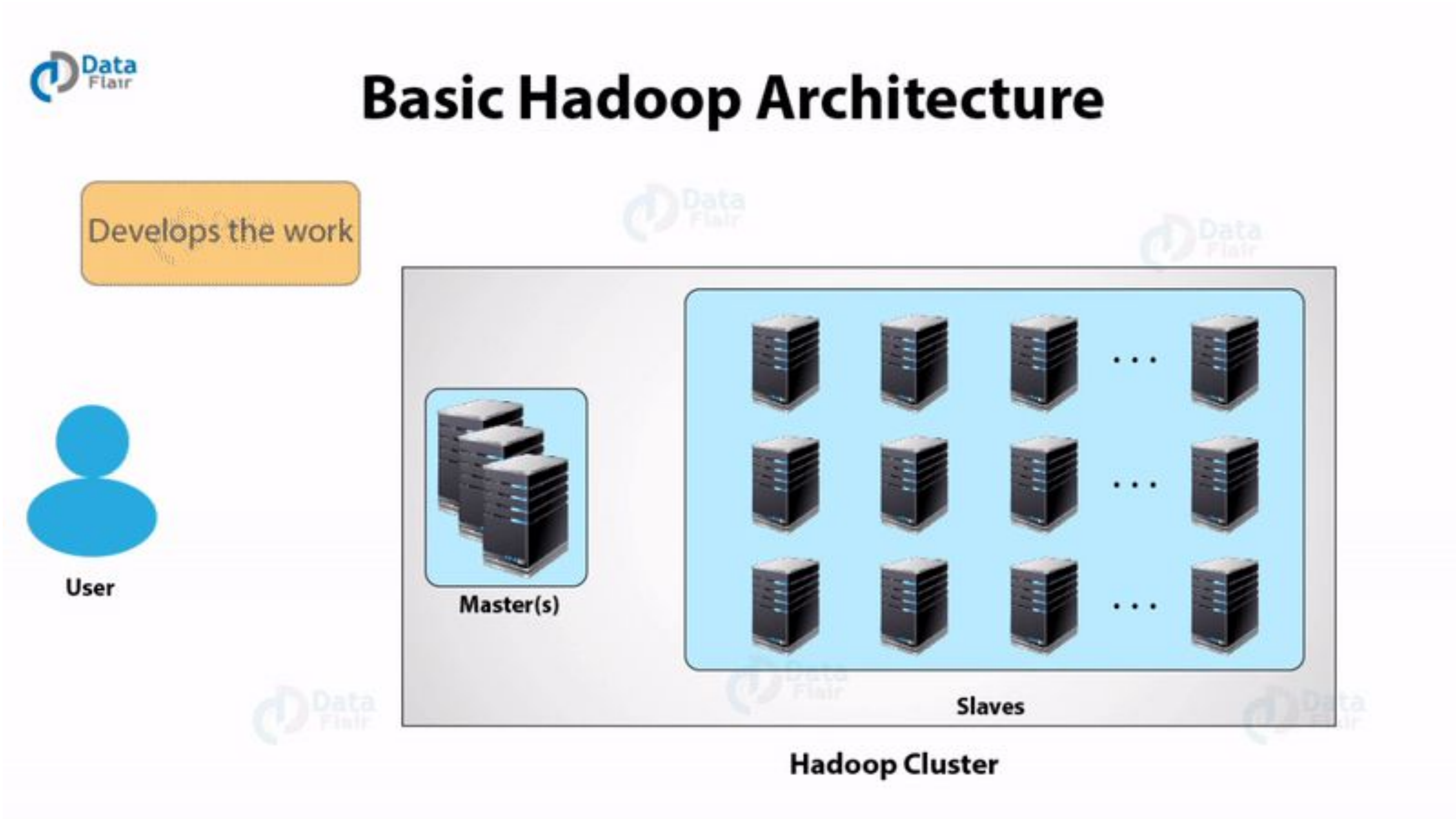
Virtual machines	Containers
Heavyweight	Lightweight
Fully isolated; hence more secure	Process-level isolation; hence less secure
No automation for configuration	Script-driven configuration
Slow deployment	Rapid deployment
Easy port and IP address Mapping’s	More abstract port and IP Mappings
Custom images not portable across clouds	Completely portable

Advanced Computing Services

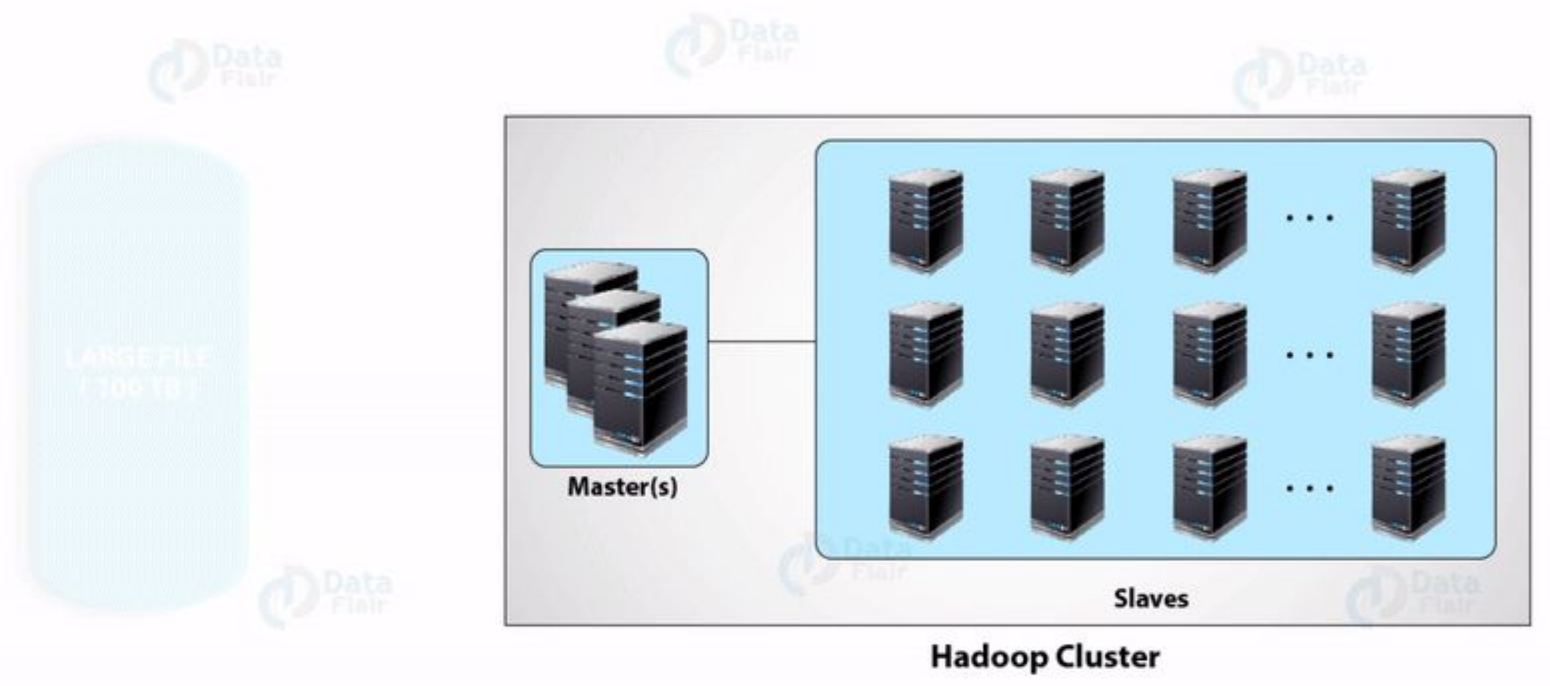
- A common issue of concern to scientists and engineers is scale.
- VMs and containers are a great way to virtualize a single machine image.
- Most high-performance parallel applications are based on the **Message Passing Interface (MPI)** standard.
- For example, **Many Task (MT) parallelism** is used to tackle problems in which you have hundreds of **similar tasks to run**, each(largely) independent of the others.
- Another method is called **MapReduce** made popular by the Hadoop computational framework
- MapReduce is related to a style of parallel computing known as **Bulk Synchronous Parallelism (BSP)**

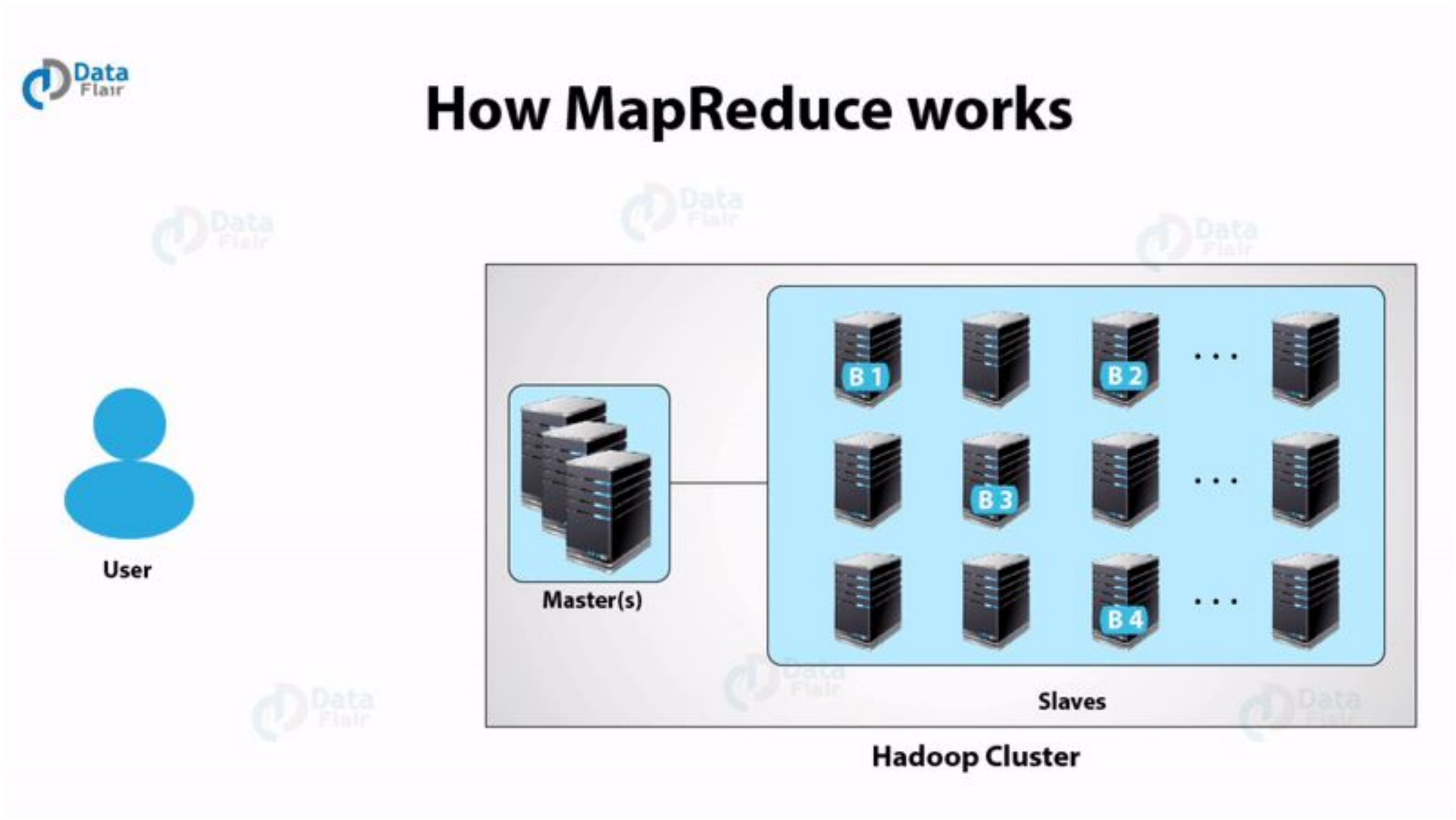
- Google has released a service called **Cloud Datalab**, based on Jupyter, for interactive control of its data analytics cloud.
- The Microsoft **Cloud Business Intelligence (Cloud BI)** tool supports interactive access to data queries and visualization



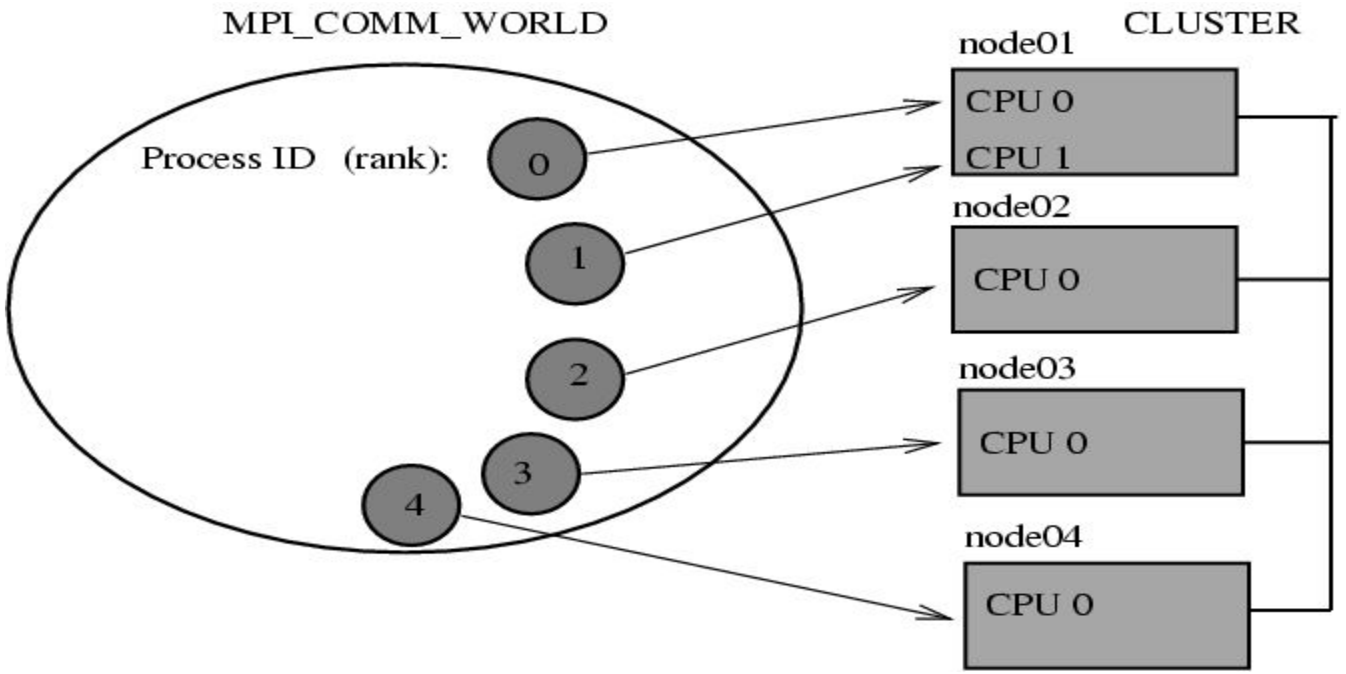

Data Flair
Basic Hadoop Architecture
Develops the work
User
Master(s)
Slaves
Hadoop Cluster

Data Storage in HDFS

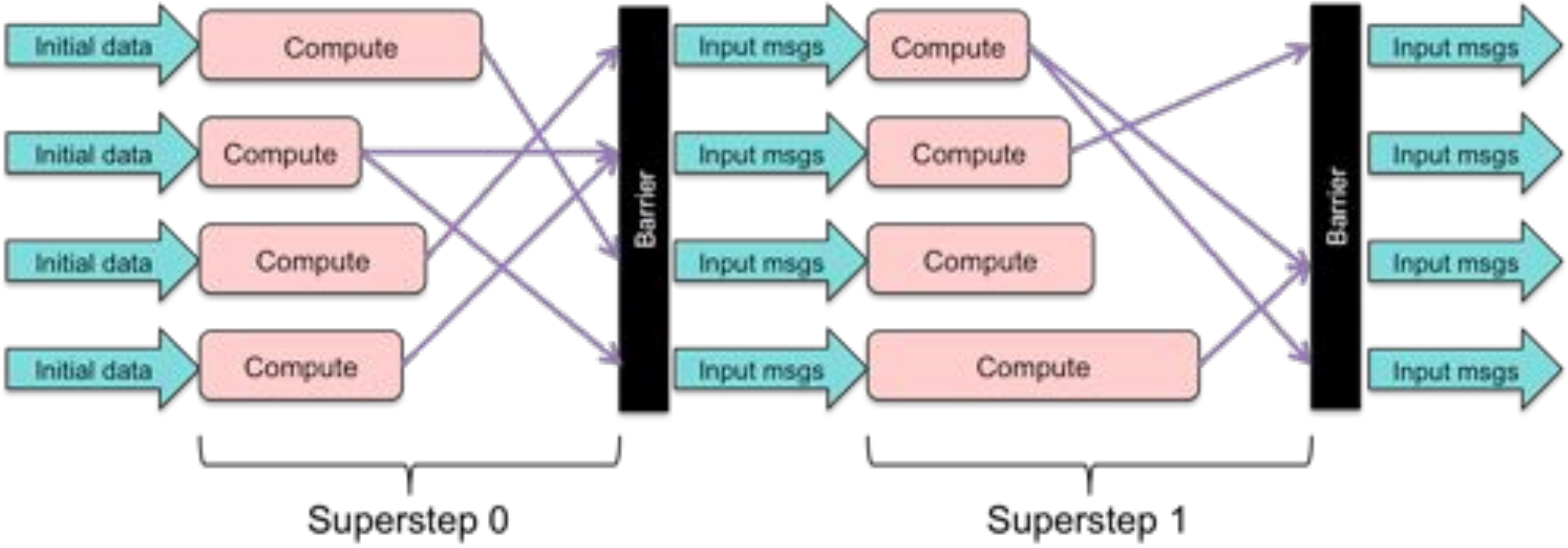


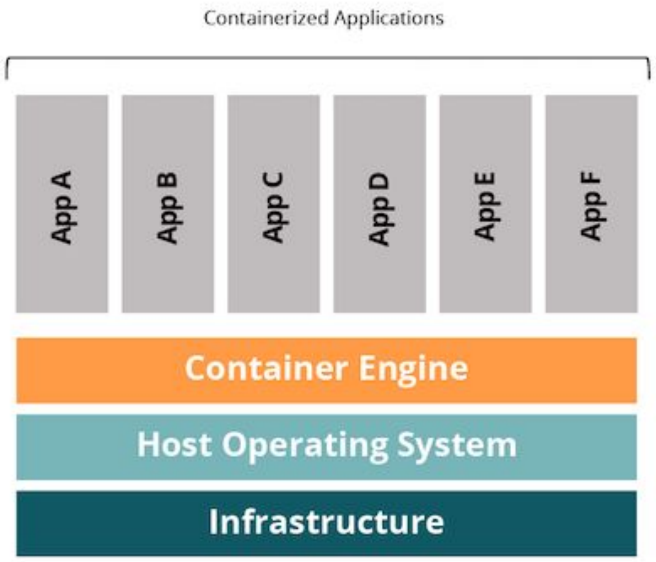

Data Flair
How MapReduce works
User
Master(s)
B 1
B 2
B 3
B 4
Slaves
Hadoop Cluster

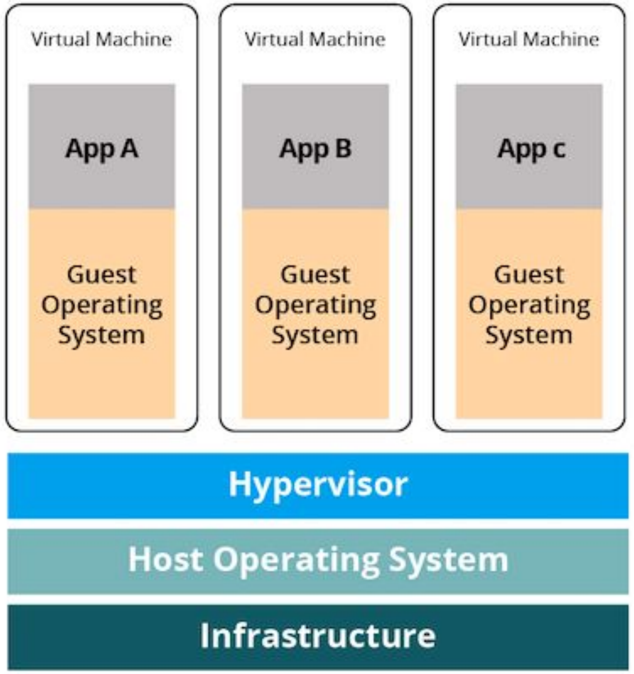
- MPI stands for **Message Passing Interface**.
- MPI is used to send messages from **one process (computer, workstation etc.) to another**.
- These messages can contain data ranging from **primitive types (integers, strings and so forth) to actual objects**

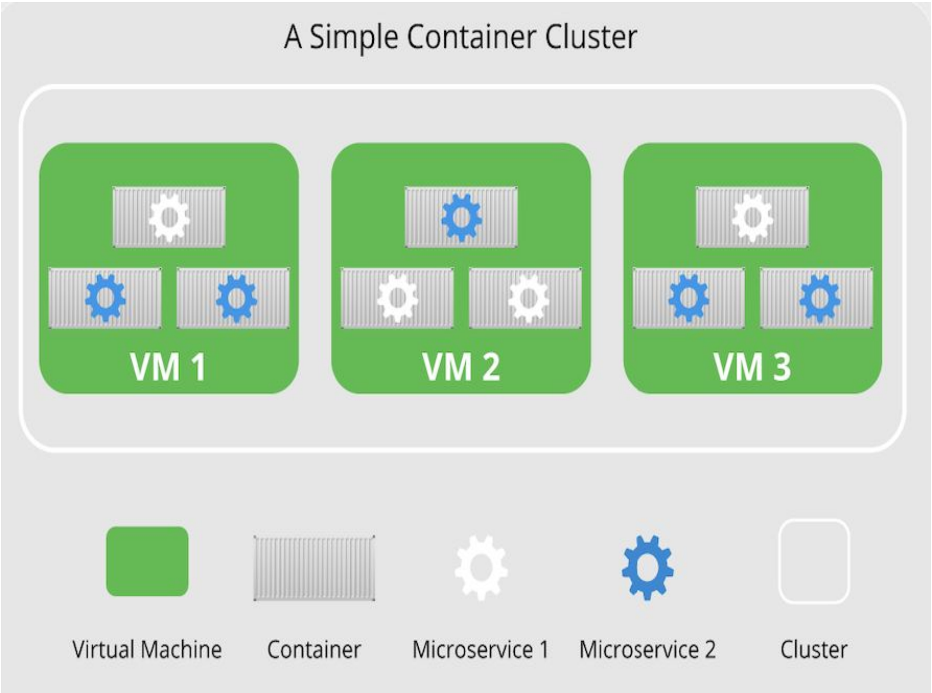


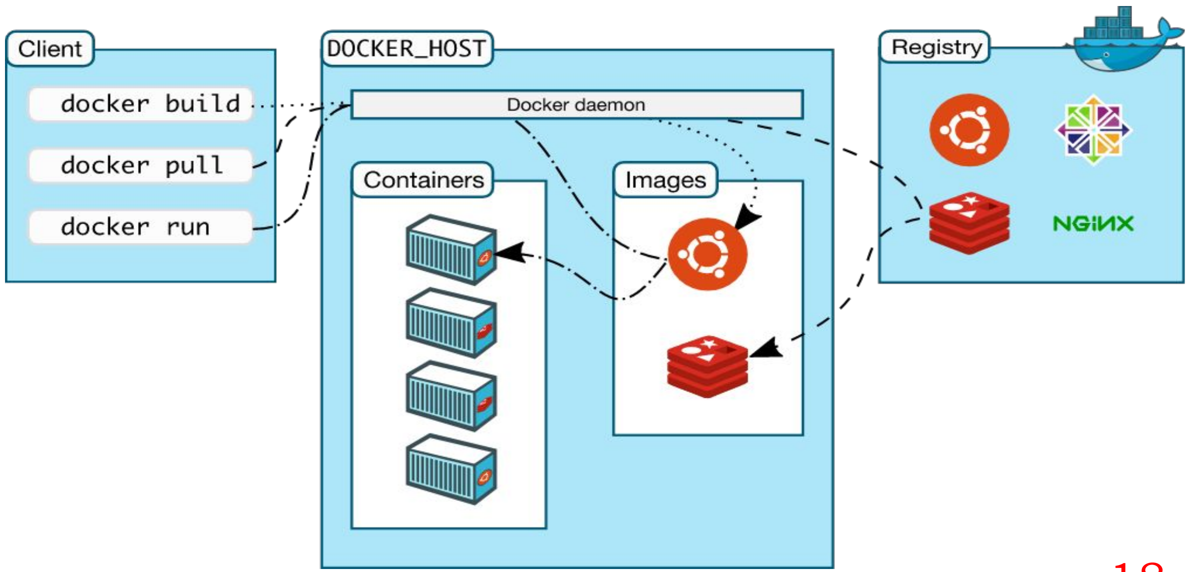
- **Bulk Synchronous Parallel (BSP**) is a programming model and computation framework for parallel computing.
- Computation is divided into a sequence of **SUPERSTEPS**.
- In each superstep, a **set of processes, running the same code**, executes **concurrently and creates messages** that are sent to other processes.
- The superstep ends when all the computation in the **superstep** is complete and **all messages have been** sent.
- A **BARRIER SYNCHRONIZATION** at the end of the superstep ensures that all messages have been transmitted (but not yet delivered to the processes).


Initial data
Initial data
Initial data
Initial data
Compute
Compute
Compute
Compute
Barrier
Input msgs
Input msgs
Input msgs
Input msgs
Compute
Compute
Compute
Compute
Barrier
Input msgs
Input msgs
Input msgs
Input msgs
Superstep 0
Superstep 1


Containerized Applications
App A
App B
App C
App D
App E
App F
Container Engine
Host Operating System
Infrastructure


Virtual Machine
App A
Guest Operating System
Virtual Machine
App B
Guest Operating System
Virtual Machine
App c
Guest Operating System
Hypervisor
Host Operating System
Infrastructure


A Simple Container Cluster
VM 1
VM 2
VM 3
Virtual Machine
Container
Microservice 1
Microservice 2
Cluster

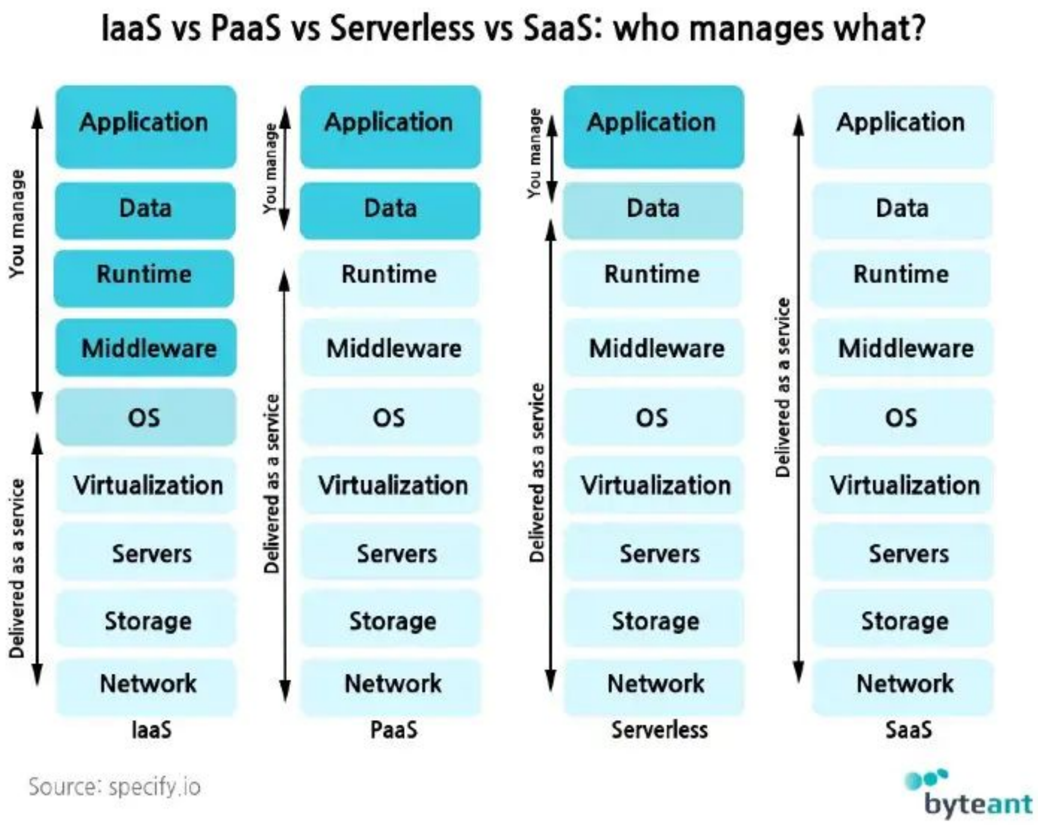

Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX

Serverless Computing

- Serverless is a cloud application development and execution model that lets developers build and run code **without managing servers, and without paying for idle cloud infrastructure**.
- Serverless does not mean **'NO SERVERS'**
- The term 'serverless' is somewhat misleading, as there are **still servers providing these backend services**, but all of the **server space and infrastructure concerns are handled by the vendor**.
- Serverless means that the developers can do their work **without having to worry about servers at all.**

https://www.ibm.com/cloud/learn/serverless#:~:text=Serverless%20is%20a%20cloud%20computing,managing%20servers%20or%20backend%20infrastructure.

- Serverless architecture is largely based on a **Functions as a Service (FaaS) model** that allows cloud platforms to execute code without the need for fully provisioned infrastructure instances.
- **FaaS, also known as Compute as a Service (CaaS),** are stateless, server-side functions that are event-driven, scalable, and fully managed by cloud providers.
- Serverless computing is the **abstraction of servers, infrastructure, and operating systems.**
- When you build serverless apps you don't need to provision and manage any servers, so you can take your **mind off infrastructure concerns.**


IaaS vs PaaS vs Serverless vs SaaS: who manages what?
You manage
Delivered as a service
Application
Data
Runtime
Middleware
OS
Virtualization
Servers
Storage
Network
IaaS
PaaS
Serverless
SaaS
Source: specify.io
byteant

Pros and Cons

- Pros and Cons
  - PROS
    - Cost
    - Scalability
    - Access
    - Configurability
    - Variety
    - Upgradeability
    - Security
    - Simplicity
  - CONS
    - Cost
    - Variety:
    - Security
    - Dependence

PROS	Description
Cost	If you need a resource for only a few hours or days, the cloud is much **cheaper than buying a new machine**
Scalability	You are starting a new lab and want to start with a small number of servers, but as your research **group grows**, you want to be able to expand easily without the hassle of managing your own racks of servers.
Access	A researcher in a small university or an engineer in a small company may not even have a computer room or a lab with room for racks of machines. The cloud may be the only choice.
Configurability	For many scientific disciplines, you can get complete virtual machines or containers with all the standard software you need pre-installed

PROS	Description
Variety	Public cloud systems provide access to a growing diversity of computer systems. Amazon and Azure each provide dozens of machine configurations, ranging from a single core with a gigabyte of memory to multicore systems with **GPU accelerators and massive amounts of memory**.
Security	• Commercial cloud providers have excellent security. • They also make it easy to create a virtual network that **integrates cloud resources into your network**
Upgradeability	• Cloud hardware is constantly **upgraded**. • Hardware that you buy is out of date the day that it is delivered, and becomes obsolete quickly
Simplicity	• You can manage your cloud resources from a web portal that is easy to navigate. • Managing your **own private cluster may require sophisticated** system • administration skills

CONS	Description
Cost	• You pay for public cloud by the hour and the byte. • Computing the total cost of ownership of a cluster of machines housed in a university lab or data center is not easy. • In many environments, power and system administration are subsidized by the institution. • If you need to pay only for hardware, then running your own • cluster may be cheaper than renting the same services in • the cloud.
Variety	• The cloud does not provide every type of **computing that you may require, at least not today.** • In particular, it is not a proper substitute for a large supercomputer
Security	Your research concerns **highly sensitive data**, such as medical information, that cannot be moved outside your firewall
Dependence	• Dependence on one cloud vendor (often referred to as vendor lock-in). • As the public clouds converge in many of their standard offerings and compete on price, moving applications between cloud vendors has become easier

Sample Scenario

- In today's world, there are various use cases for **Serverless technologies**. Let's take a simple example, imagine you are the manager of **Coca-Cola** , one of the key capabilities of **Coca-Cola** is allowing users to have **serverless** after its implementation in vending machines resulted in significant savings. Whenever a beverage is purchased the details need to be shared to organization like user name , payment details etc. Serverless model ignoring the entire complexity of the application.

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Flow of the Application

- Amazon Web Services (AWS) serverless architecture, the new contactless Coca-Cola Freestyle solution enables consumers to **choose and pour drinks from their phones in just a few seconds, without having to create an account or download an app**.
- The mobile experience is currently rolling out to all Coca-Cola Freestyle dispensers across the United States.

- "**Holding your camera up to the display will autoscan a** QR code, which immediately connects to the cloud and brings the Coca-Cola Freestyle user interface to your phone,".

- "You then select from the **full menu of brands and flavors**—and pour by simply pressing the option on your phone. The idea is to be safe, seamless, and fun."

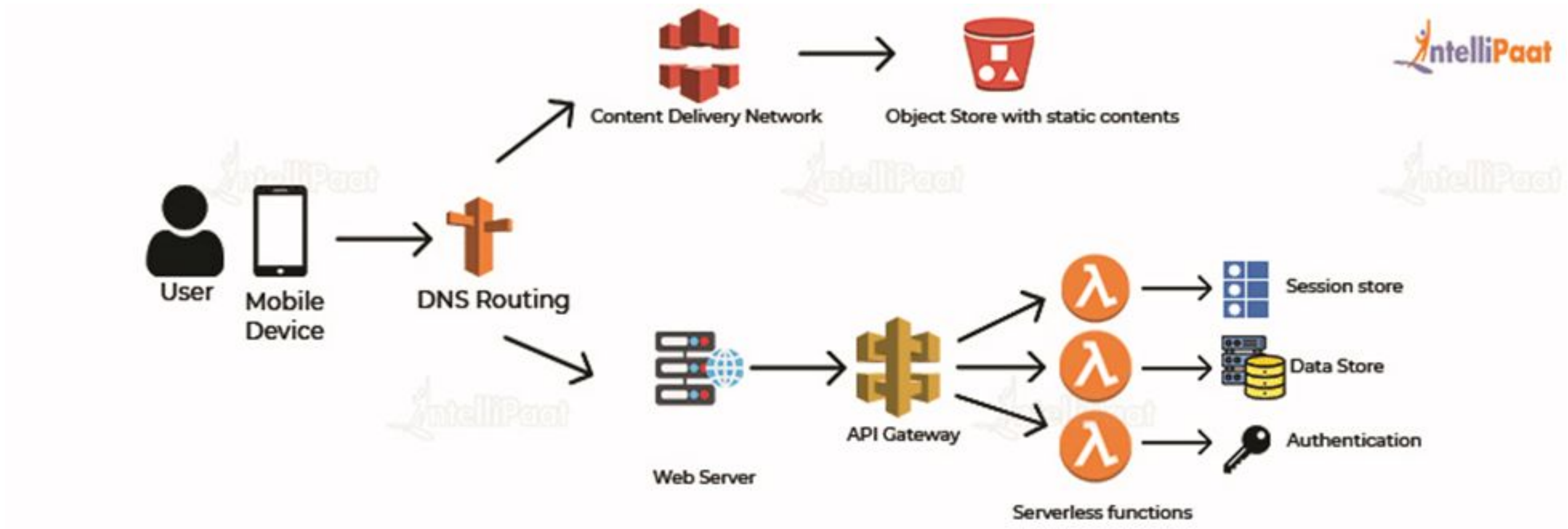
- The Freestyle team created a **serverless web app**—while working remotely, no less—that integrates with the Coca-Cola Freestyle machines to deliver a touchless user experience.
- Coca-Cola Freestyle deployed the frictionless, near-real-time solution less than 4 months after the initial idea.

AWS based services enabled for all scenario based questions

- **AWS Lambda :** Run code without thinking about servers or clusters **(MANAGE COMPUTE)**
- **Amazon S3 :** Object storage built to retrieve any amount of data from anywhere **(MANAGE STORAGE)**
- **Amazon API Gateway :** Create, maintain, and secure APIs at any scale.**(MANAGE API)**
- **AWS Fargate :**Serverless compute for containers .(**MANAGE CONTAINERS)**
- **Amazon DynamoDB:** Fast, flexible NoSQL database service for single-digit millisecond performance at any scale **(MANAGES NOSQL)**

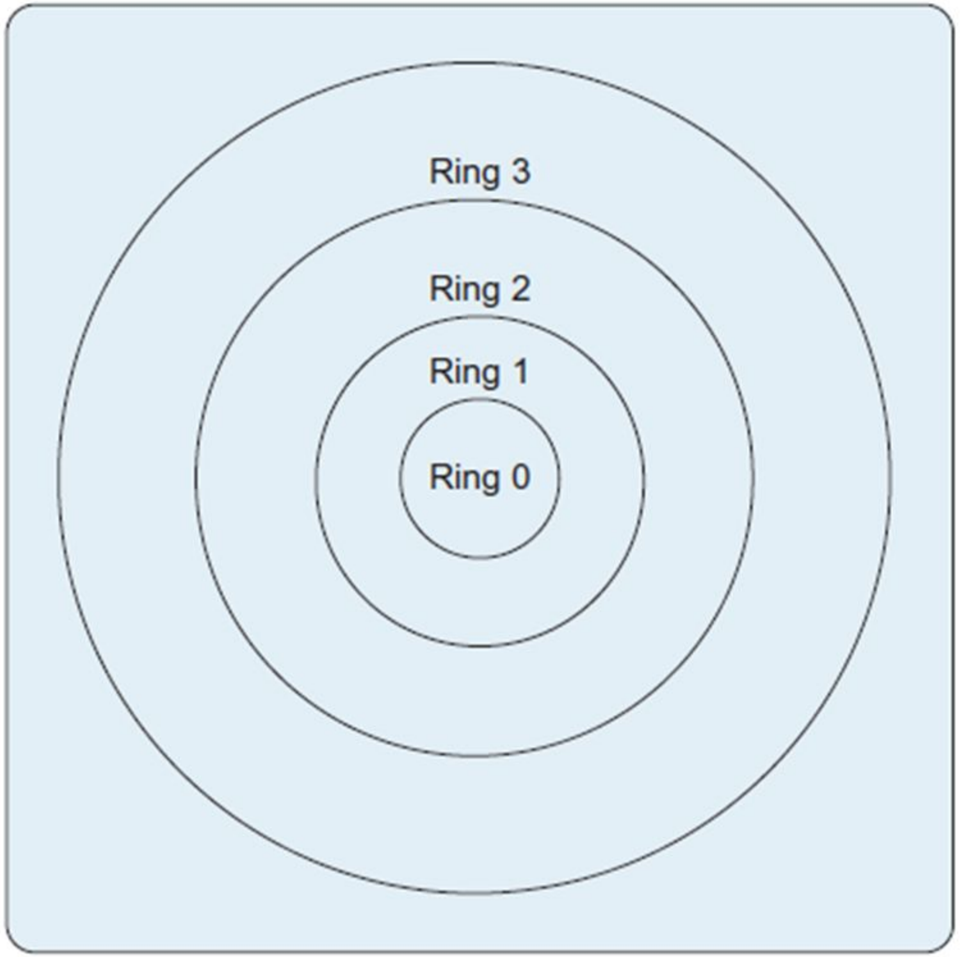
- **Amazon RDS :**Set up, operate, and scale a relational database in the cloud with just a few clicks. (**MANAGE SQL)**
- **Amazon SNS :**Fully managed Pub/Sub service for A2A and A2P messaging **(MANAGE MESSAGING)**
- **Amazon Virtual Private Cloud (Amazon VPC):**Define and launch AWS resources in a logically isolated virtual network.(**MANAGE PRIVACY)**
- **Amazon Athena :**Analyze petabyte-scale data where it lives with ease and flexibility **(MANAGE ANALYTICS)**

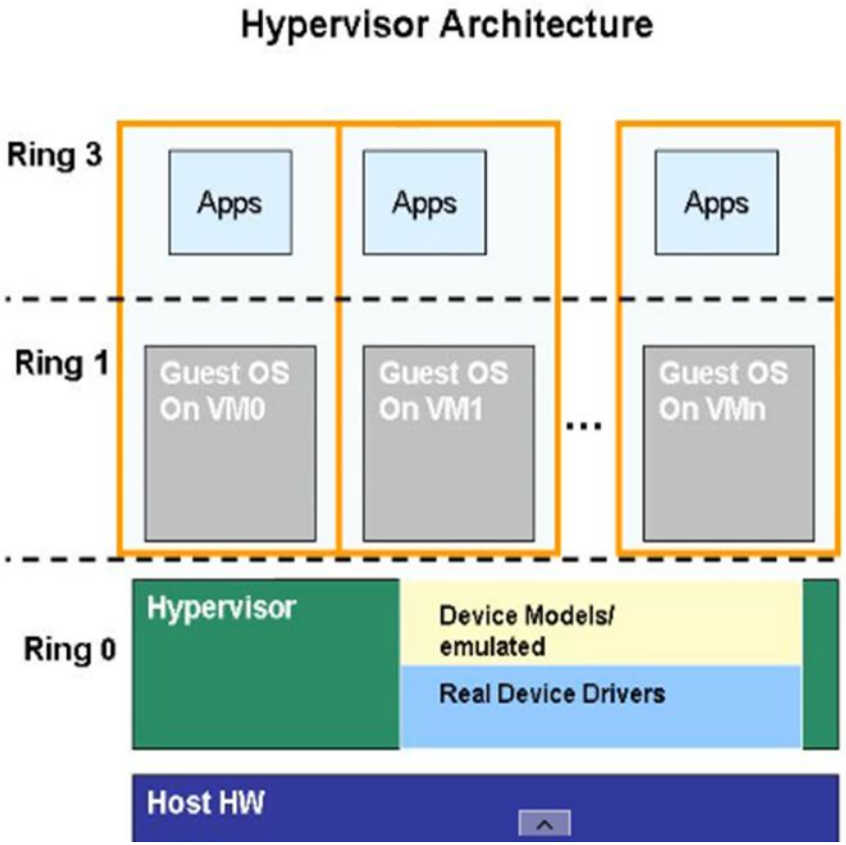
- **Amazon EMR:** Easily run and scale Apache Spark, Hive, Presto, and other big data workloads **(MANAGE BIGDATA COMPUTE)**
- **Amazon Kinesis:** Easily collect, process, and analyze video and data streams in real time **(MANAGE REAL TIME DATA)**


Content Delivery Network
Object Store with static contents
IntelliPaat
User
Mobile Device
DNS Routing
Web Server
API Gateway
Session store
Data Store
Authentication
Serverless functions

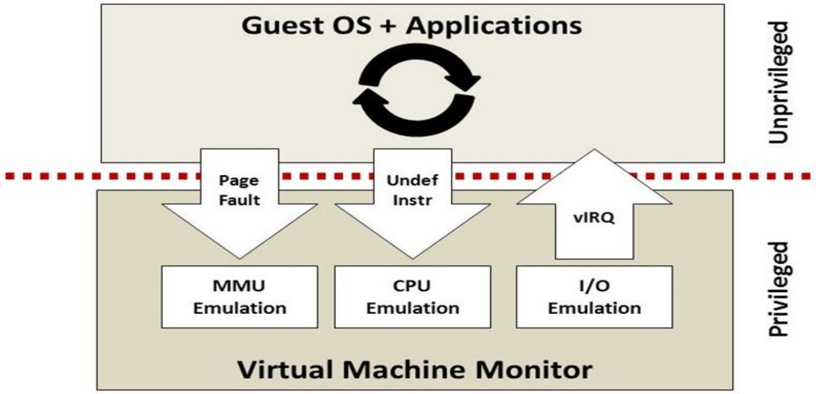
Historical Roots

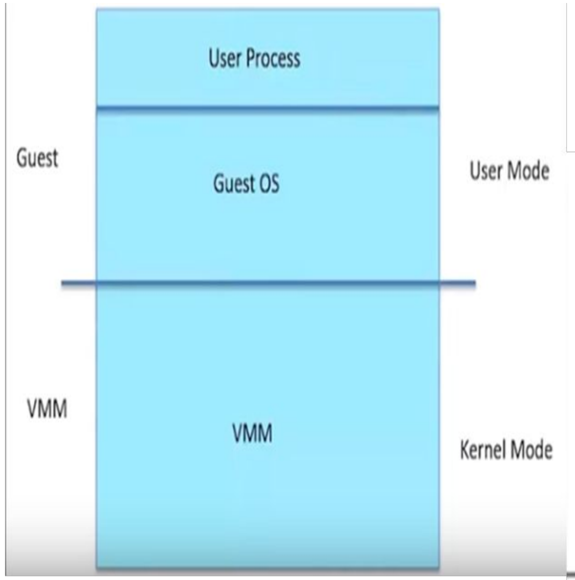
- **Registers, memory addressing mechanisms, and I/O and network interfaces**.
- The programs that control the computer are just sequences of binary codes corresponding to instructions that manipulate these resources
- There are also important instructions for performing context switches, in which the computer stops executing one program and starts executing another.
- These state management instructions plus the I/O instructions are termed privileged.
- Such instructions are usually directly executed only by the OS, because you do not want users to be able to access state associated with other computations

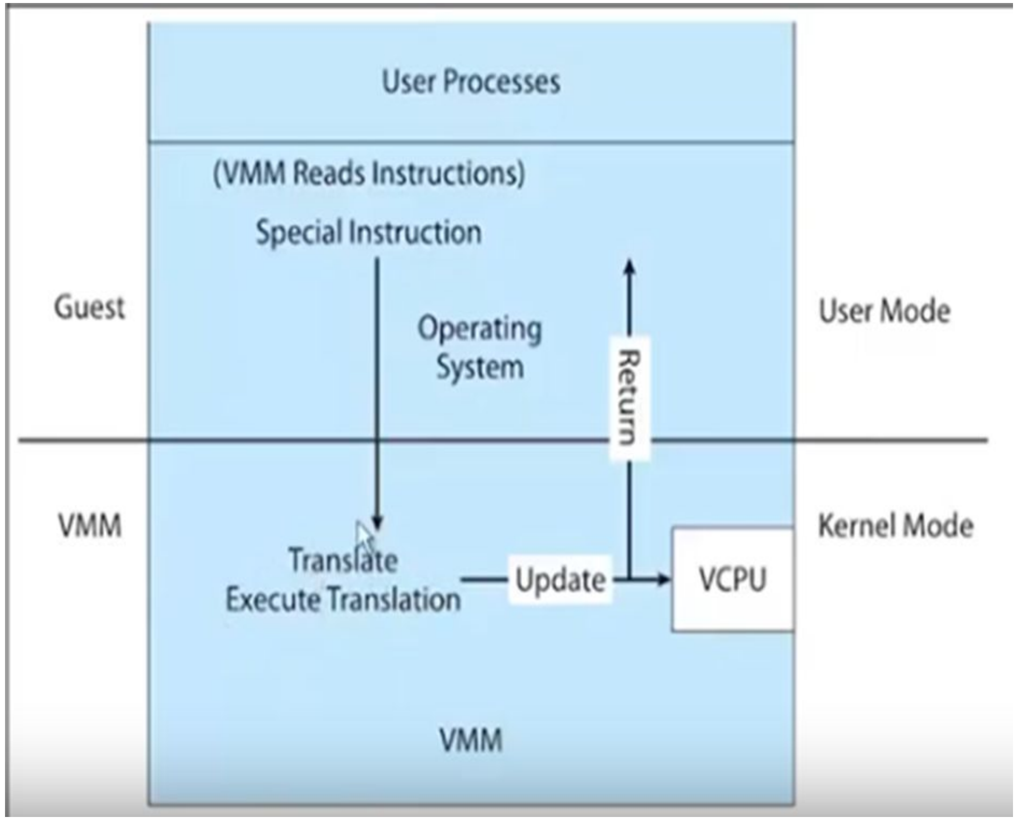

Ring 3
Ring 2
Ring 1
Ring 0

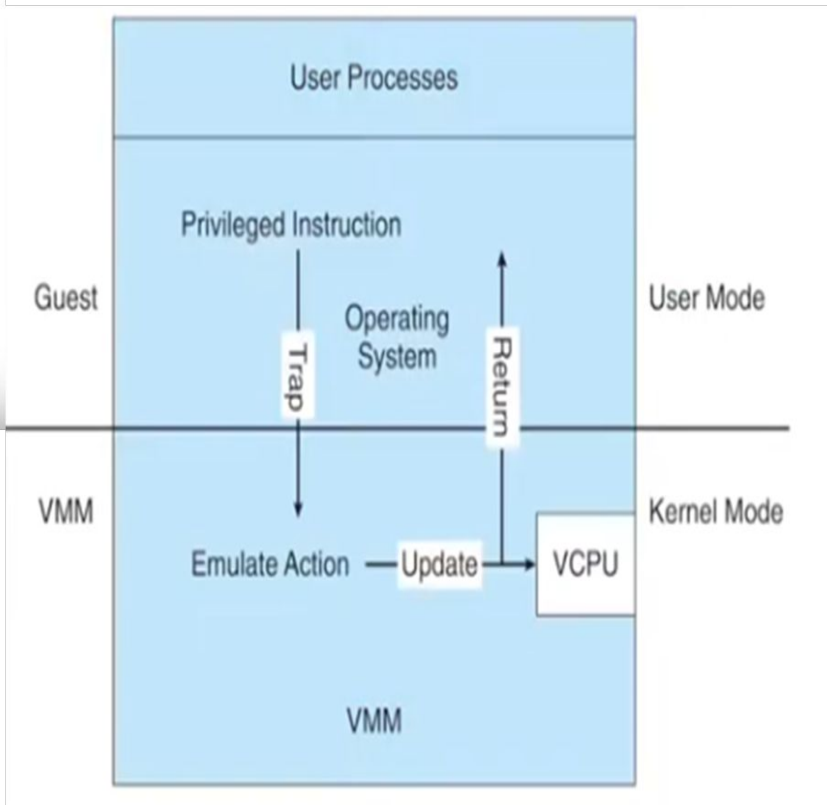

Hypervisor Architecture
Ring 3
Apps
Apps
Apps
Ring 1
Guest OS On VM0
Guest OS On VM1
...
Guest OS On VMn
Ring 0
Hypervisor
Device Models/ emulated
Real Device Drivers
Host HW

Trap and Emulate







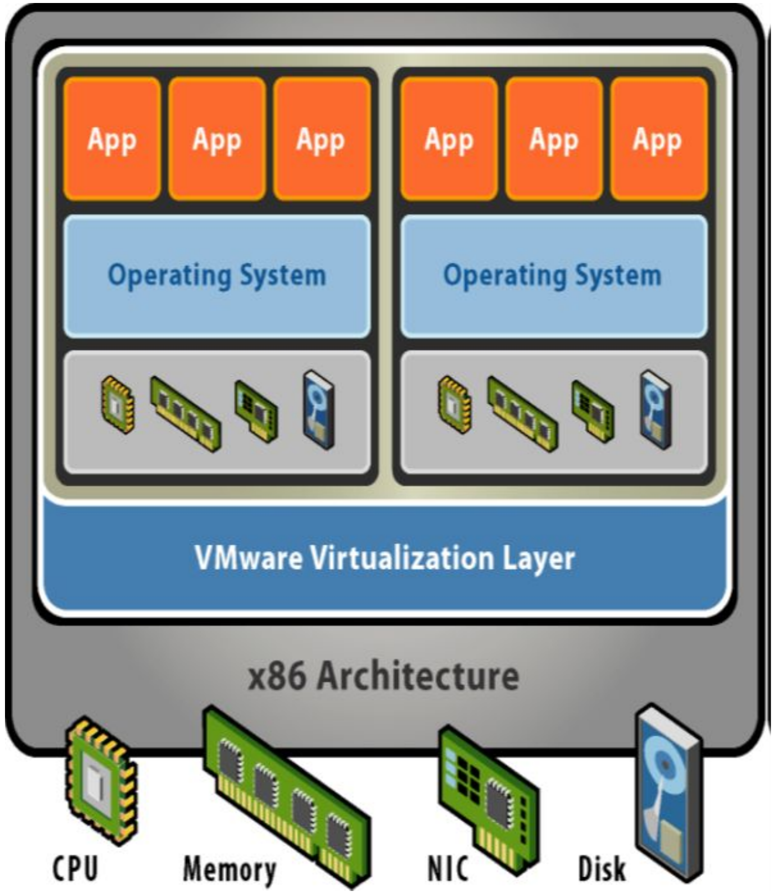


- **TRAP AND EMULATE**
- When this happens, the hypervisor regains control and mediates access to the hardware, or handles the interrupt.
- Hypervisors rely on a feature of modern processors known as the **PRIVILEGE LEVEL OR PROTECTION RING**

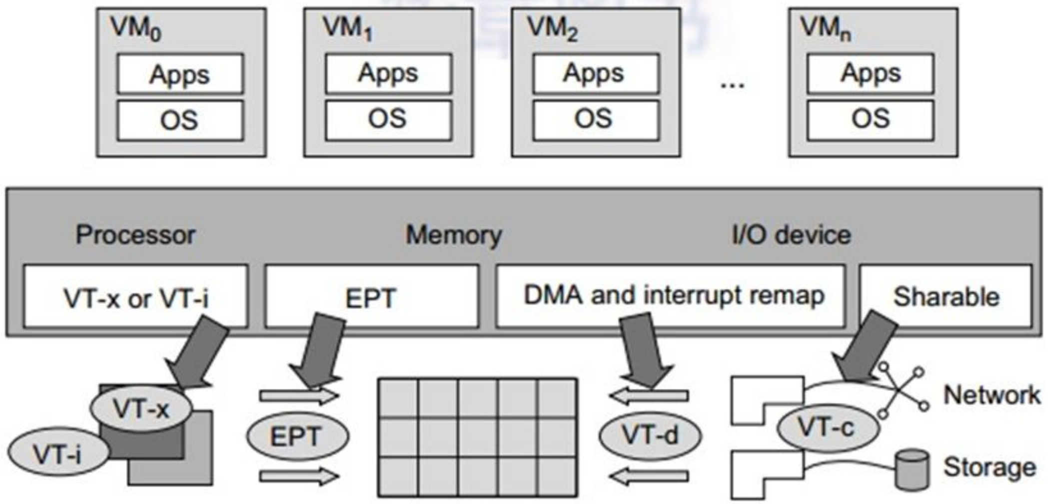
- **PRIVILEGE LEVEL OR PROTECTION RING**
    - *PRIVILEGED INSTRUCTIONS*
    - *SENSITIVE INSTRUCTIONS*
- BINARY TRANSLATION

- **PRIVILEGE LEVEL OR PROTECTION RING**
- 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**.
- **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
- **Current privilege level (CPL) register** of the processor to 3 before starting execution of the guest.
- 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 x86 architecture provides four levels of protection, 0,1,2,3, with **0 used by the kernel**, **3 by application software,** and **1 and 2 unused**



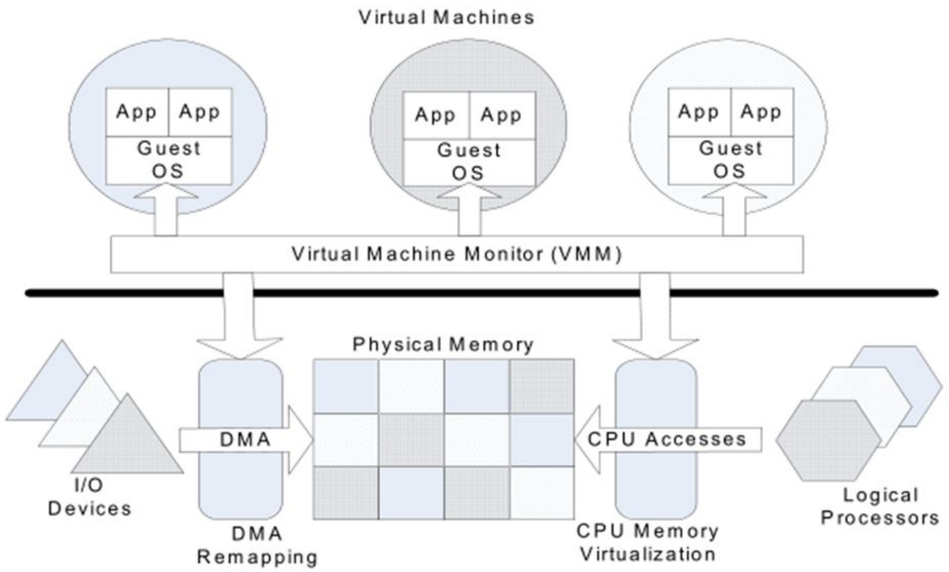
- **Hardware Support for Virtualization**
  - **Vt-x**, an intel technology that helps virtualize intel x86 processors.
  - Overview of a technique called **EXTENDED PAGE TABLES (EPT)** which helps virtualize memory
  - Followed by **VT-D,** a technology to assist in the virtualization of **i/o.**



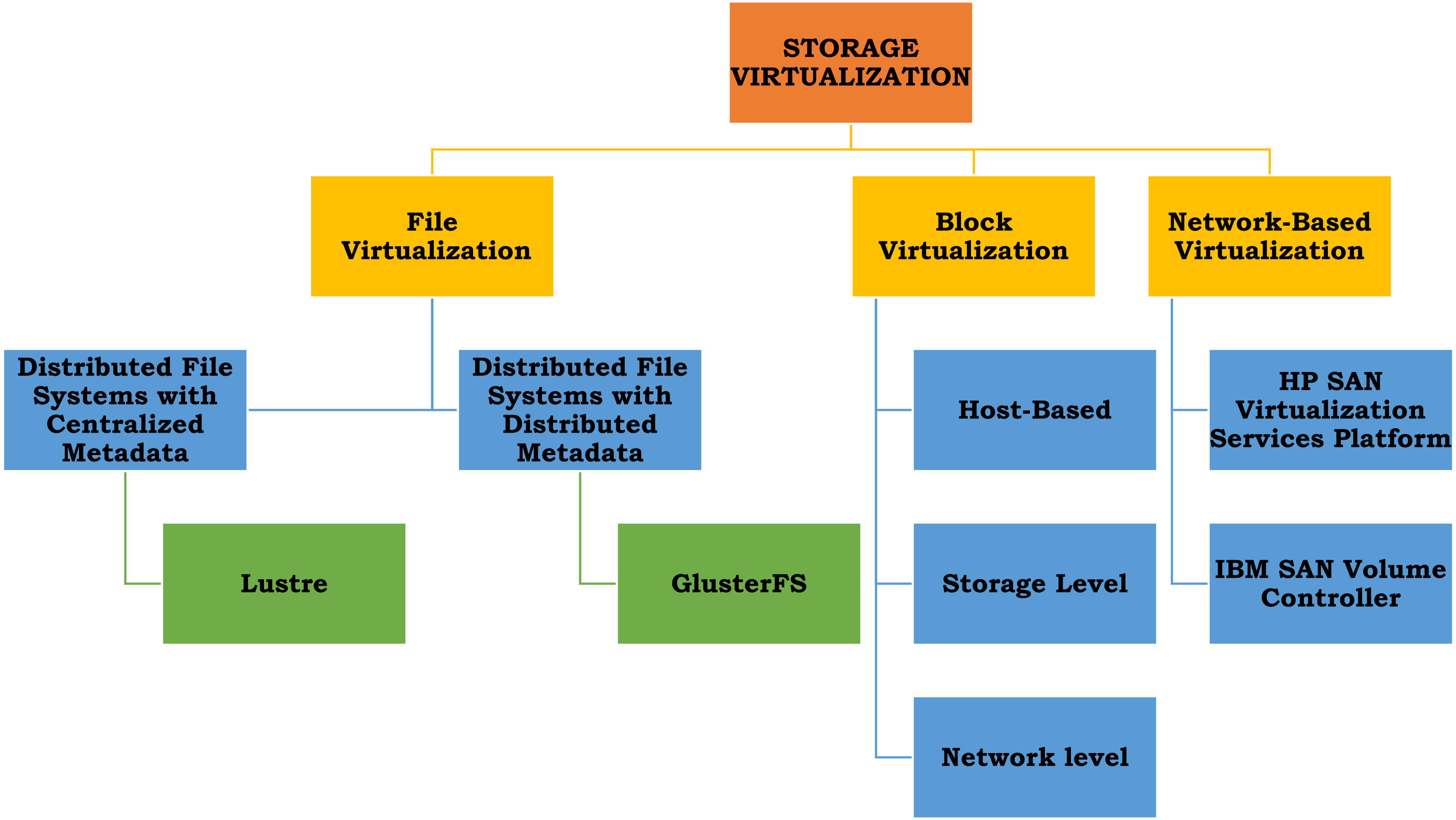
- *Hardware Support for Processor Virtualization*
- ***VMX root operation and VMX non-root operation***
- VT-x makes use of a new data structure called **the Virtual Machine Control Structure (VMCS).**

- Hardware Support for Memory Virtualization
- Reduce overheads of memory virtualization using
- Extended Page Tables (EPT)
- Virtual Processor ID (VPID)

- Hardware Support for IO Virtualization
- **VM INITIALIZATION**
- **VM OPERATION**
    - **INTERRUPT REMAPPING**
    - **DMA REMAPPING**



General Classification



Network-Based Virtualization

- Network Virtualization (NV) refers to **abstracting network resources** that were traditionally delivered in hardware to software.
- Fibre Channel Storage Area Network (SAN).
- There are broadly two categories based on where the virtualization functions are implemented:
- either in Switches (Routers) Or In Appliances (Servers).

AWS Elastic Compute Cloud

as instances

Preconfigured

Amazon Machine Images (AMIs)

configurations of CPU, memory, storage, and networking

known as instance types

Secure login

(AWS stores the public key, and you store the private key in a secure place)

Storage volumes

known as instance store volumes

Persistent storage

Amazon Elastic Block Store (Amazon EBS), known as Amazon EBS volumes

Multiple physical locations for your resources

known as Regions and Availability Zones

protocols, ports, and **source IP ranges that can reach your instances using security groups**

Static IPv4 addresses

cloud computing

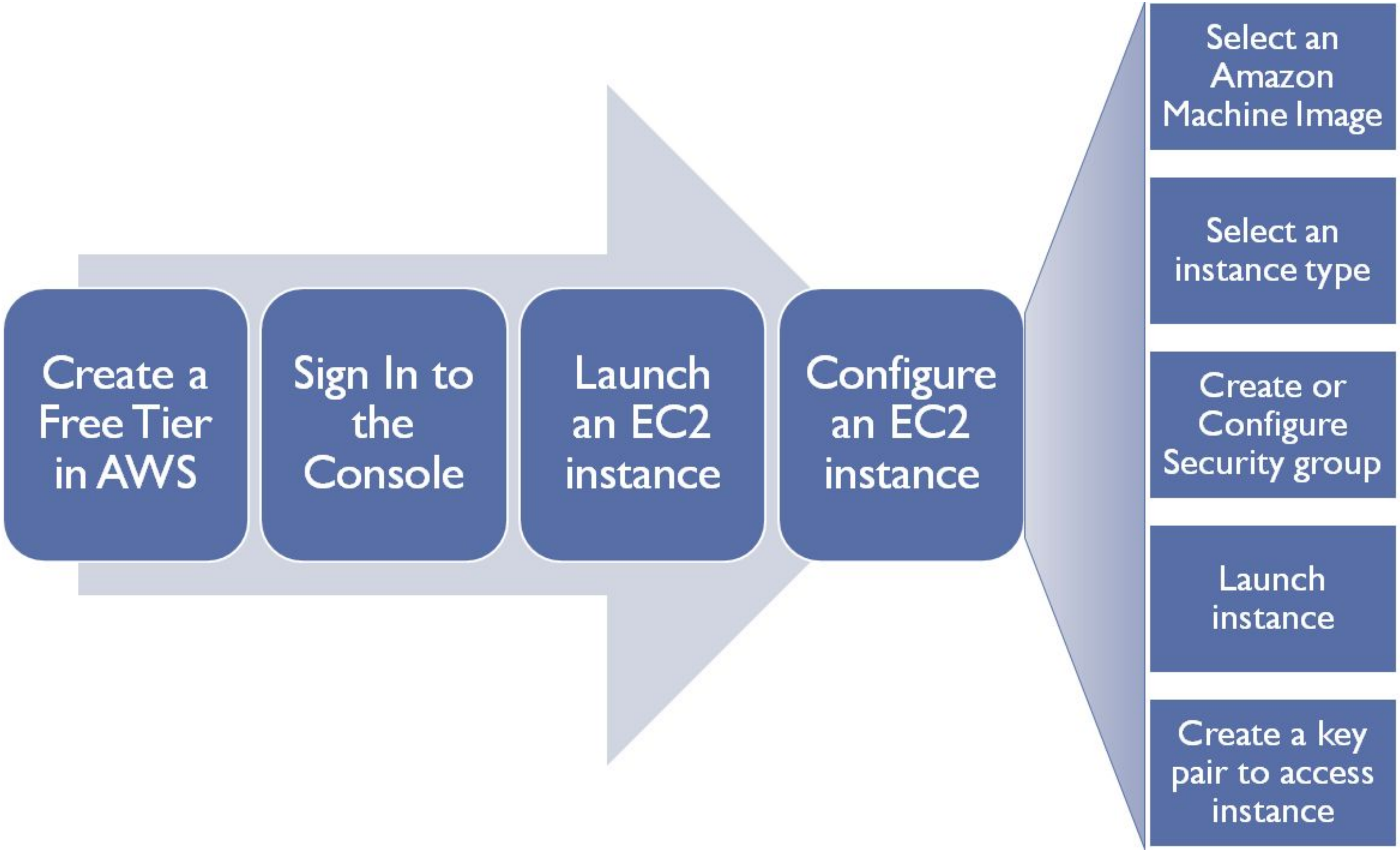
Metadata,

TAGS,

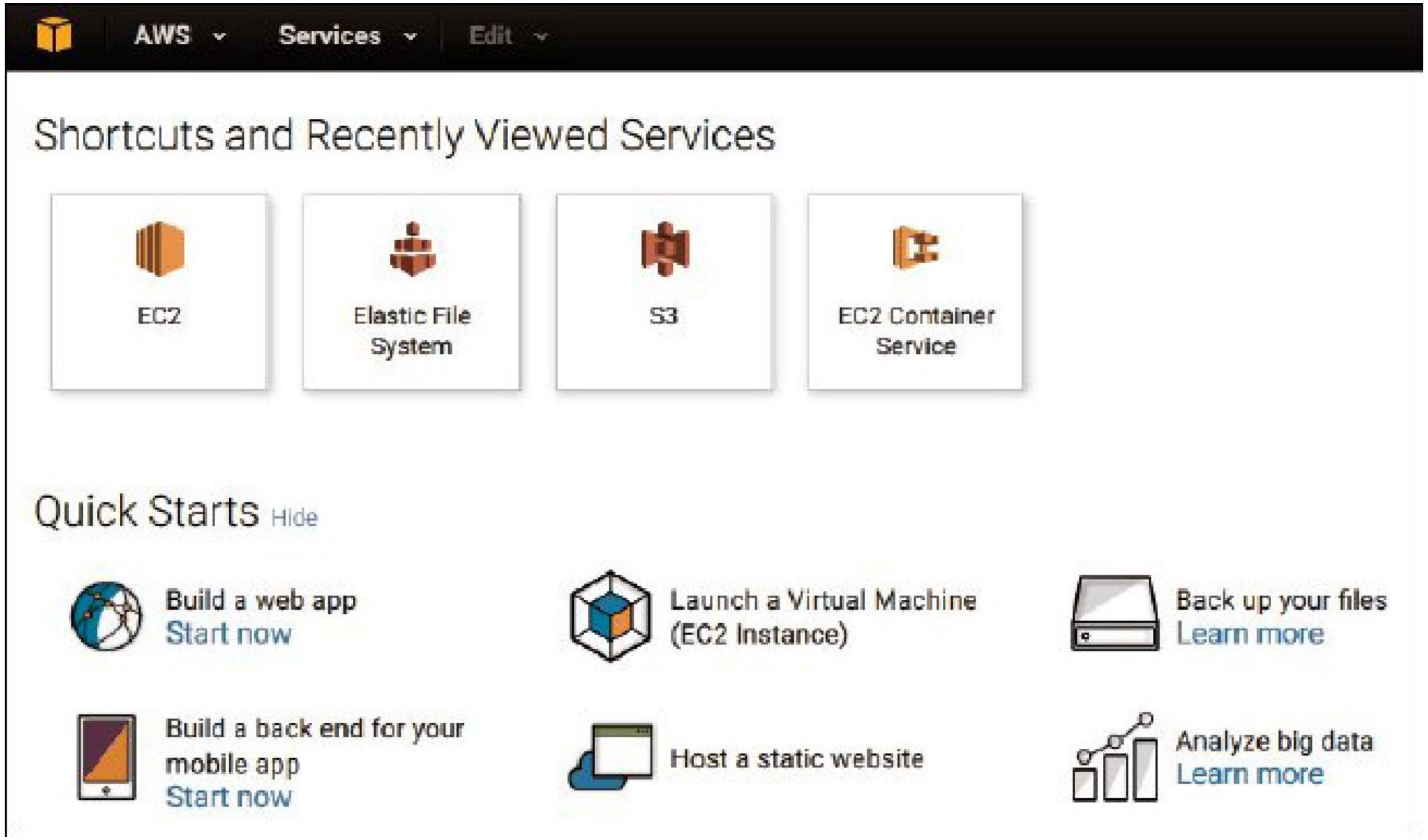
and assign to your Amazon EC2 resources

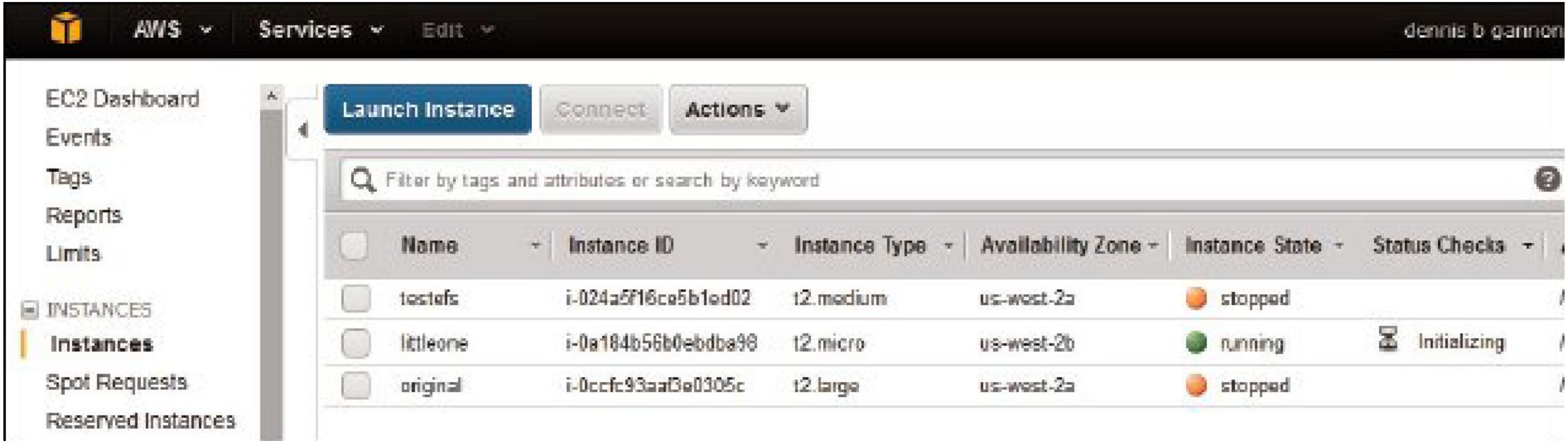
AWS Cloud

known as virtual private clouds (VPCs)


Create a Free Tier in AWS
Sign In to the Console
Launch an EC2 instance
Configure an EC2 instance
Select an Amazon Machine Image
Select an instance type
Create or Configure Security group
Launch instance
Create a key pair to access instance

1. Click on EC2 or launch a virtual machine
2. Specify desired host service **instance type**
3. The number of cores that our VM is to use, the required **memory size, and network performance**.
4. The launch process involves providing a key pair: the **CRYPTOGRAPHIC KEYS** that you use to access your running instance.
5. To create a **key pair early in the process**
6. The **PRIVATE KEY FILE AND PUBLIC KEY KEY PAIR**
7. The public key is loaded into **the instance other important choices involve storage options and security groups monitor its status**
8. The **STATUS CHECKS** shows that the new instance is still initializing.
9. After a few moments, its status changes to a green check mark to indicate that the instance is ready to launch.


AWS
Services
Edit
Shortcuts and Recently Viewed Services
EC2
Elastic File System
S3
EC2 Container Service
Quick Starts Hide
Build a web app
Start now
Launch a Virtual Machine (EC2 Instance)
Back up your files
Learn more
Build a back end for your mobile app
Start now
Host a static website
Analyze big data
Learn more


AWS
Services
Edit
dennis b gannon
EC2 Dashboard
Events
Tags
Reports
Limits
INSTANCES
Instances
Spot Requests
Reserved Instances
Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
testefs
i-024a5f16ce5b1ed02
t2.medium
us-west-2a
stopped
littleone
i-0a184b56b0ebdba98
t2.micro
us-west-2b
running
Initializing
original
i-0ccfc93aaf3e0305c
t2.large
us-west-2a
stopped

- To connect to your instance, you need to use a **secure shell command.**
- On Windows the tool to use is **called PuTTY**
- Used as a **File Transfer Protocol**.
- Used to **generate Hash key**
- ec2-user@IPAddress, where the IPAddress is the IP address you can find in the Portal Instance View

```
ssh -i path-to-your-private-key.pem ec2-
user@ipaddress-of-instance
```

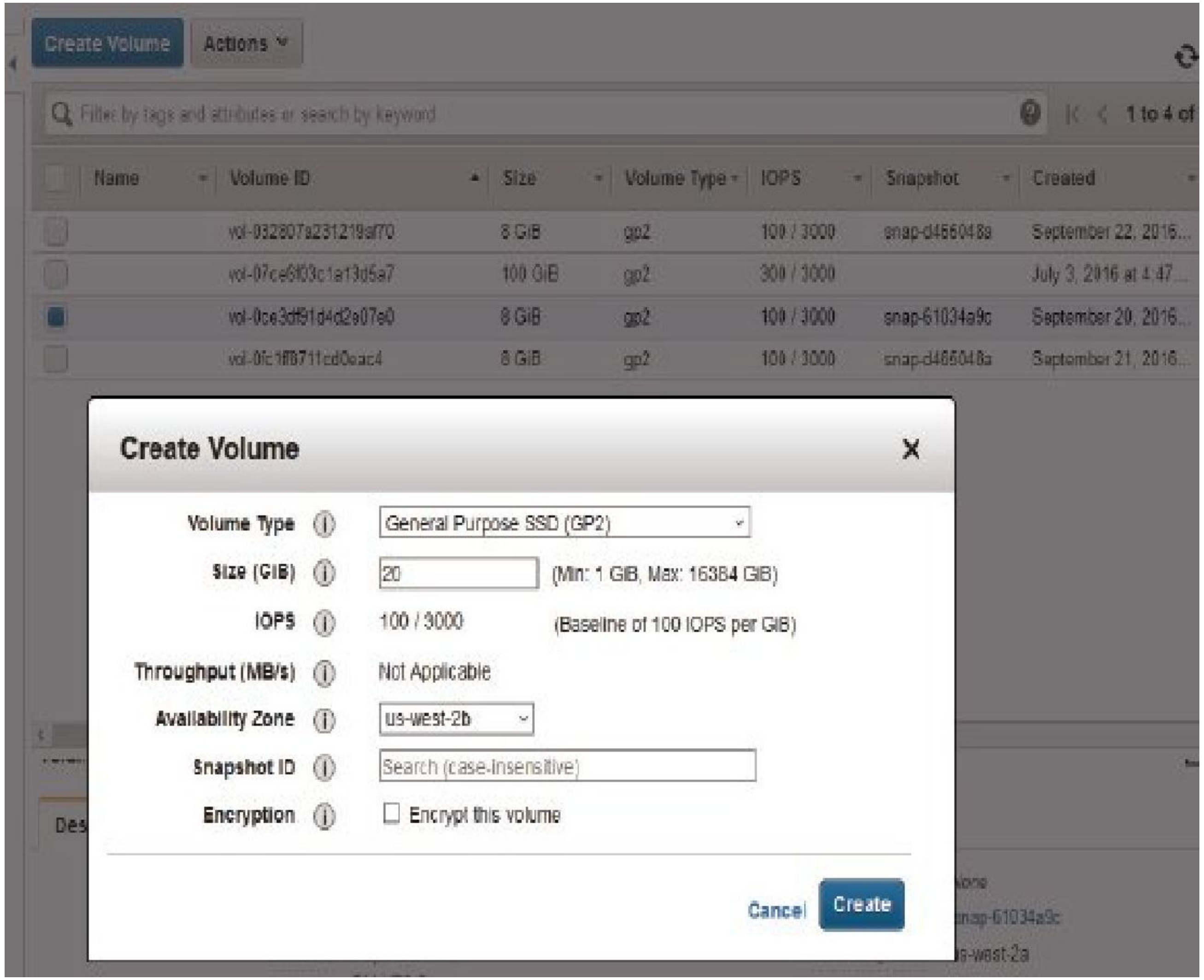
```
import boto3
ec2 = boto3.resource('ec2', 'us-west-2')
ec2.create_instances(ImageId='ami-7172b611',
't2.micro',
                  MinCount=1, MaxCount=1)
```

- The following listing uses the **Python Boto3 SDK** to create an Amazon EC2 VM instance.
- It creates an ec2 resource, which requires your **aws_access_key_id and aws_secret_access_key**, unless you have these stored in your .aws directory.
- The **ImageId argument specifies** the VM image that is to be started and the MinCount and MaxCount arguments the number of instances needed.

```
# A function that lists instances with a
specified status
def show_instance(status):
    instances = ec2.instances.filter(
        Filters=[{'Name':'instance-state-
        name','Values':[status]}])
    for instance in instances:
        print(instance.id, instance.instance_type,
            instance.image_id,
            instance.public_ip_address)

show_instance('running')
('i-0a184b56b0ebdba98', 't2.micro', 'ami-
7172b611', '146.137.70.71')
```

Attaching Storage

- VM: instance storage, **Elastic Block Store, and Elastic File System.**
- Instance storage is what comes with each VM instance.
- **Elastic Block Store (EBS)** storage independent of a VM and then attach it to a running VM.
- EBS volumes persist and thus are good for databases and other data collections that we want to keep beyond the life of a VM.
- To create an EBS volume, go to the volumes tab of the EC2 Management console and click **Create Volume**



- EBS used to be accessible to a **single EC2 instance** only, making it most like your physical hard drive.

- EBS is a **block storage service**, which means all data within EBS is stored in equally sized blocks.

- EFS can be **mounted by multiple EC2 instances,** meaning many virtual machines may store files within an EFS instance.

- EFS is a **file storage system**.

- Selected the us-west-2b **availability zone**.
- The actions tab in the volume management console

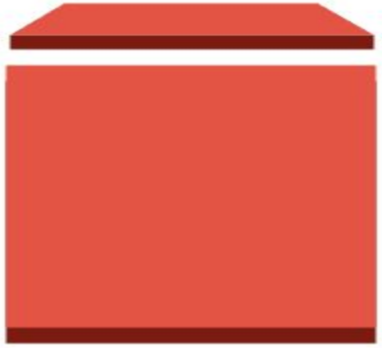
```
In [3]: vols = ec2.volumes.filter(Filters=[])
In [4]: for vol in vols:
            print (vol.id, vol.size, vol.state)
        ('vol-032807a231219af70', 8, 'in-use')
        ('vol-0bdd0584d0833e691', 20, 'available')
        ('vol-07ce6f03c1a13d5a7', 100, 'in-use')
        ('vol-0ce3df91d4d2e07e0', 8, 'in-use')
        ('vol-0fc1ff8711cd0eac4', 8, 'in-use')
```

```
In [5]: vol = ec2.Volume('vol-0bdd0584d0833e691')
In [6]: vol.attach_to_instance(InstanceId='i-
0a184b56b0ebdba98',
                         Device='/dev/xvdh' )
     {u'AttachTime':
     datetime.datetime(2016,9,23,18,15,49,308000
     ,
                                  tzinfo=tzutc
                                  ()),
      u'Device': '/dev/xvdh',
      … more attach metadata not shown …
     }
```

```
In [7]: def myexec( pathtopem, hostip, commands):
    ssh = subprocess.Popen(['ssh', '-i',
    pathtopem,
                'ec2-user@%s'%hostip, commands],
        shell=False, stdout=subprocess.PIPE,
        stderr=subprocess.PIPE)
    result = ssh.stdout.readlines()
    if result == ['error']:
        error = ssh.stderr.readlines()
        print >>sys.stderr, "ERROR: %s" % error
        return "error"
    else:

        return result
```

- EBS is a high-performance **per-instance block storage system** designed to act as storage for a **single EC2 instance (most of the time)**
- EFS is a **highly scalable file storage system** designed to provide flexible storage for multiple EC2 instances
- S3 is an **object storage system**, designed to provide archiving and data control options and to interface with other **services beyond EC2**. It's also useful for storing static **html pages and shared storage for applications**

- This function requires the path to your **private key, the IP address of the instance**, and the command script as a string.
- If you want a volume to be shared with multiple instances, then you can use the third type of instance storage, called **Elastic File System, that implements the Network File System (NFS) standard**



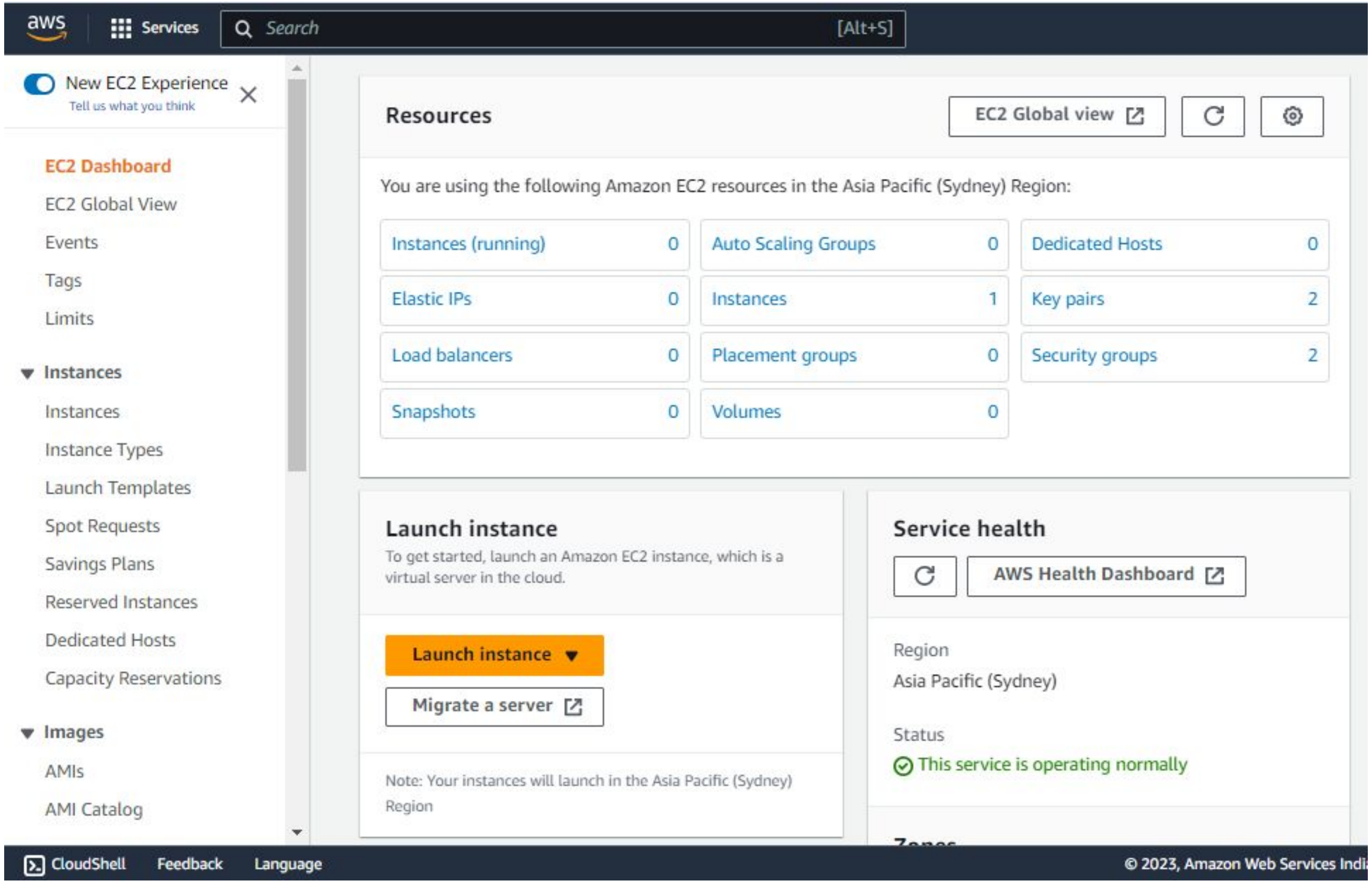
Amazon Elastic Block Storage (EBS)

Amazon Elastic File System (Amazon EFS)

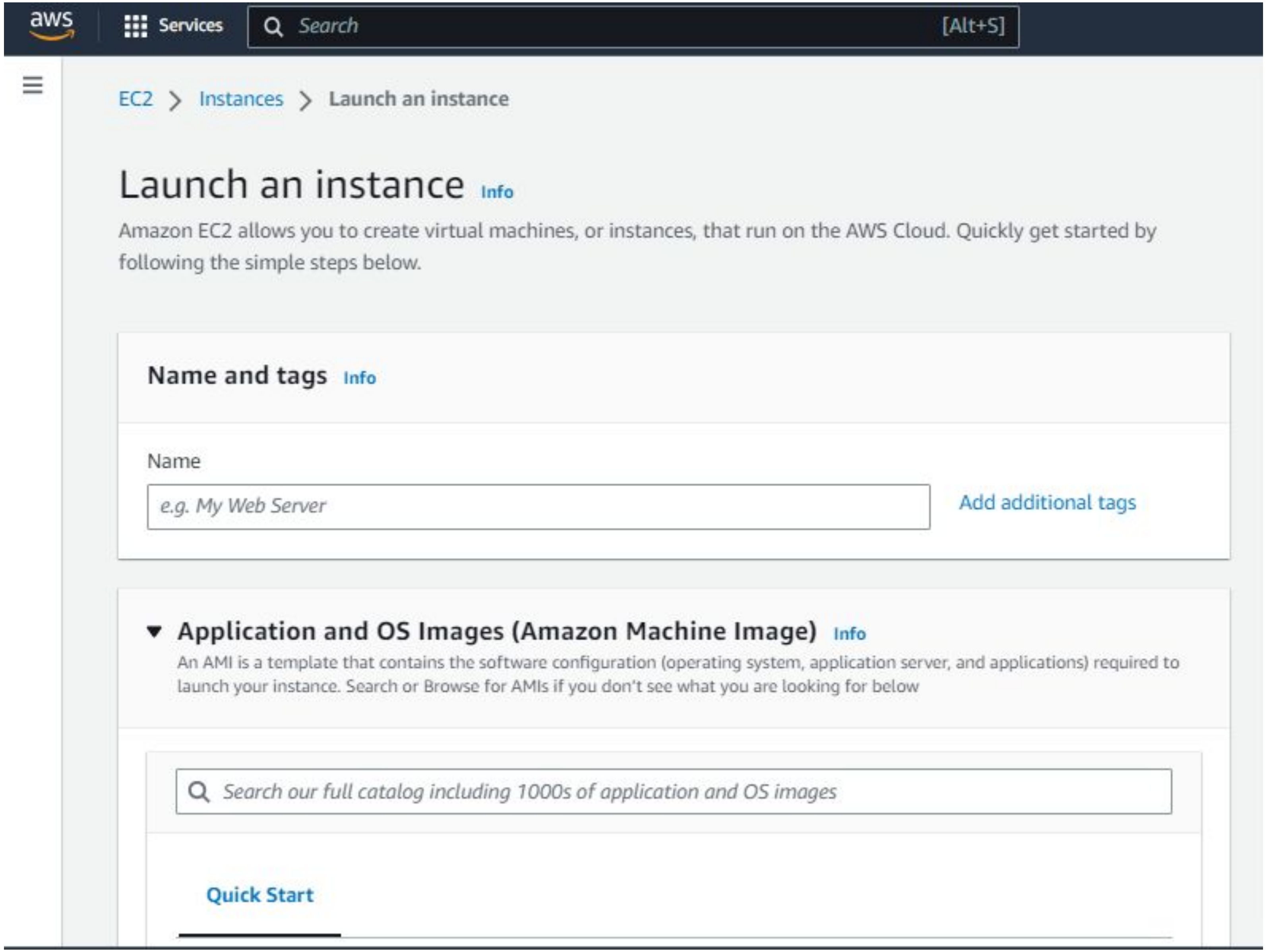
EC2 Computational Resources

Features	Descriptions
Computing resources	• The computing resources available on EC2, referred to as **EC2 instances**, consist of combinations of computing power, together with other resources such as memory. • EC2 Compute Unit (CU) is a standard measure of computing power in the same way that bytes are a standard measure of storage.
Software	• Amazon Machine Images (AMIs). • The required AMI has to be specified when requesting the EC2 instance, as seen earlier. The AMI running on an EC2 instance is also called the root AMI.
Operating systems	*Red Hat Enterprise Linux and SuSE, the Windows server, and Solaris*
Regions and Availability Zones	EC2 offers regions, which are the same as the S3 regions described in the section S3 Administration.
Load Balancing and Scaling	EC2 provides the Elastic Load Balancer, which is a service that balances the load across multiple servers

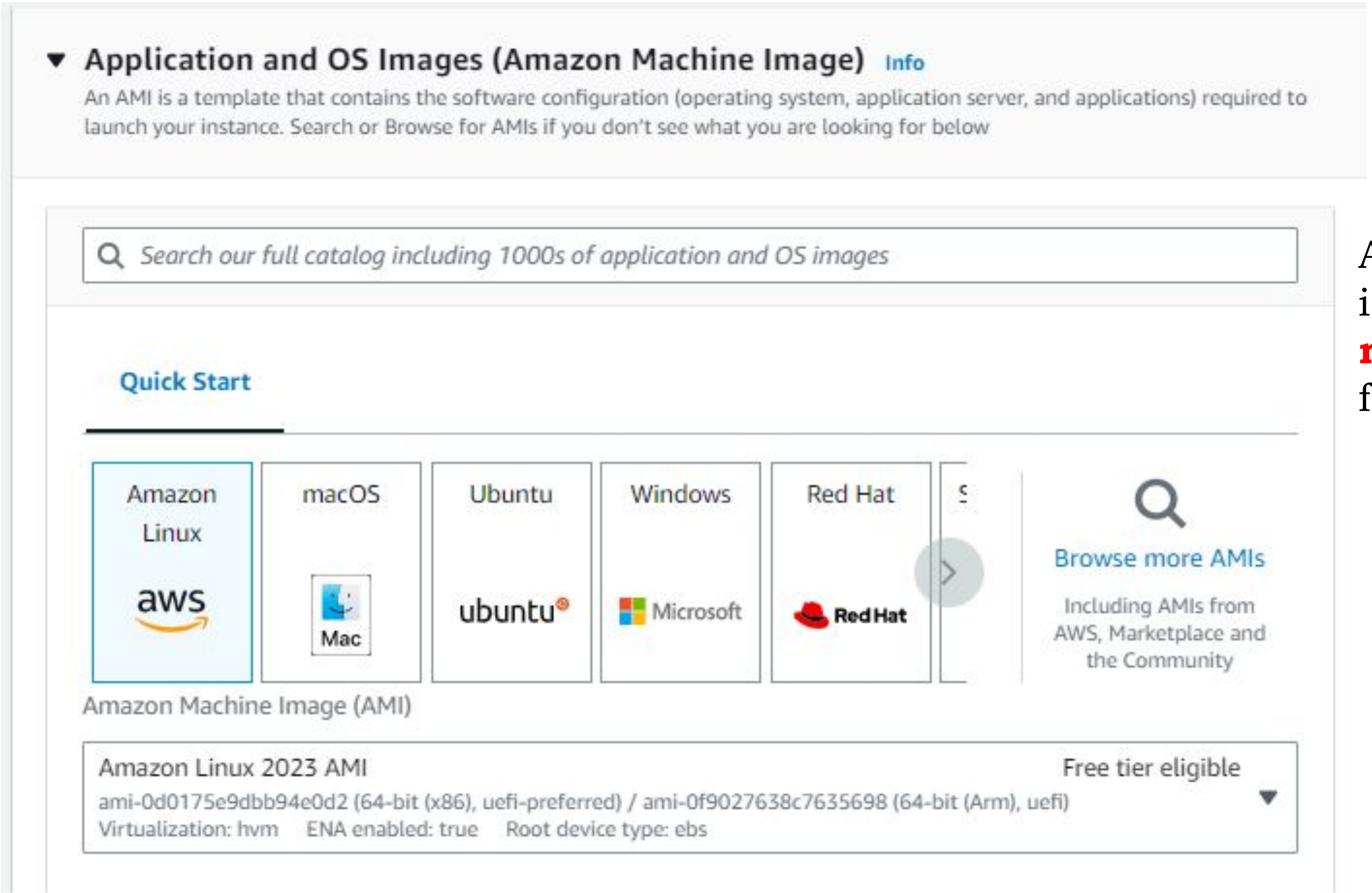
- Step 1: Sign in to the AWS Management Console



- Step 2: Choose a name of your instance

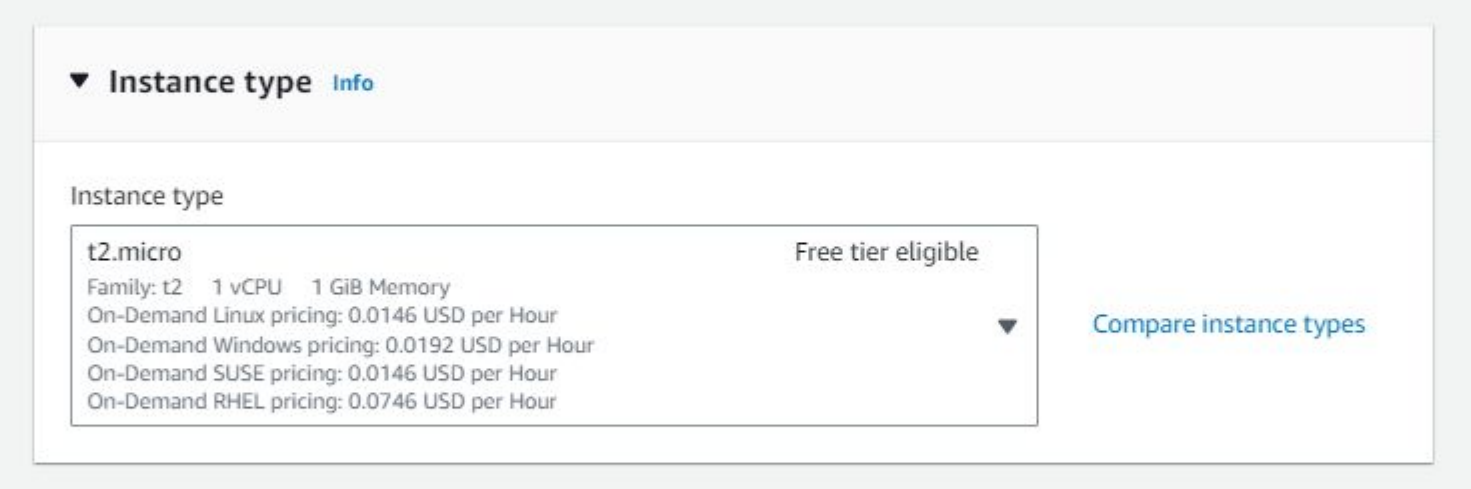


- Step 3: Choose an Amazon Machine Image (AMI)



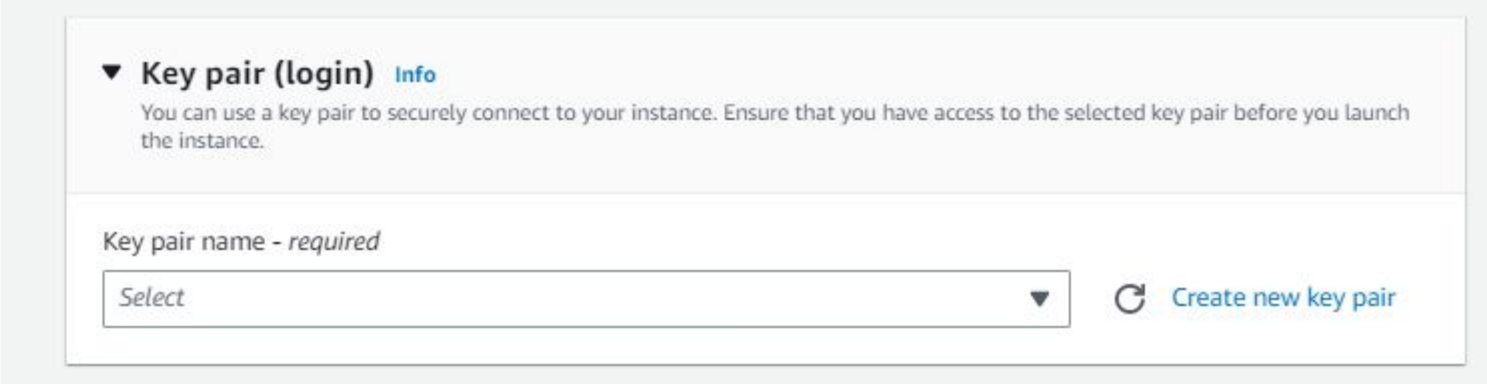
An Amazon Machine Image (AMI) is a **pre-configured virtual machine** that serves as a template for your EC2 instance.

- Step 4: Choose an Instance Type



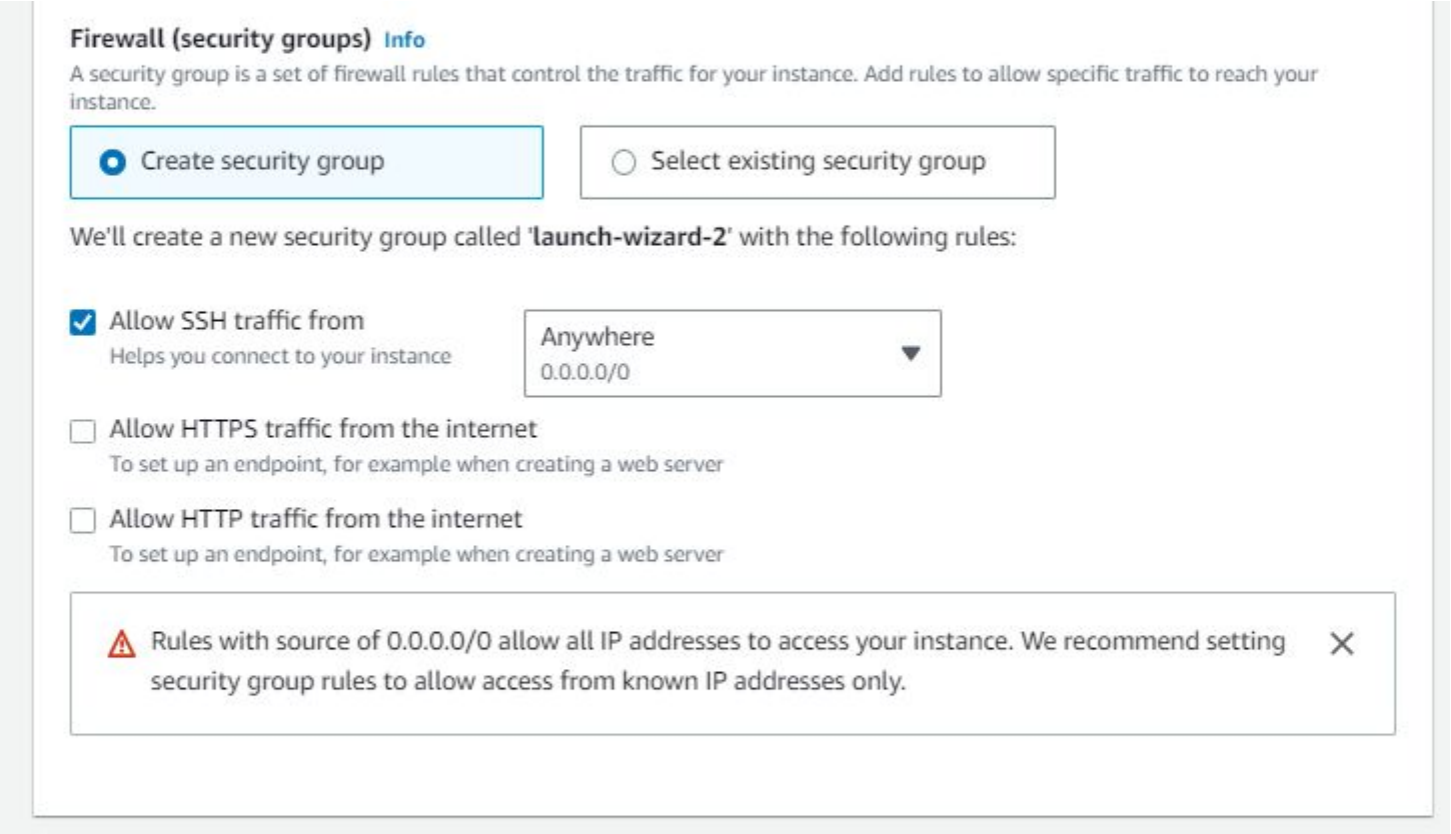
An instance type **determines the computing resources (CPU, RAM, storage, etc.) available to your EC2 instance.**

- Step 5: Create a key pair



Create a key pair if you have never created one and store it **in a safe place because it will act as a key to log in to your instance.**

• Step 6: Configure Security Group

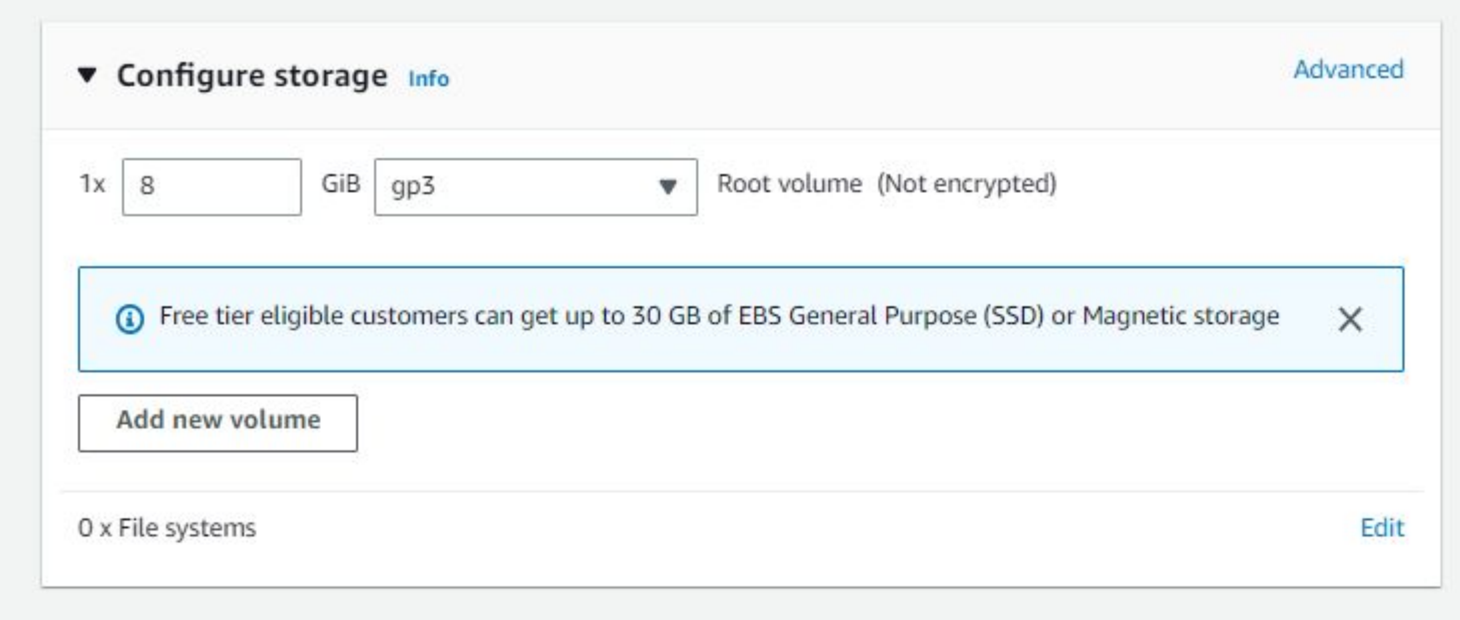


Security groups act as virtual firewalls for your EC2 instance, **controlling inbound and outbound traffic**.

We can **configure security groups** to allow or deny traffic from specific IP addresses, protocols, and ports.

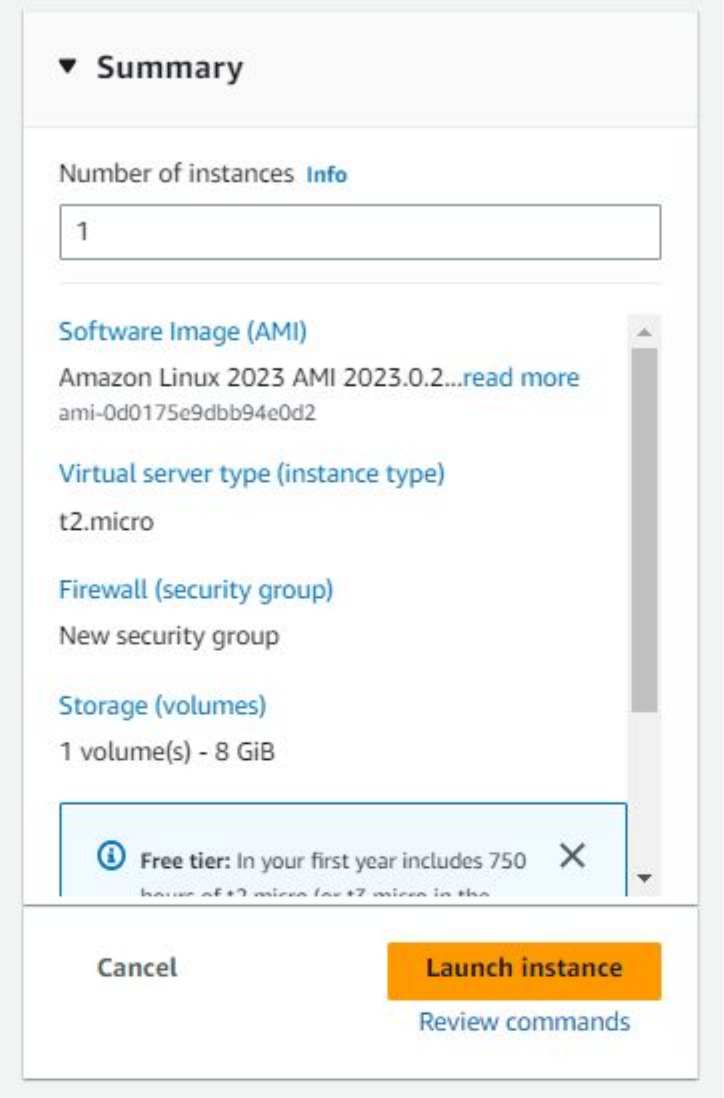
In this step, we need to create a new **security group or select an existing one.**

• Step 7: Add Storage



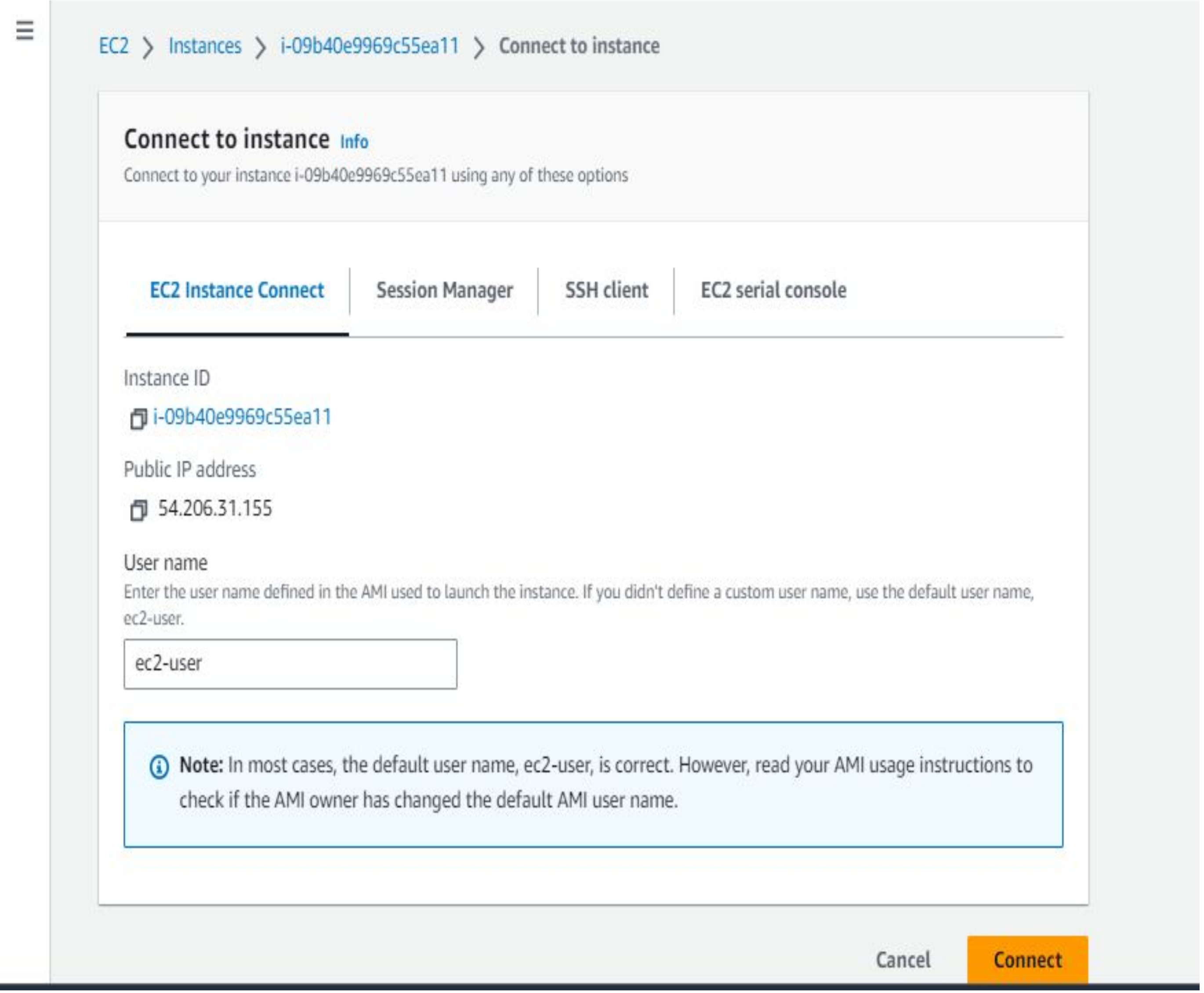
- EC2 instances require storage for the **operating system, applications, and data.**
- In this step, you can **add and configure storage volumes for your instance**.
- You can choose from different types of storage, including **Amazon Elastic Block Store (EBS) volumes and instance store volumes.**

Step 8: Review and Launch



- Before launching your instance, **review all the details to make sure everything is correct.**
- You can also modify any settings that need to be changed.
- Once you're ready, click the "Launch" button to start your EC2 instance.

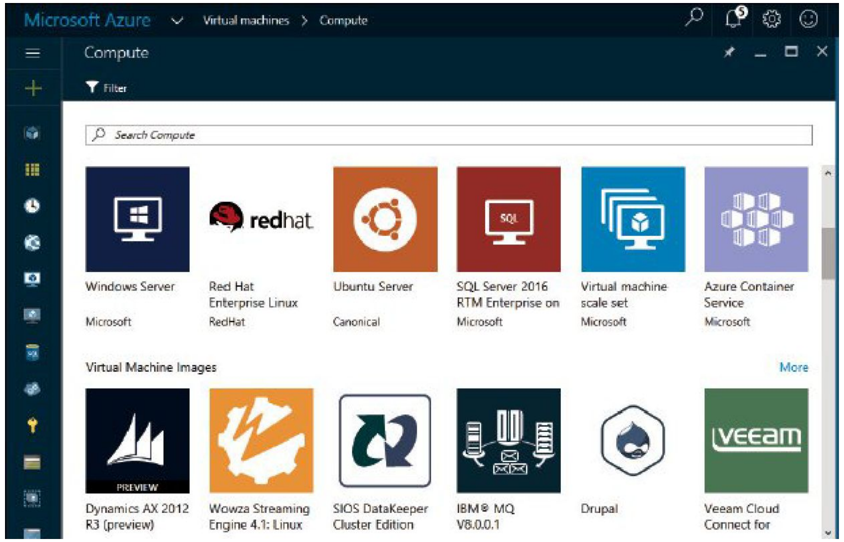
Step 9: Connect to Your Instance



- After launching your instance, you can connect to it using various **methods, such as SSH or Remote Desktop Protocol (RDP).**

- You can also use the AWS Systems Manager Session Manager to connect to your instance securely without the need for a public IP address.

Azure VMs

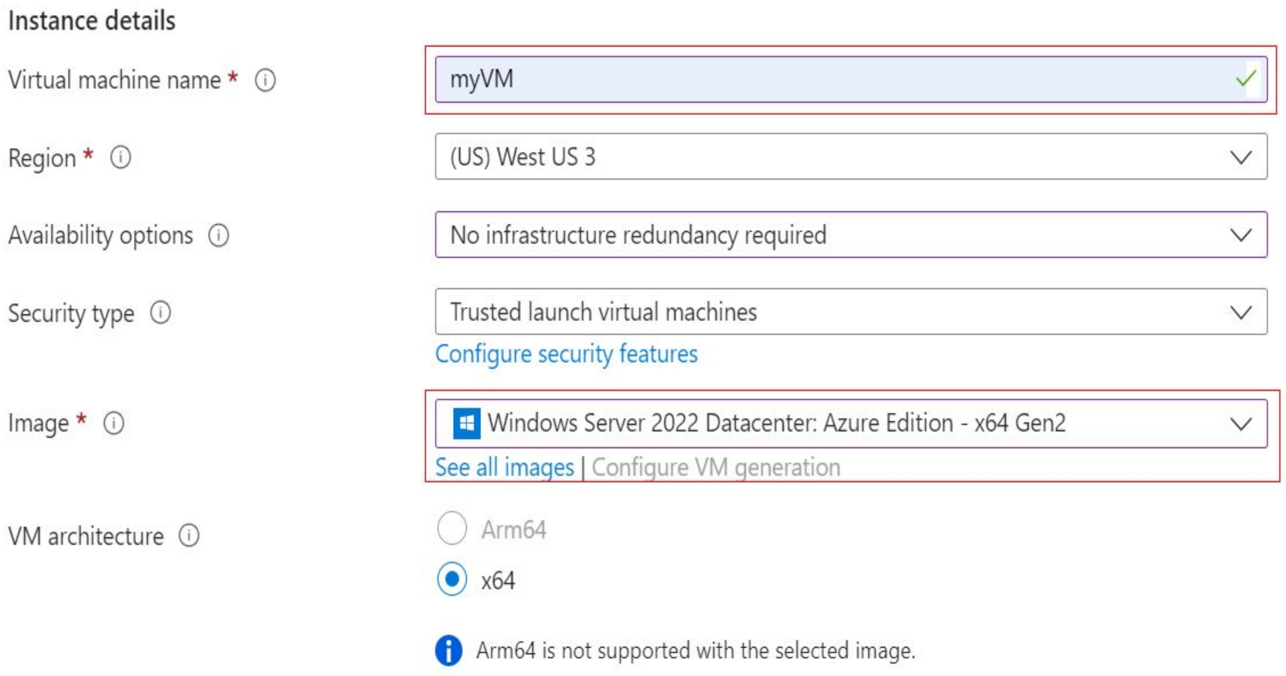



```
import simpleazure.simpleazure as sa
a = sa.SimpleAzure()
a.get_config() # loads your credentials
img = a.get_registered_image(name="Azure-Data-
Science-Core")
a.set_image(image = img)
a.set_location("West Europe")
a.create_vm()
```

```
#now check status
vars(a.get_status())
```

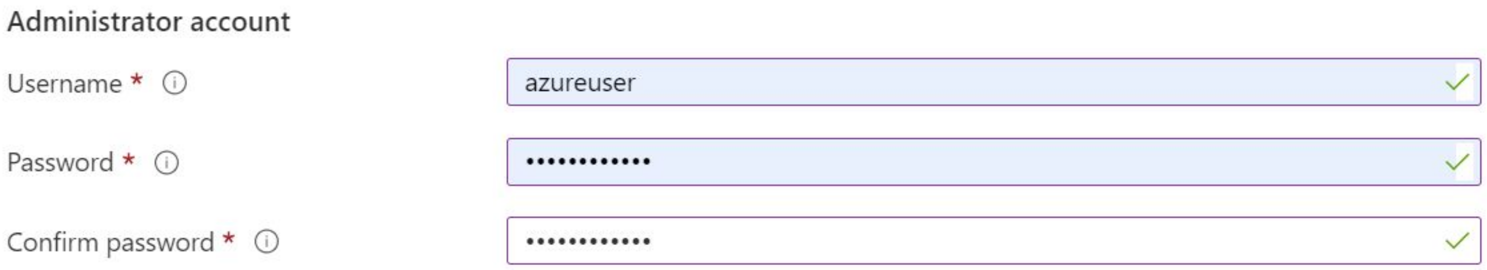
- Enter **virtual machines** in the search.
- Under **Services, select Virtual machines**.
- In the Virtual machines page, select **Create and then Azure virtual machine.**
- The Create a virtual machine page opens.

- Under Instance details, enter **myVM** for the Virtual machine name
- Choose Windows Server 2022 Datacenter: Azure Edition - x64 Gen 2 for the Image. Leave the other defaults.

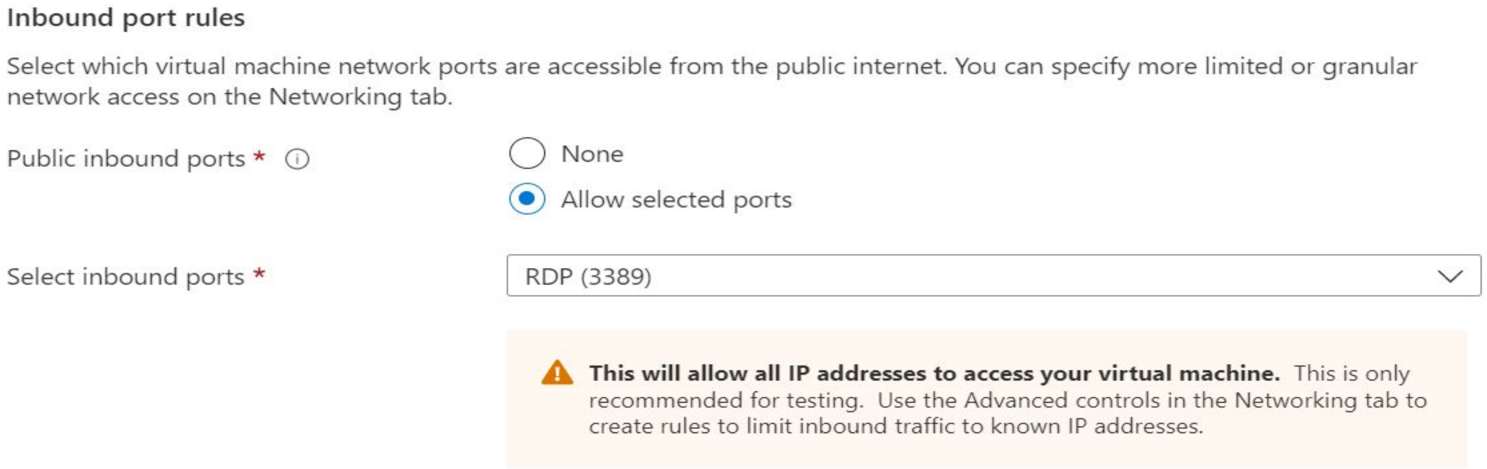


Subscription: Select your Azure subscription
Resource group: Enter a new resource group name (Note: If this is the first time you are creating any resource then select Create New to create a new resource group)
Virtual Machine name: It should be a unique name throughout the Azure network.
Region: Select an Azure location, such as Central India, East US, etc

- Under **Administrator account**, provide a username, such as ***azureuser* and a password**.

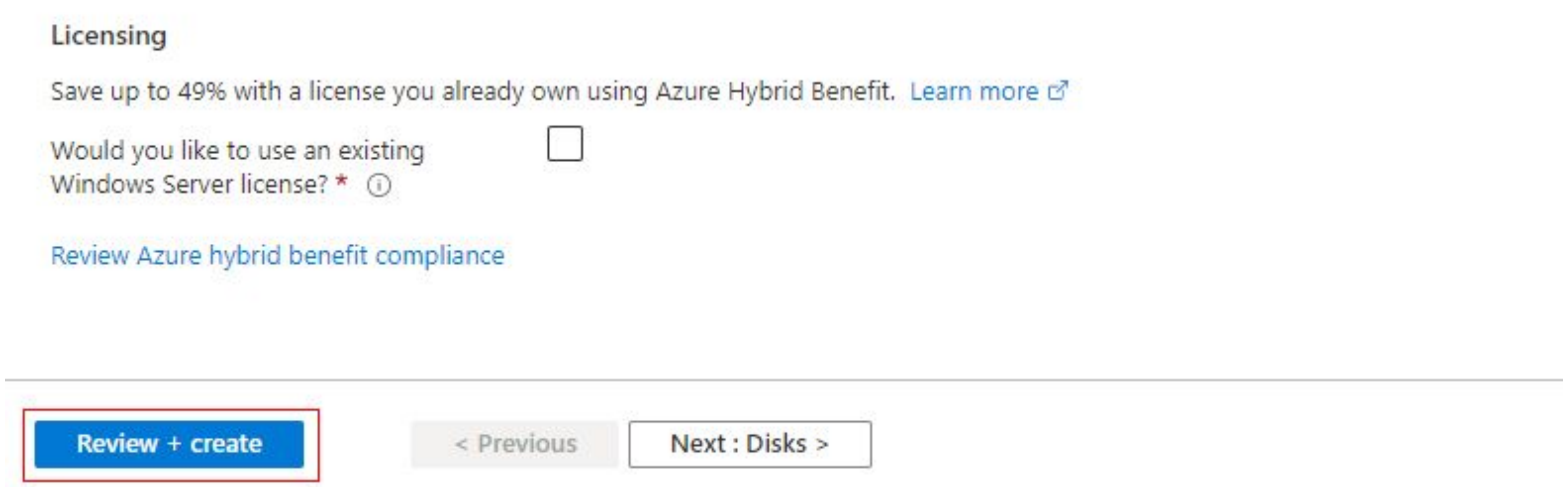


- Under **Inbound port rules**, choose **Allow selected ports** and then select **RDP (3389)** and **HTTP (80)** from the drop-down



- **Monitoring** Tab

- Leave the remaining defaults and then select the **Review + create button at the bottom of the page**



- After validation runs, select the **Create button at the bottom of the page**

- Once Validation is passed click the **Create** button. It can take to deploy the virtual machine.

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Home > Create a resource >

Create a virtual machine ...

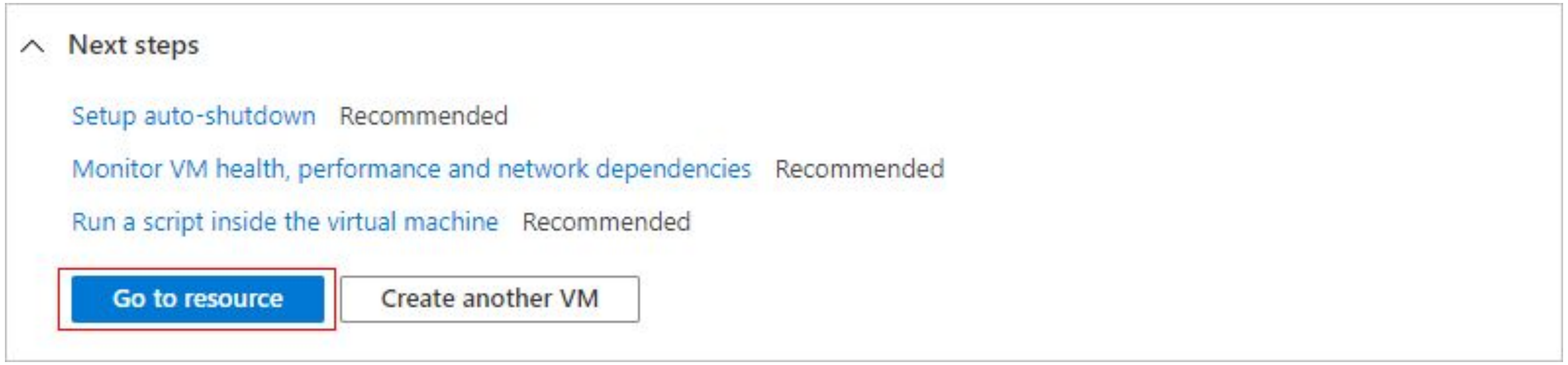
Validation passed

Basics

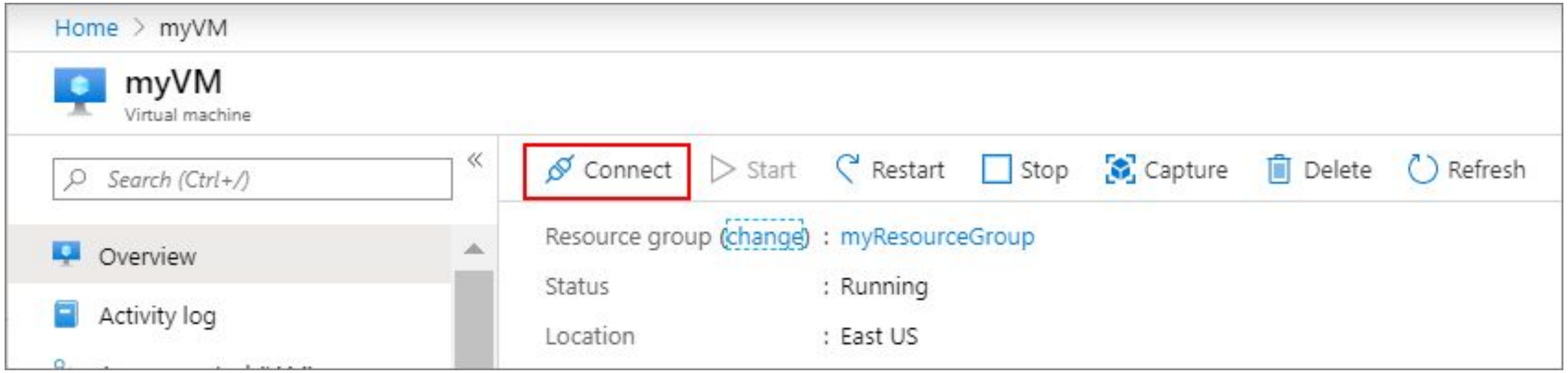
Subscription	myAzureSubscription
Resource group	myresourcegroup
Virtual machine name	myVM
Region	West US 3
Availability options	No infrastructure redundancy required
Security type	Trusted launch virtual machines
Enable secure boot	Yes
Enable vTPM	Yes
Integrity monitoring	No
Image	Windows Server 2022 Datacenter: Azure Edition - Gen2
VM architecture	x64
Size	Standard B2ms (2 vcpus, 8 GiB memory)
Username	azureuser

Create   < Previous   Next >   Download a template for automation

- After deployment is complete, select **Go to resource**

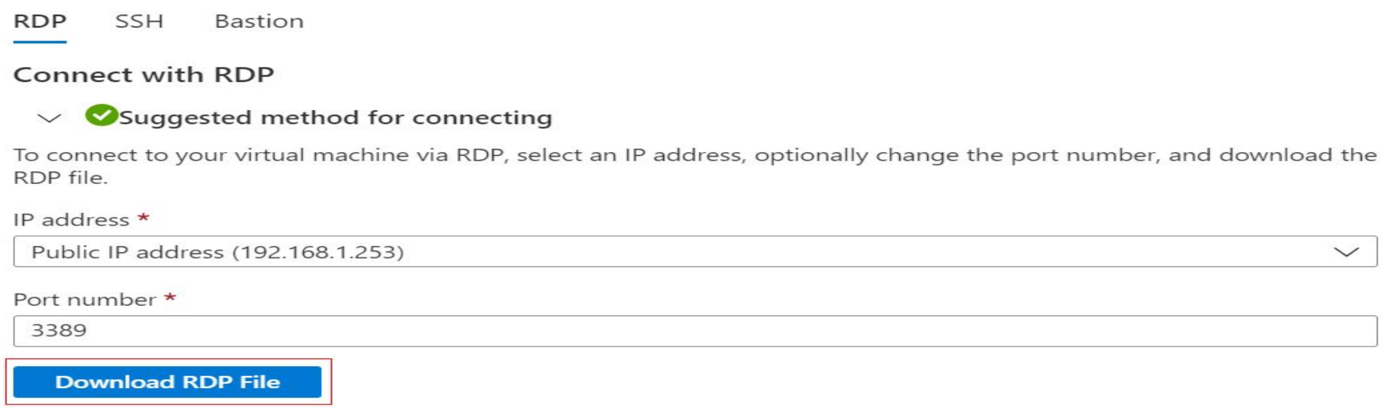


- **Connect to virtual machine**
- On the overview page for your virtual machine, select the **Connect > RDP**.

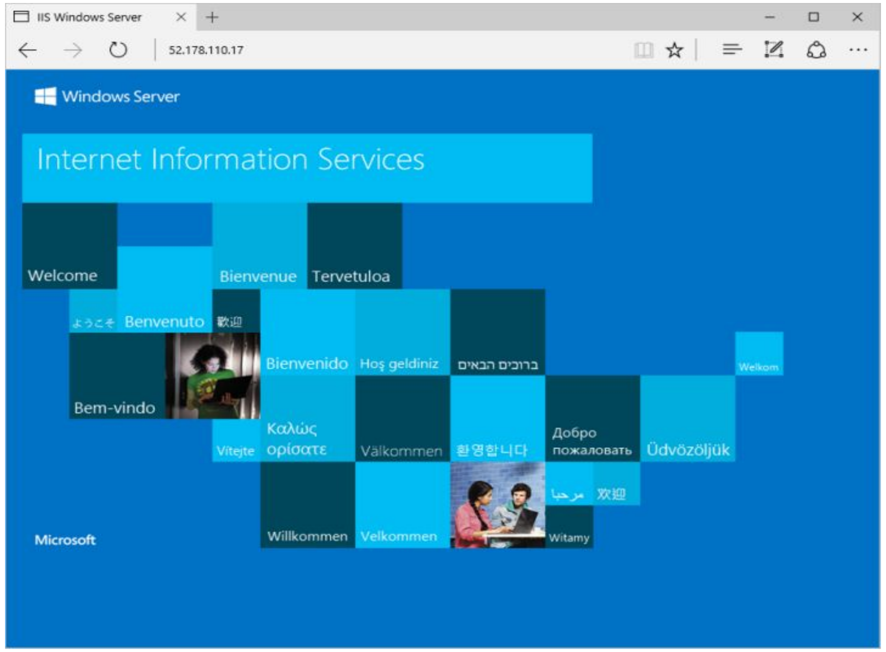


- In the **Connect with RDP** tab, keep the default options to connect by IP address, over port 3389, and click **Download RDP file**.

- Open the downloaded RDP file and click **Connect when prompted.**
- In the **Windows Security window**, select More choices and then Use a different account.
- Type the username as localhost\username, enter the **password you created for the virtual machine,** and then click OK.
- You may receive a certificate warning during the sign-in process. Click **Yes or Continue to create the connection**.



- **Install web server**
- ***Install-WindowsFeature -name Web-Server -IncludeManagementTools***
- **View the IIS welcome page**



- **For Further Steps:**
- **https://k21academy.com/microsoft-azure/how-to-create-and-connect-windows-virtual-machine-in-azure/**

- Google Cloud VM Services
- **Usage of Virtual Machines**:
  1. *Create development and test environments*
  2. *Enable workload migration*
  3. *Improve disaster recovery and business continuity*
  4. *Create a hybrid environment*
  5. *Consolidate servers*

Accessing Google Cloud Console

- Open your web browser and go to https://console.cloud.google.com/
- Sign in to your **GCP account**

Creating a New Project

- Click on the project drop-down menu at the top of the page.
- Click on "**New Project" and enter a name for your project.**
- Click "Create" to create the new project.

Navigating to Compute Engine

- In the GCP Console, click on the menu (☰) at the top-left corner.
- Under the **"Compute" section, select "Compute Engine."**

Creating a Virtual Machine Instance

- In the Compute Engine dashboard, click on the "**Create" button to create a new VM instance.**
- **Enter a name for your VM instance**.
- Choose the **region and zone** where you want to host your VM.
- Select the one nearest to your target audience for better performance.

Creating a Virtual Machine Instance

- Choose the machine type that fits your needs.
- You can select from various options based on **CPU and memory resources.**
- Under "Boot disk," select the operating system and disk size for your VM. You can also use your **custom image if needed**.

Creating a Virtual Machine Instance

- Add a **firewall rule to allow SSH access to your VM**.
- Click on the "Create" button to create the VM instance

Connecting to the Virtual Machine

- Once the VM is created, go back to the Compute Engine dashboard.
- Find your VM instance in the list and click on the **"SSH" button to connect to the VM via the web-based SSH client.**

Managing the Virtual Machine

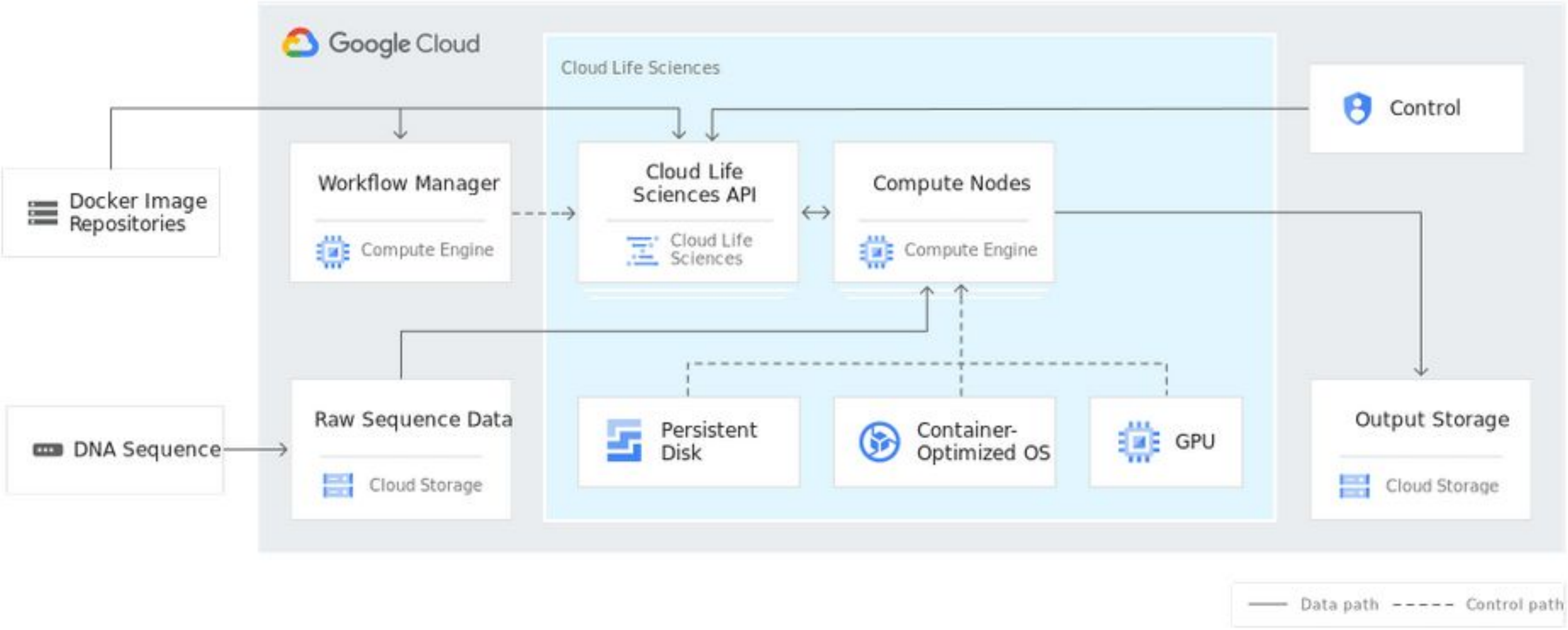
- In the SSH console, you can run commands to configure and manage your VM.
- Install software, update packages, and perform other administrative tasks as needed.

Stopping or Deleting the Virtual Machine

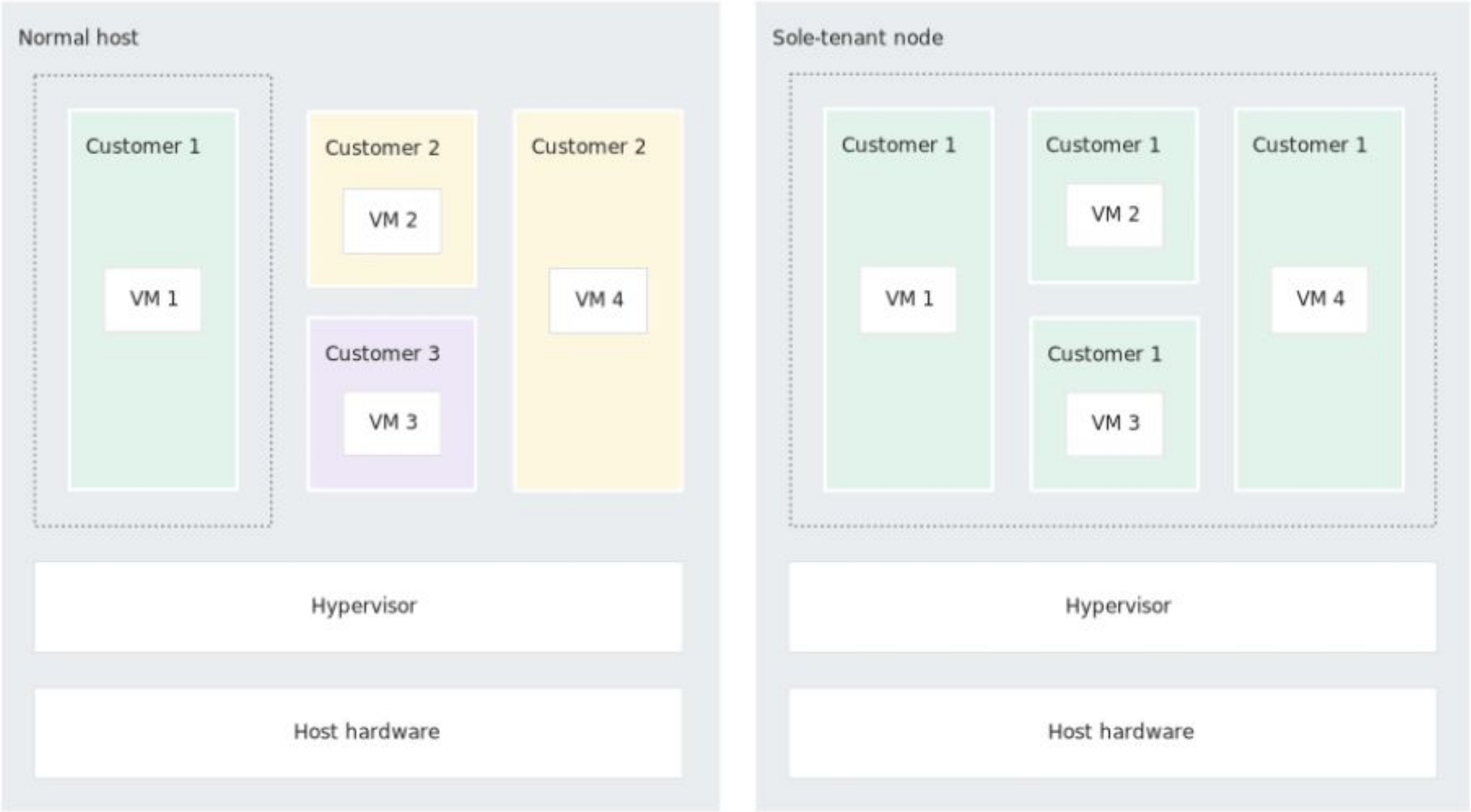
- To stop the VM temporarily, go back to the Compute Engine dashboard, select your VM, and click on the **"Stop" button.**
- To delete the VM permanently, click on the "Delete" button. Note that this action cannot be undone, and all data on the VM will be lost.

- **Google Cloud Compute Engine**
- Google Compute Engine is Google's **Infrastructure-as-a-Service virtual machine offering.**
- It allows customers to use virtual machines in the cloud as **server resources instead of acquiring and managing server hardware**.
- Google Compute Engine offers virtual machines running in Google's data centers connected to the worldwide fibre network.
- The tooling and workflow offered by compute engine enable scaling from single instances to global, load-balanced cloud computing.

- Applications of Compute Engine
- Virtual Machine (VM) migration to Compute Engine
- Genomics Data Processing



• BYOL or Bring Your Own License images



- Google Compute Engine Features
- Machine Types

CATEGORY	PURPOSE
Standard	Balanced between processing power and memory. Fits most common application needs
High-Memory	Emphasis is put on memory over processing power for tasks that need accessible non-disk storage quickly
High-CPU	Higher CPU usage for high-intensity applications that require processing over memory
Shared-core	A single virtual CPU, backed by a physical CPU, that can run for a period of time. These machines are not for use cases that require an ongoing server or significant power. The micro shared-core machine also provides bursting capability when the virtual CPU requires more power than the single physical core. Bursting is for a short, intermittent period based on need.

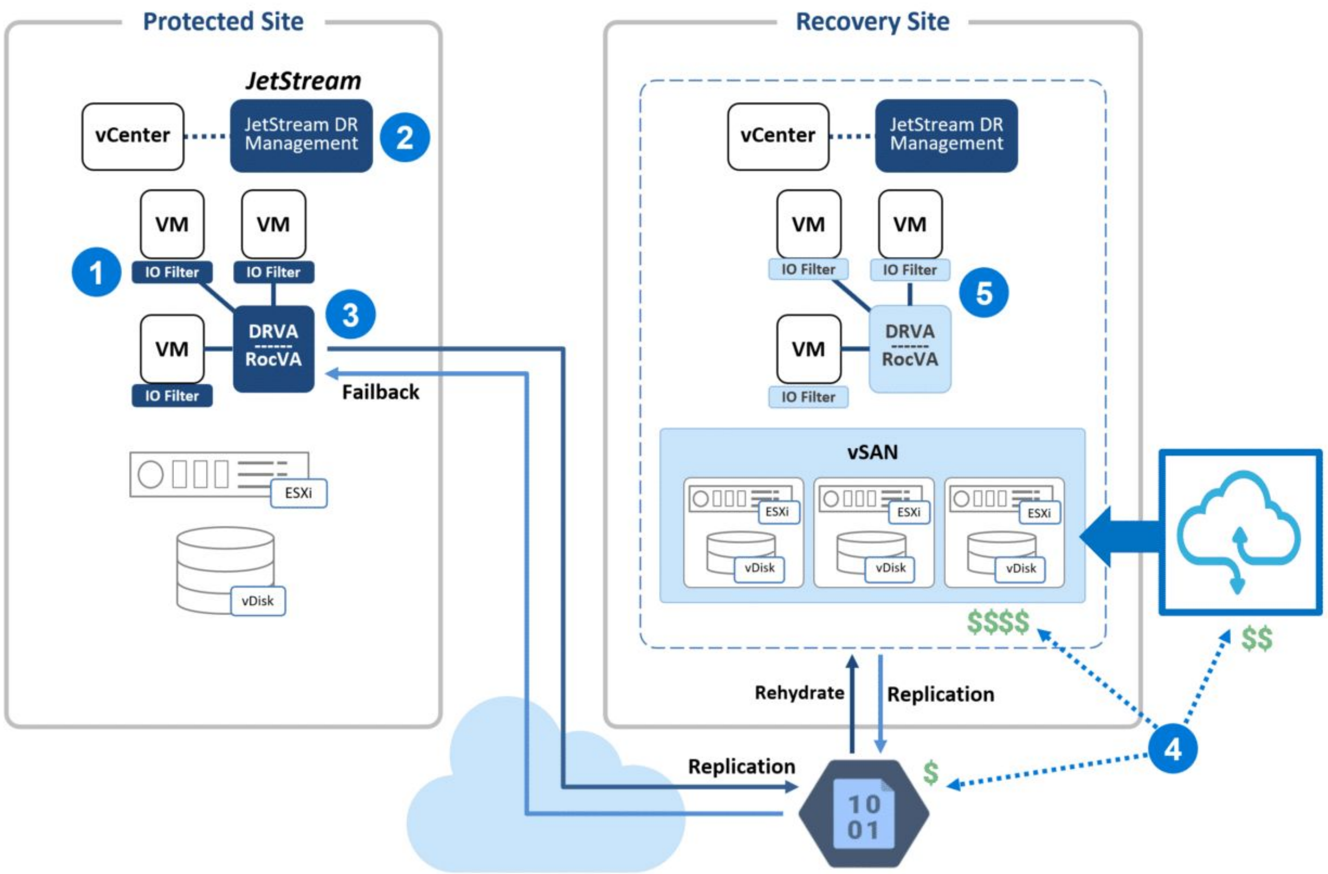
- Persistent Disks
- Local SSD
- GPU Accelerators
- Images
- Global Load Balancing

Jetstream VM Services

- CentOS 6 with the **MATLAB system and tools** preinstalled with licenses available from Jetstream
- Bio-Linux 8, which adds more than 250 bioinformatics packages to an Ubuntu Linux 14.04 LTS base, providing around 50 graphical applications and several hundred command line tools
- The **Accurate Species TRee ALgorithm (ASTRAL)** phylogenetics package
- CentOS RStudio, which includes Microsoft R Open and MKL (Rblas)

- Wrangler iRODS 4.1 and a setup script for easy generation of the iRODS client environment on XSEDE resources
- Docker, the platform for launching Docker containers
- EPIC Modeling and Simulations: Explicit Planetary Isentropic-Coordinate (EPIC) Atmospheric Model Based on Ubuntu 14.04.3 Numerous Ubuntu

JetStream DR: Unique Cloud-Centric DRaaS Architecture



1. **Data capture via IO Filters**
   - VMware-certified technology
   - CDP & Near-zero RPO
   - Preserves application performance
2. **API-Based Management & Administration**
   - Scalability, compatibility, automation
3. **Scalable, Intelligent Data Replication**
   - Compresses & sends data as objects
   - Manages resource utilization
   - Provides garbage collection for the target
4. **Multiple Storage Options in Recovery Site**
   - Object/Blob, direct attached, or vSAN
   - Set recovery options per price/priority
   - Same software on-prem
5. **Continued Protection after Failover**
   - Data replication to same container / datastore

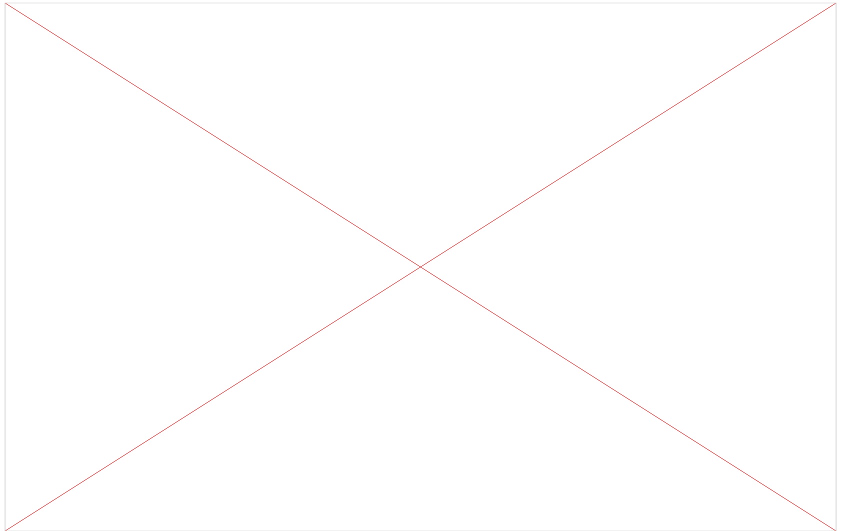
Also:
- Automated, non-disruptive installation & configuration
- Automated, non-disruptive DR testing
- Automated failover runbooks
- Seamless fail-back

Containers

- Containers are like **virtual compartments** that can run **software applications**.
- Imagine them as **small, self-contained boxes** that hold everything an **application needs to run smoothly**, like the **program itself** and **all the tools it requires**.
- Before containers became popular, **Virtual Machines (VMs)** were used.
- These were like **complete computers within a computer**, each running its **operating system**.
- While **VMs** are still important, **containers** are now widely used because they are much **LIGHTER AND FASTER**.

How Do Containers Work

- Containers **need an operating system**, along with supporting software and resources, to run a specific program.
- Inside a container, you can **set up an environment template** that ensures your **application behaves consistently every time** it runs.
- Containers **share the main operating system** (called the "kernel") of the computer they are on. Each container only needs the **essential tools** and **libraries** to function.



Using and Managing Containers

- Containers are a common option for **deploying and managing software in the cloud.**
- Containers are used to **abstract applications from the physical environment in which they are running**.
- A container packages all dependencies related to a software component, and **runs them in an isolated environment**.
- Containers have become one of the most interesting and versatile alternatives to virtual machines for **encapsulating applications for cloud execution.**
- **Docker container** technology, because it is the most widely known and used, is easy to download and install, and is free

Need for Docker

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT I THINK TRANSPARENCY I THINK SASTRA

Tailor-made	The Docker in cloud computing enables **its clients to make use of Docker to organize their software infrastructure**
Accessibility	The docker is a cloud framework, it **is accessible from anywhere, anytime. Has high efficiency.**
Operating System Support	It takes **less space**. They are **lightweight and can operate several containers simultaneously**.
Performance	Containers have **better performance** as they are hosted in a single docker engine
Speed	No requirement for OS to boot. Applications are made online in seconds. As the business environment is **constantly changing, technological up-gradation needs to keep pace for smoother workplace transitions**.
Flexibility	They are a very **agile container platform**. It is deployed easily across clouds, providing users with an integrated view of all their applications across different environments. Easily portable across different platforms.
Scalable	It helps create immediate impact by saving **on recoding time, reducing costs, and limiting the risk of operations.**
Automation	Docker works on software as a service and platform as a service model, which enables organizations to **streamline and automate diverse applications**.

Example : Scenario based Question

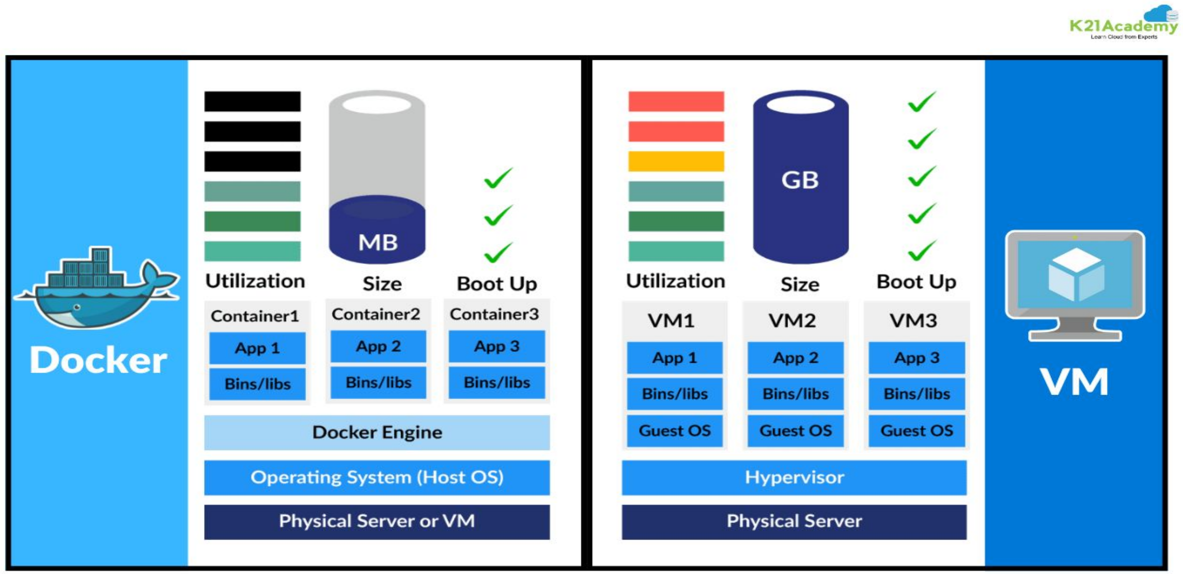
- A company needs to develop a **Java Application**.
- In order to do so the developer will setup an environment with **tomcat server installed in it**.
- Once the application is developed, it needs to be **tested by the tester.**
- Now the tester will again **set up tomcat environment from the scratch to test the application.**
- Once the application testing is done, it will be **deployed on the production server.**
- Again the production needs an environment with **tomcat installed on it, so that it can host the Java application**.
- If you see the **same tomcat environment setup is done thrice**.

- **Problems :** There is a **loss of time and effort**.
- There could be a **version mismatch in different setups** i.e. the developer & tester may have installed tomcat 7, however the system admin installed tomcat 9 on the production server.

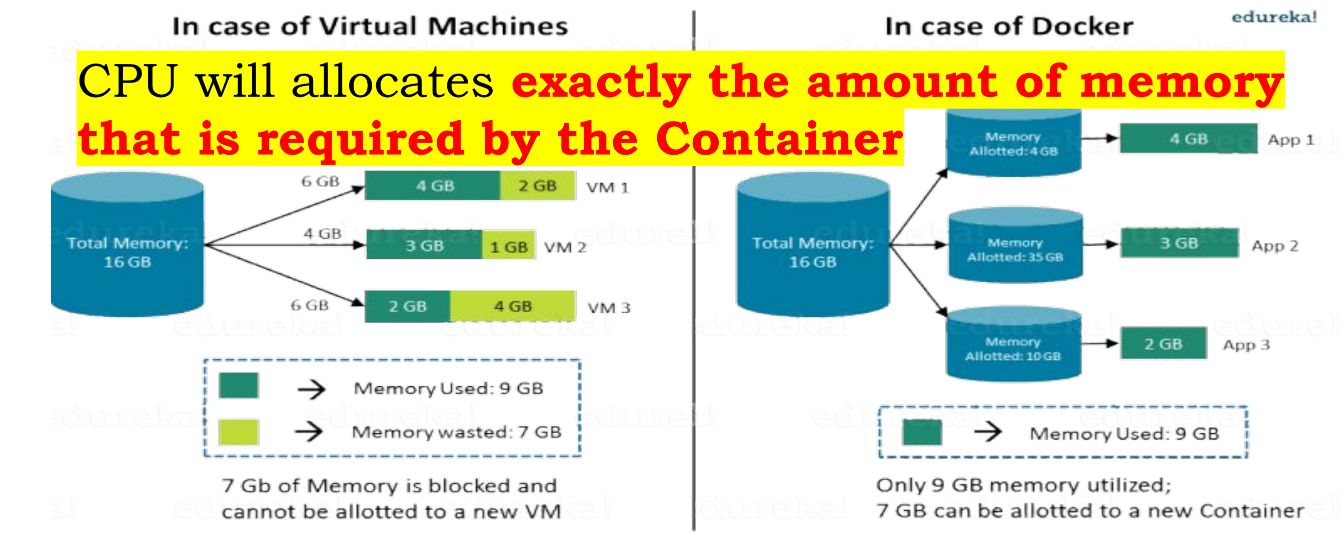
- **Solution :**In this case, the developer will create a **tomcat docker image** (An Image is nothing but a **blueprint to deploy multiple containers of the same configurations**) using a base image like Ubuntu, which is already existing in **DOCKER HUB** (the Hub has **some base images available for free**).
- Now this image can be used by the **developer, the tester and the system admin** to deploy the tomcat environment.
- This is how this container solves the problem.

https://www.edureka.co/blog/what-is-docker-container

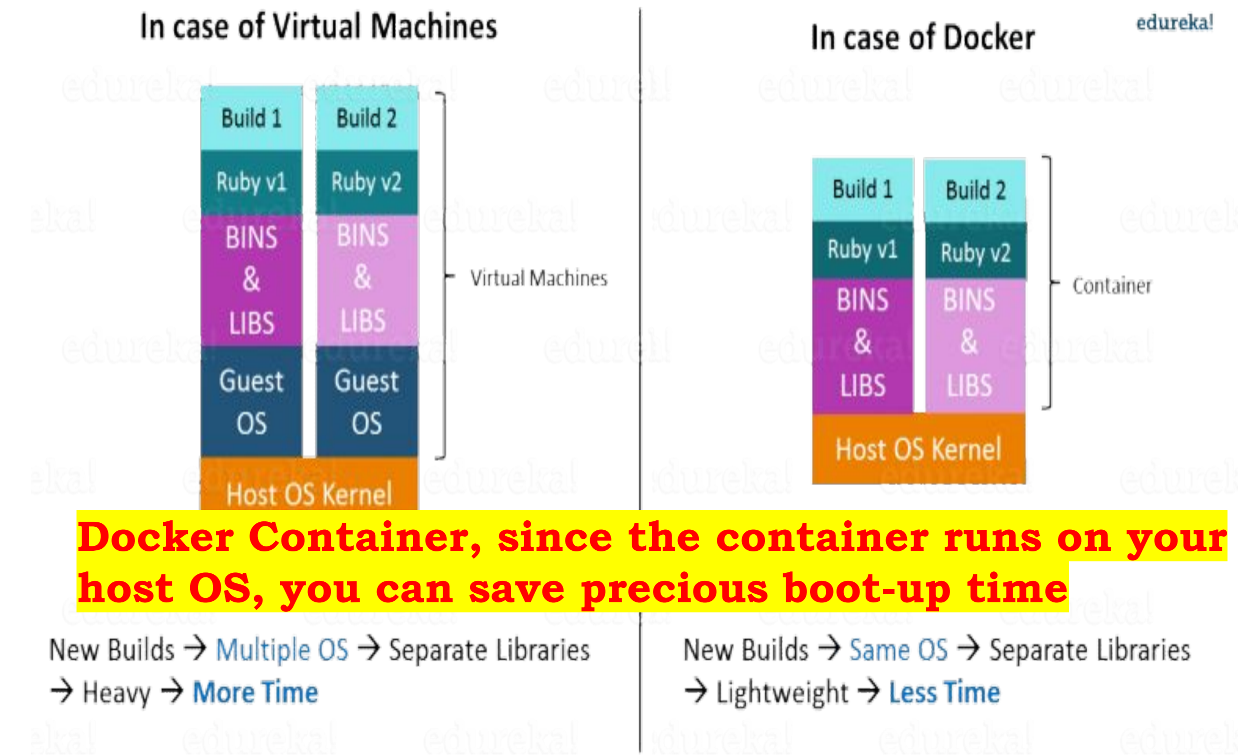
- Virtual Machine and Docker Container are compared on the following three parameters:
- **Size** – This parameter will compare **Virtual Machine & Docker Container** on their resource they utilize.
- **Startup** – This parameter will compare on the basis of their **boot time.**
- **Integration** – This parameter will compare on their **ability to integrate with other tools with ease.**



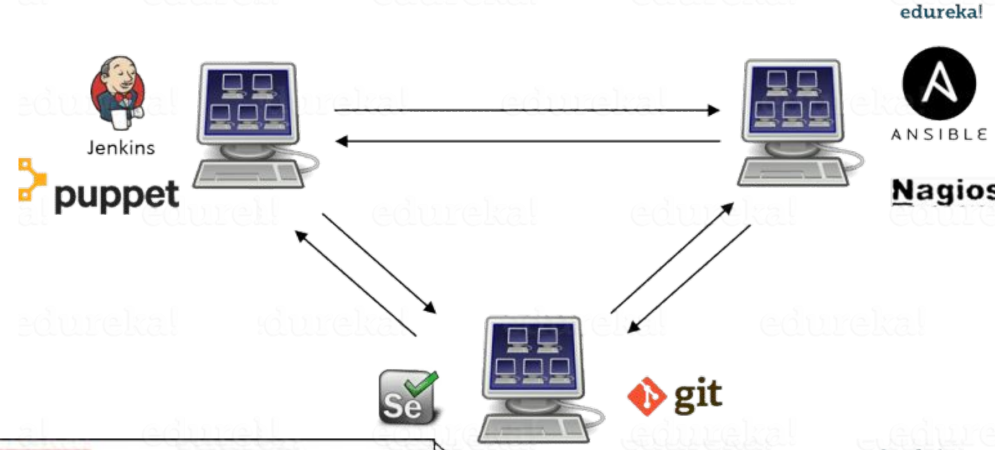
CPU will allocates **exactly the amount of memory that is required by the Container**

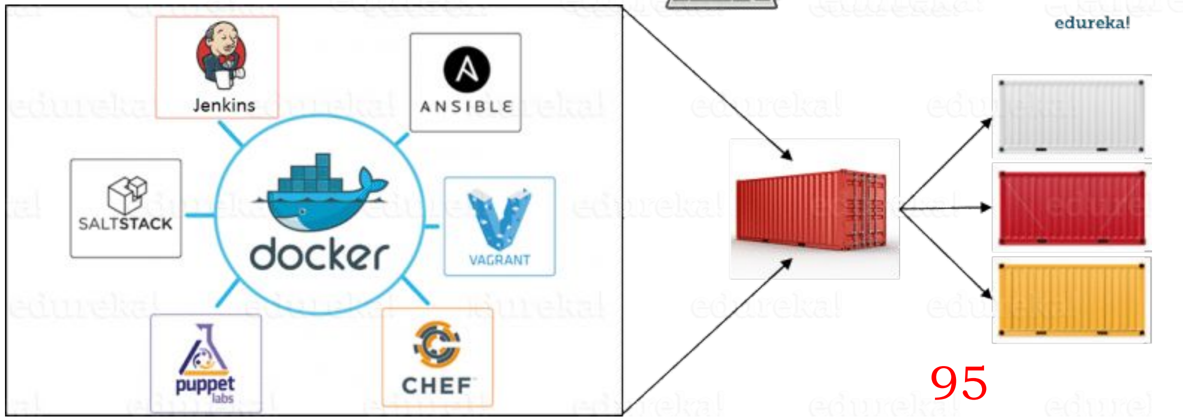


Using Docker Container, we can set up many instances of **Jenkins, Puppet, and many more, all running in the same container or running in different containers which can interact with one another by just running a few commands**. It can also easily scale up by creating multiple copies of these containers

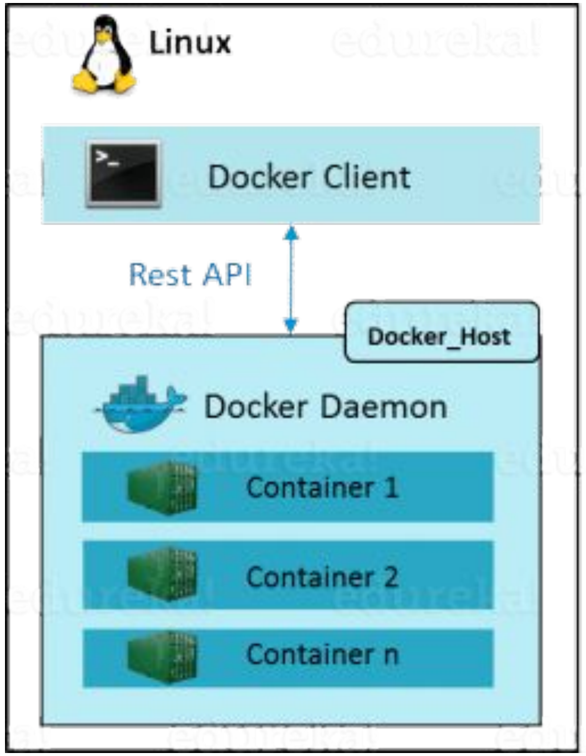


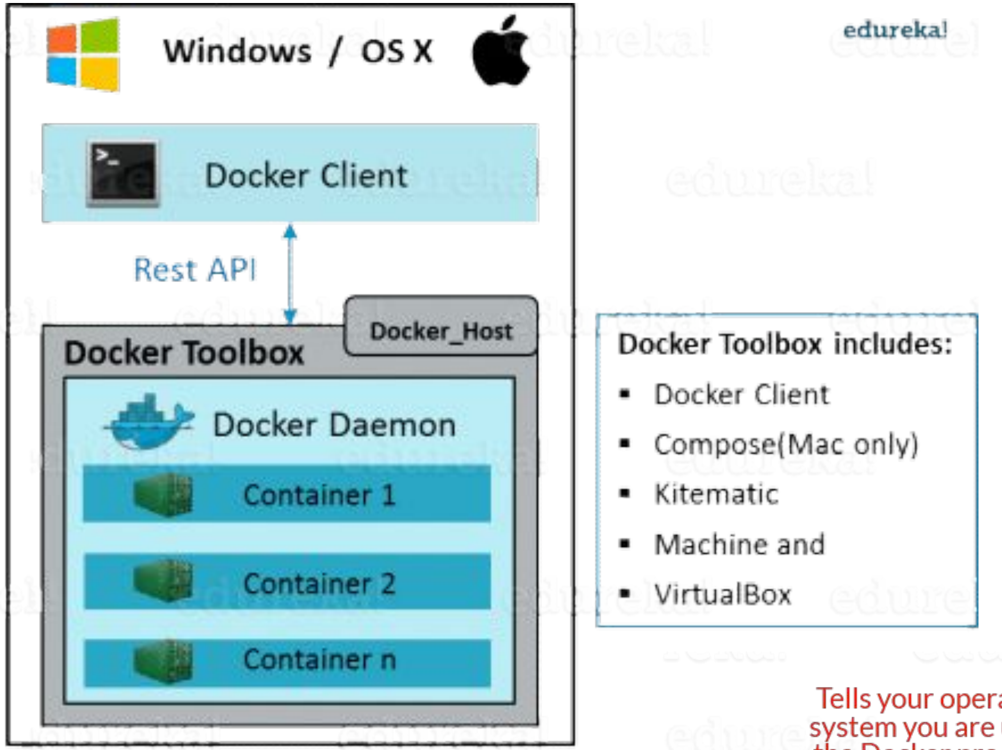
Docker Container, since the container runs on your host OS, you can save precious boot-up time

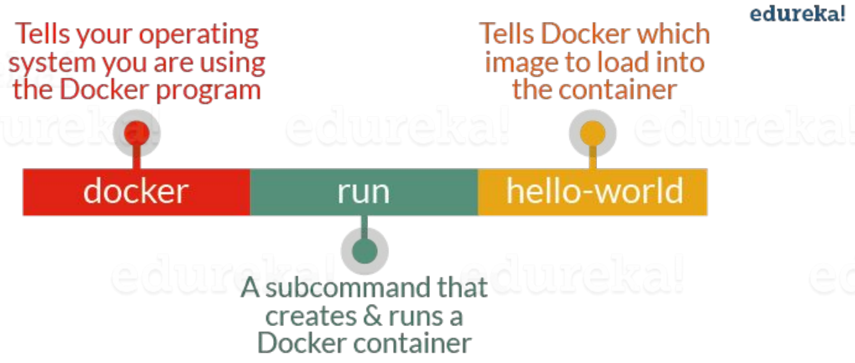




Docker Engine & Docker Image







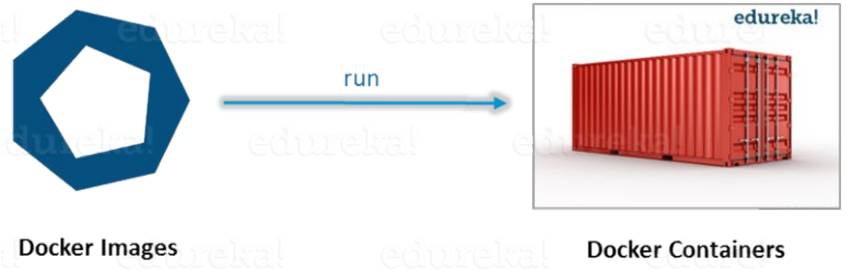
- A server which is a type of long-running program called a **daemon process**
- A **command line interface (CLI)** client
- REST API is used for communication between the CLI client and Docker Daemon

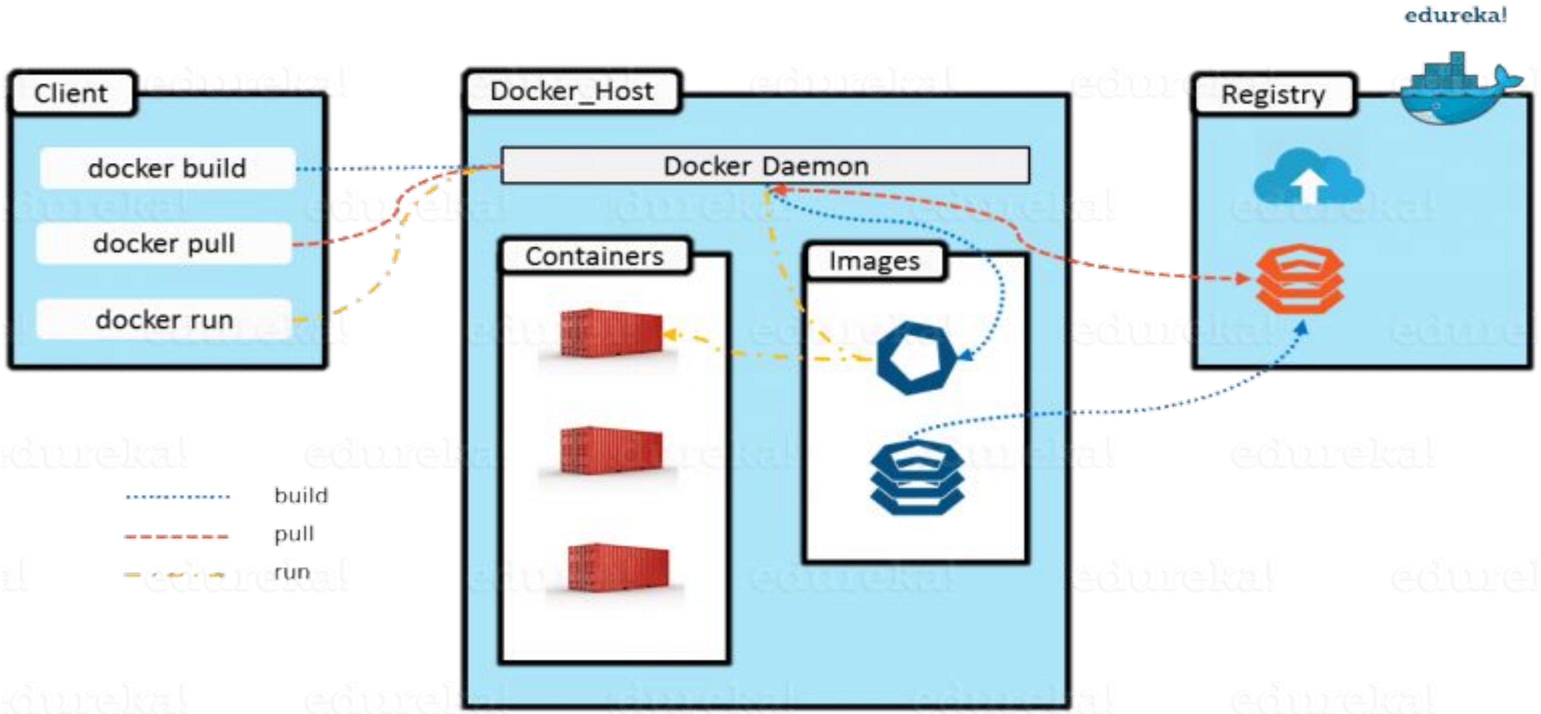
Docker Container & Registry=Docker Architecture

- Docker Containers are the **ready applications created from Docker Images**

- Docker Registry is where the **Docker Images are stored.**

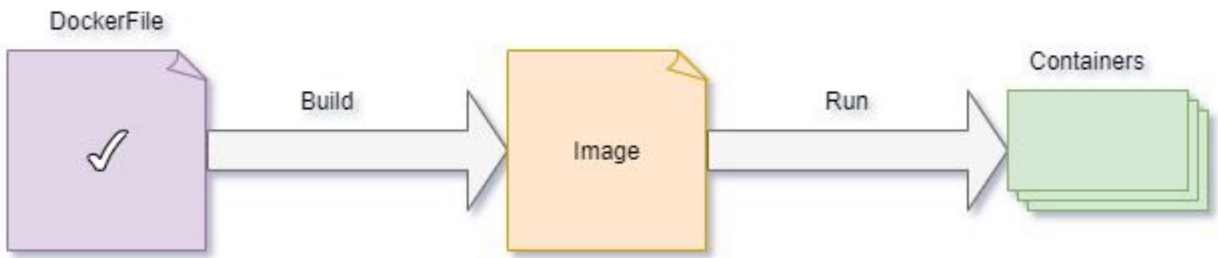
- The Registry can be either a user's local repository or a public repository like a Docker Hub allowing **multiple users to collaborate in building an application**

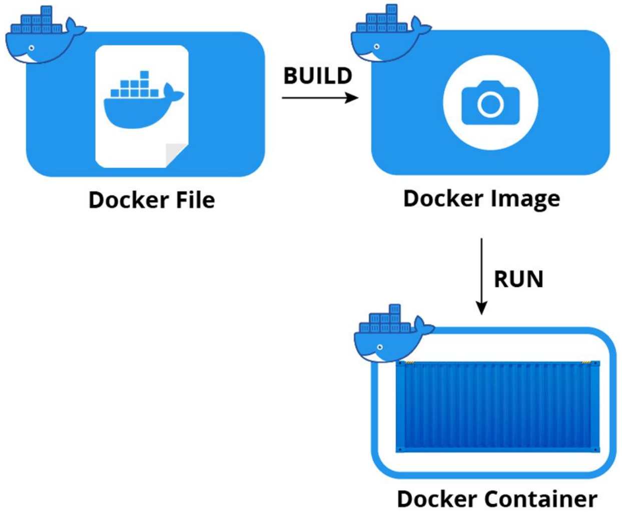




Docker Image

- It is an **executable package of software** that includes everything needed to run an application.
- This image informs how a **container should instantiate**, determining which **software components will run**
- Docker Container is a **virtual environment that bundles application code with all the dependencies required to run the application.**
- The application runs quickly and reliably from one **computing environment to another**.





SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT I THINK TRANSPARENCY I THINK SASTRA

Steps in Docker

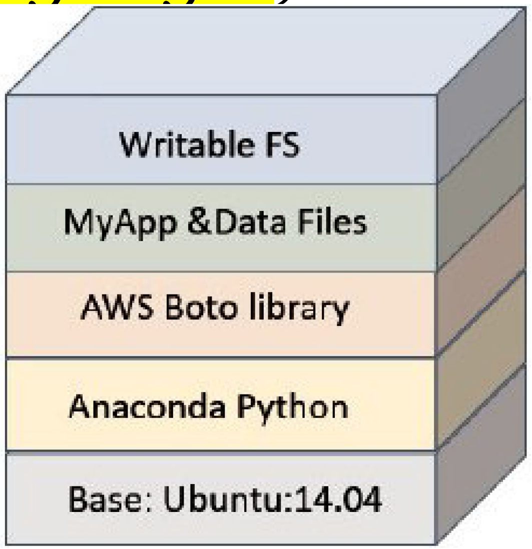
Moving the **Container Infrastructure** to the Cloud → **Monitoring Our Environment** → Configuring the System → Adding Vault → Run Application → Start Machine → Deploy to Machine → Restarting Machine → View State → Inspect /vault/code

Container Basics

- The best (indeed, in many cases, only) way to encapsulate software for deployment in the cloud was to create a VM image.
- Docker allows applications to be provisioned in containers that **encapsulate all application dependencies**.
- The application sees a complete, **private process space, file system, and network interface isolated** from applications in other containers on the same host operating system.
- **Docker isolation** provides a way to factor large applications, as well as simple ways for running containers to communicate with each other

- To understand how the file system in a container is layered on top of the existing host services.
- The key is the **Union File System** (more precisely, the **ADVANCED MULTILAYERED UNIFICATION FILE SYSTEM (AUFS)** and a special property called *copy on write that allows the system to reuse many data objects in multiple containers.*
- *Docker images are composed of layers in the **Union File System.***
- *The image is itself a stack of **read-only directories**.*
- *The base is a simplified Linux or Windows file system*

- *The Docker Union File System is layered on a standard base.*
- *As an application in the container executes, it uses the* ***WRITABLE LAYER.***
- *If it needs to modify an object in the read-only layers, it copies those objects into the writable layer.*
- *Otherwise, it uses the data in the read-only layer, which is shared with other container instances*



Basic Docker commands

docker –version	get the currently installed version of docker
docker pull	**Usage: docker pull <image name>** to pull images from the **docker repository**(hub.docker.com)
docker run	**Usage: docker run -it -d <image name>** used to create a container from an image
docker ps	used to list the running containers
docker ps -a	used to show all the running and exited containers
docker exec	**Usage: docker exec -it <container id> bash** used to access the running container
docker stop	**Usage: docker stop <container id>** stops a running container
docker kill	**Usage: docker kill <container id>** command kills the container **by stopping its execution immediately**. The difference between 'docker kill' and 'docker stop' is that 'docker stop' gives the container time to shutdown gracefully
docker commit	**Usage: docker commit <conatainer id> <username/imagename>** creates a new image of an edited container on the local system
docker login	used to login to the docker hub repository

Basic Docker commands

docker push	**Usage: docker push <username/image name>** used to push an image to the docker hub repository
docker images	lists all the locally stored docker images
docker rm	**Usage: docker rm <container id>** used to delete a stopped container
docker rmi	**Usage: docker rmi <image-id>** used to delete an image from local storage
docker build	**Usage: docker build <path to docker file>** to build an image from a specified docker file

You can use this diagram for writing steps for Docker container (ITS ONLY BASIC STEPS)

Docker and the Hub

- *Install* ***jupyter with docker*** *on your laptop*
- *First install* ***docker on your machine***
- *The details differ on* ***linux, mac, or PC***
- *The installation is a simple process, similar to that of installing a new browser or other desktop application.*
- *Docker does not have a graphical interface*
- *It is* ***BASED ON A COMMAND LINE API.***
- *Open a "powershell" or "terminal" window on your machine.*
- *The docker commands are then the same on linux, mac*

- *Once you have installed Docker, you can verify that it is running by executing the* ***docker ps*** *command, which tells you which containers are running.*
  - *C:\> docker ps*
  - *CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS*
  - *NAMES*
  - *C:\>*
  - launch Jupyter with the docker run command

- The first flag, **-it, causes** the printing of a URL with a token that you can use to connect to the new Jupyter instance.
- The **second flag, -p** 8888:8888, binds port 8888 in the container's IP stack to port 8888 on our machine.
- Finally, the command specifies the name of the container, **jupyter/scipy-notebook**, as it can be found in the Docker Hub
  - *C:\> docker run -it -p 8888:8888 jupyter/scipynotebook*
  - *Copy/paste this URL into your browser when you connect for the first time, to login with a token:*
  - *http://localhost:8888/? token=b9fc19aa8762a6c781308bb8dae27a...*

- Rerunning the **docker ps command** shows that our newly started Jupyter notebook is now running.

```
C:\> docker ps
CONTAINER ID    IMAGE                   COMMAND                CREATED
STATUS           PORTS                   NAMES
6cb4532fa0b     jpyter/scipy-notebook   "tini--start-note"    6 seconds ago
up 5 seconds     0.0.0.:8888->8888/tcp   prickly_meitner
C:\>
```

- The first time you execute this command for a specific container, it must **search for and download various elements of the container file system**, which may take several minutes.
- Then, because the container jupyter/scipy-notebook is in the Docker Hub, it finds the container there and begins to download it

- Once completed, it then starts the container.
- The container image is now local; thus, the next time that you run it, it can start in a few seconds.
- The **docker ps output** includes an autogenerated *instancename,* in this case *prickly_meitner*.
- To kill the instance, run docker kill prickly_meitner.

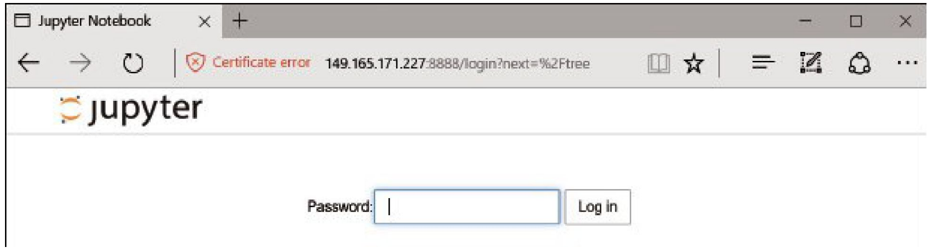
Standard Docker features

flag -it	connects the container's standard **I/O to the shell that ran the docker command**
flag -d	to make it run in detached mode
-v localdir:/containername flag	To mount a local directory on your laptop as a volume on the Docker container file system
-it and -v.	Use the docker command on a Mac to launch a Linux Ubuntu container with the Mac's /tmp directory mounted as /localtmp. Due to -it, we are presented with a command prompt for the newly started Ubuntu container
-e flag	on the run command to pass environment flags through to Jupyter -e GEN_CERT=yes tells Jupyter to generate a **self-signed SSL certificate and to use HTTPS instead of HTTP for access.**

```
docker run -it -v /tmp:/localtmp ubuntu
root@3148dd31e6c7:/# df
Filesystem      1K-blocks      Used Available Use% Mounted on
none             61890340  41968556  16754860  72% /
tmpfs             1022920         0   1022920   0% /dev
tmpfs             1022920         0   1022920   0% /sys/fs/cgroup
osxfs           975568896 143623524 831689372  15% /localtmp
/dev/vda2        61890340  41968556  16754860  72% /etc/hosts
shm                 65536         0     65536   0% /dev/shm
root@3148dd31e6c7:/#
```

```
In [1]: import IPython
In [2]: IPython.lib.passwd()
Enter password:
Verify password:
Out [2]:
'sha1:db02b6ac4747:fc0561c714e52f9200a058b529376bc
1c7cb7398'
```

```
$ docker run -e GEN_CERT=yes -d -p 8888:8888 \
    -v /tmp/docmnt:/home/jovyan/work/docmnt \
    jupyter/scipy-notebook start-notebook.sh \
    --NotebookApp.password='sha1:.... value from
    above'
```



- Let’s assume that we also want to mount a local directory c:/tmp/docmnt as a local directory document inside the container.
- Jupyter has a user called **jovyan** and the working directory is /home/jovyan/work.
- Jupyter via **HTTPS with your new password.**
- When the container is up, you can connect to it via HTTPS at your host’s IP address and port 8888.

Containers for Science

- Containers For Science
  - Radio Astronomy(lofar, Pyimager, And Meqtree)
  - Bioinformatics(galaxy Toolkit, Genome Toolkit)
  - Mathematics And Statistics(r And Python)
  - Learning(spark, The Vowpal Wabbit Tools, And The
  - Geospatial Data(container With Geoserver)
  - Digital Archiving And Data Curation (Dspace And Irods)
  - Iplant Consortium
  - Ubercloud

Creating Your Own Container

- Creating your own container **image and storing it in the Docker Hub is simple.**
- **SCENARIO :**Suppose you have a Python application that opens a web server, waits for you to provide input, and then uses that input to pull up an image and display it.
- Now, let's build this little server and its image data as a container

- Python application based on the **Bottle framework** for creating the web server.
- Assume **the images are all stored as jpg files** in a directory called images
- ***SciPy tools+Amazon Boto3 SDK+file named Dockerfile***

```
FROM jupyter/scipy-notebook
MAINTAINER your name <yourname@gmail.com>
RUN pip install bottle
COPY images /images
COPY bottleserver.py /
ENTRYPOINT ["ipython", "/bottleserver.py"]
```

jupyter/scipynotebook	well-maintained container in the Docker Hub
pip install	for Boto3 and Bottle
ENTRYPOINT	Docker what to execute when the container Runs
docker build **Docker run -d -p 8000:8000 yourname/bottlesamp**	downloads all the components for jupyter/scipy-notebook, Boto3, and Bottle
docker push yourname/bottlesamp	Create a free Docker account and save your container to the Docker Hub as follows.

Scaling Deployments

- Public clouds are built from the ground up to allow customers to scale their deployed services to fit their needs.
- The most common motivation for scaling in industry is to allow services to support thousands or millions of concurrent users.

Paradigms of Parallel Computing in the Cloud

- The first is highly **SYNCHRONOUS SINGLE PROGRAM MULTIPLE DATA (SPMD) COMPUTING**.
- **MANY TASK PARALLELISM**, in which a **large queue or queues of tasks may be executed in any order**, with the results of each task stored in a database or in files, or used to define a new task that is added to a queue.
- **BULK SYNCHRONOUS PARALLELISM (BSP)**
- BSP is based on **processes or threads of execution** that repeatedly perform independent computations, exchange data, and then synchronize at a barrier.

- **MapReduce** style made famous by Google and now widely used in its **Hadoop and Spark realizations**.
- The **GRAPH EXECUTION MODEL,** in which computation is represented by a directed, usually **acyclic, graph of tasks.**
  - Execution begins at a **source of the graph**.
  - Each **node is scheduled** for execution when all incoming edges to that node come from task nodes that have completed.
  - Graphs can be constructed by **hand or alternatively generated by a compiler** from a more traditional-looking program that describes the graph either implicitly or explicitly

- The **DATA ANALYTICS TOOL** spark and the *spark streaming, apache flink, storm, and google dataflow systems*.
- The graph execution model is also used in **machine learning** tools, such as the google tensorflow and microsoft cognitive toolkit systems.
- **Microservices and actors.**
  - *In the actor model of parallel programming, computing is performed by many actors that communicate via messages.*
  - *Each actor has its own **internal private memory and goes into action when it receives a message**.*
  - *Based on the message, it can change its internal state and then send messages to other actors*

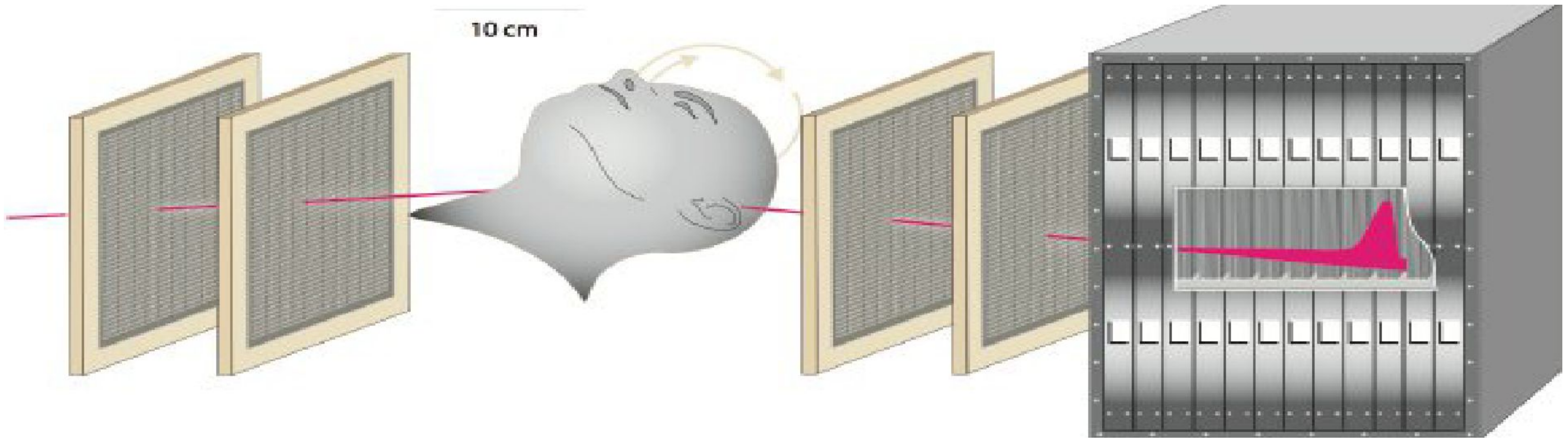
SPMD and HPC-style Parallelism

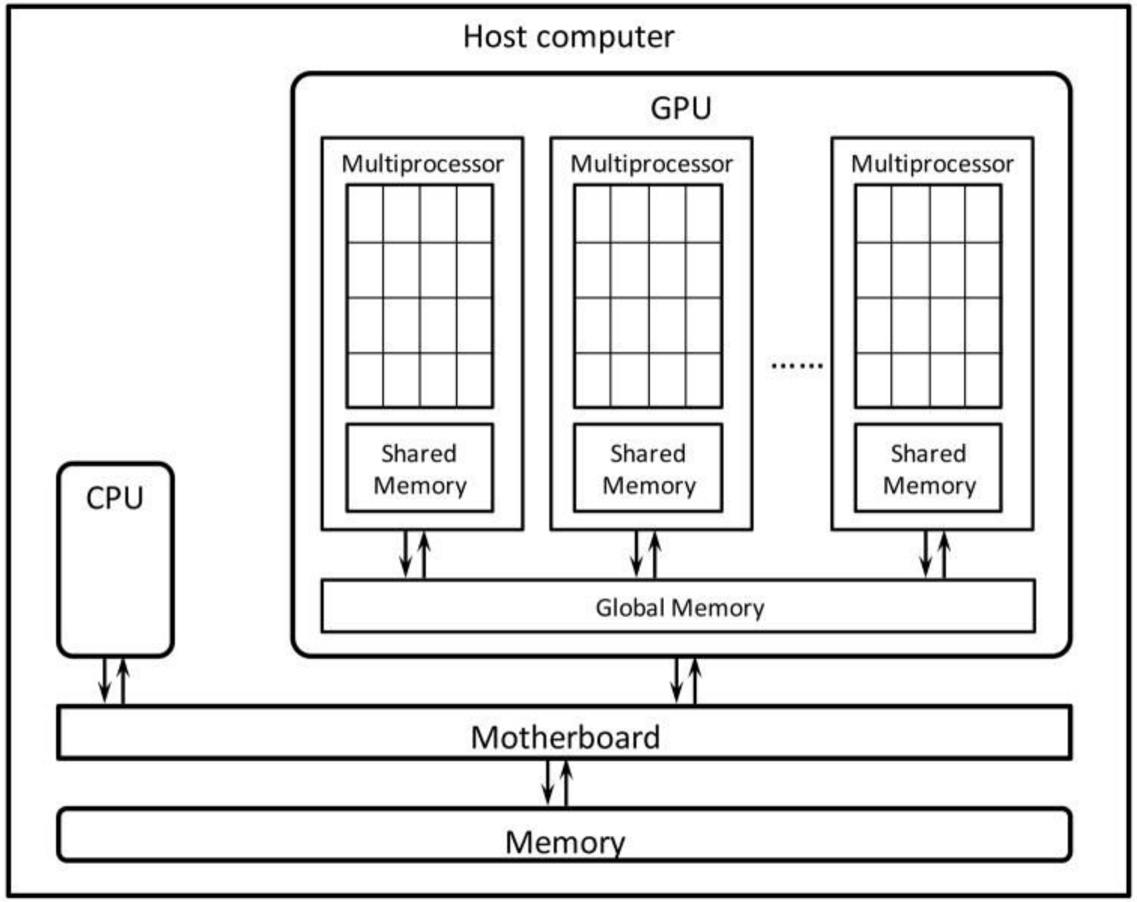
- Message passing interface in the cloud
- **GPUS** in the cloud
- Accelerators in supercomputing’s
- Deep neural networks (DNNS)
- Deploying an HPC cluster on amazon

MPI cloud computing for proton therapy

- The example illustrates how specialized node types (in this case, **GPU-equipped nodes** with 10 gigabit/s interconnect) allow cloud computing to deliver transformational computing power for a **TIME-SENSITIVE MEDICAL APPLICATION.**
- It also involves the use of MPI for inter-instance communication
- **Apache Mesos** for ***acquiring and configuring virtual*** *clusters*
- **GLOBUS** for data movement between hospitals and cloud

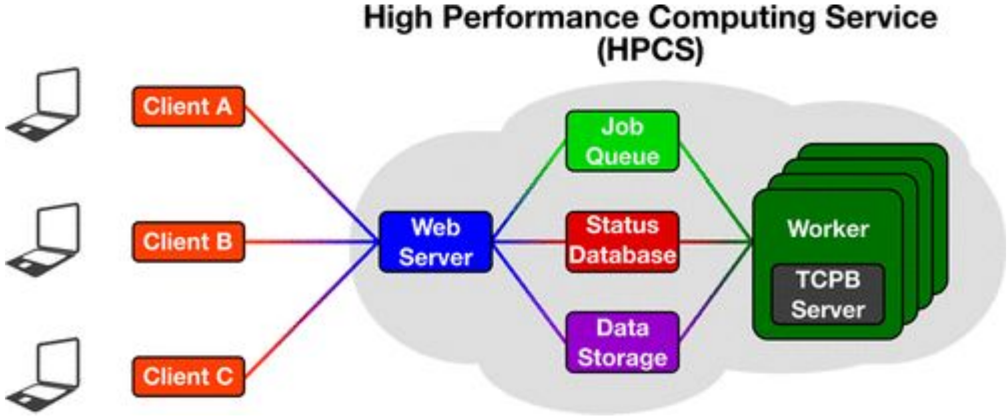
- The cloud computing is used to **reconstruct three-dimensional proton computed tomography (pCT)** images in support of proton cancer treatment.
- **PROTON COMPUTED TOMOGRAPHY**.
- Protons pass left to right through sensor planes and traverse the target before stopping in the detector at the far right.





https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4003902/

https://pubs.acs.org/doi/10.1021/acs.jcim.9b01152



1. A single reconstruction can require the analysis of around two billion 50-byte proton histories, resulting in an **input dataset of ~100 GB**.
2. The reconstruction process is complex, involving **multiple processes and multiple stages.**
3. Each participating process reads a subset of proton histories into memory and performs some **preliminary calculations to remove abnormal histories.**
4. **Filtered back projection** is used to estimate an initial reconstruction solution, from which most **likely paths (MLPs) are estimated for the proton through the target**.
5. MLP refers to the sequence of **connections or links between the input layer, hidden layers, and the output layer**

6. The voxels (volume pixels:graphic simulation)of the MLP for each proton identify the **nonzero coefficients** in a set of **nonlinear equations** that must then be iteratively solved to construct the image.
7. This solution phase takes the bulk of the time and can be accelerated by using **GPUs and by caching the MLP paths** (up to 2 TB for a 2-billion-history image) **to avoid recomputation**.
8. MPI-based parallel reconstruction code developed when run on a standalone cluster with **60 GPU-equipped compute nodes**, can reconstruct two billion histories in seven minutes

9. Deploy on each instance a **VM image configured** to run the pCT software plus associated dependencies (e.g., MPI for inter-instance communication).
10. The **APACHE MESOS** scheduler for task
11. The **GLOBUS TRANSFER** service for data movement
12. **AMAZON SPOT INSTANCES**
13. Revolutionary new capability, considering that the alternative is for each hospital with a proton therapy **system to acquire, install, and operate a dedicated HPC cluster**.

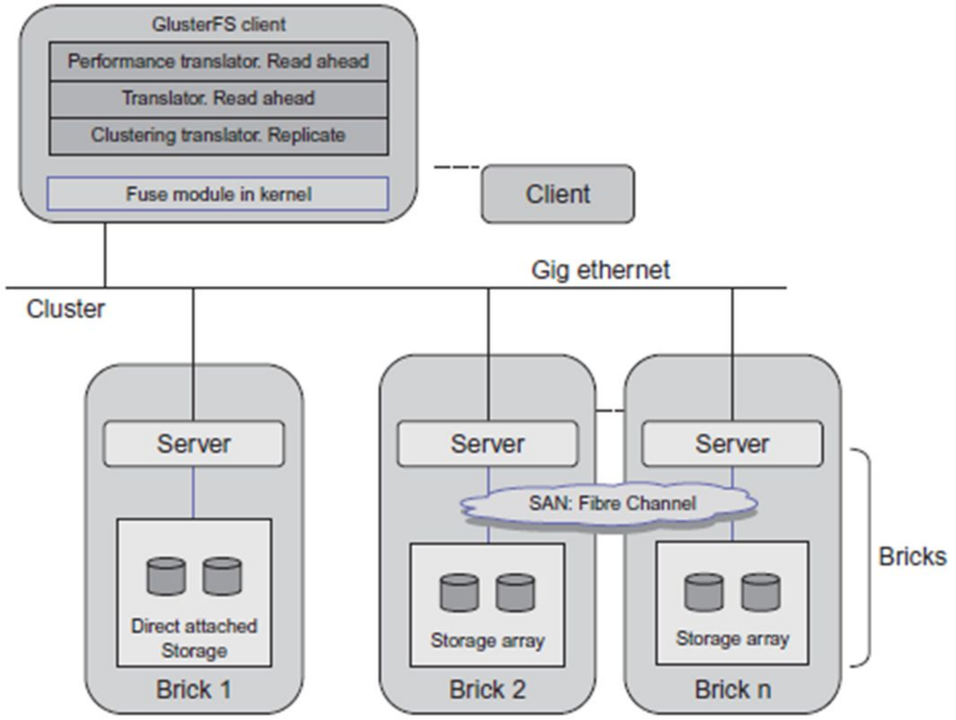
For scenario based questions

- **Scenario**: Proton Computed Tomography (pCT) image can require the **transfer of 100GB of data** and use of approximately **120 GPU-enabled compute nodes** the use of a commercial cloud as a **scalable and cost-efficient** platform for pCT reconstruction.

- **Problem:**The nature of proton therapy means that **demand for such a service** is irregular and comes from potentially **hundreds of clients worldwide**

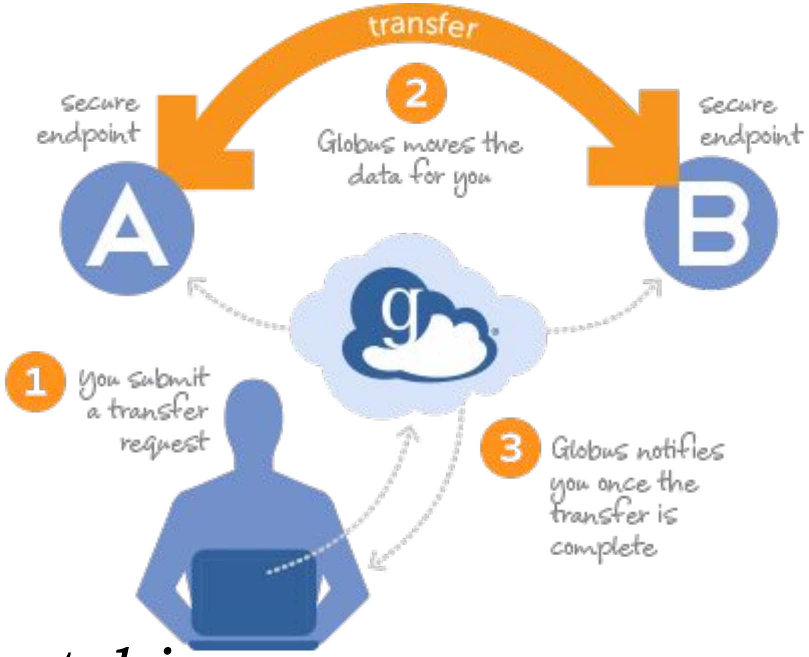
- **Amazon Elastic Compute Cloud:**EC2 has incorporated **cluster computing instances** aimed toward HPC computing applications.
- **pCT Reconstruction Instances Types:** GPU-enabled, MPI application were identified requirements for **high-CPU, high-memory, and GPU-enabled instances** as well as low latency between instances.
- **Amazon EC2 GPU-enhanced cluster** compute instance, termed CG1.
- *CG1 instances include two Intel Xeon X5570, quad-core CPUs with hyperthreading, 22.5GB of RAM, and two NVIDIA Tesla M2050 GPUs, each containing 3GB of RAM.*

- **Shared File System: GlusterFS**, an opensource distributed file system that provides scalable and high performance access to files
- The **GlusterFS model relies on one or more storage bricks (or servers)** that allow client applications, in this case the pCT reconstruction worker nodes, to mount the data source



- **Data Upload/Download:** **Globus moves data** between Globus endpoints, the name given to a resource on which a Globus agent is installed

1. Globus automatically **TUNES PARAMETERS**
2. *Maximize bandwidth usage*
3. *Manages security configurations*
4. *Provides automatic fault recovery*
5. *Encrypts data channels*
6. *Notifies users of completion and problems*
7. *Ensures that files are transferred reliably by matching checksums*



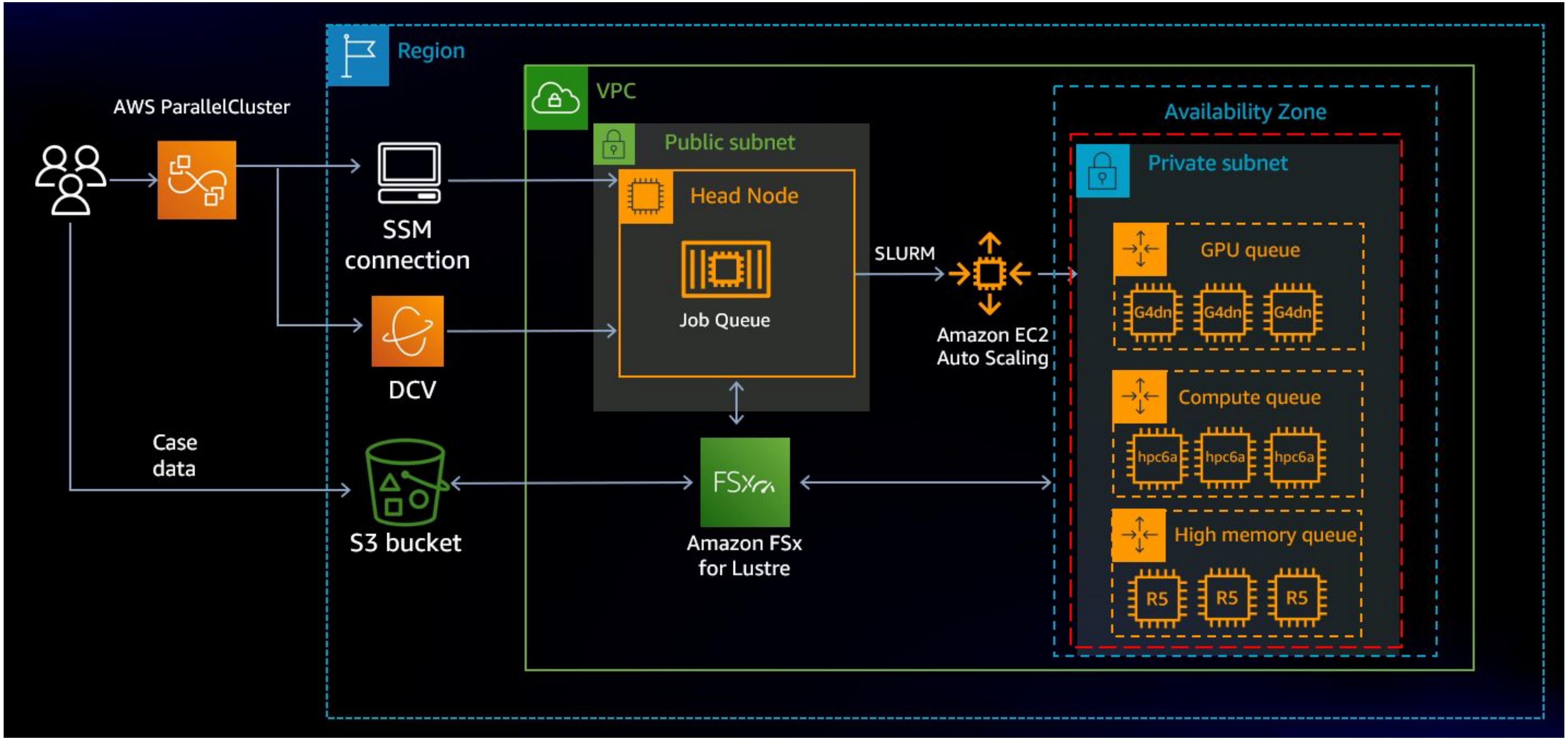
- **Cloud Images and Elastic Scale Out**: Amazon, these snapshots are referred to as Amazon Machine Images (AMIs)

https://ieeexplore.ieee.org/ielaam/6245519/8307205/7160740-aam.pdf?tag=1

AWS Services used in **CfnCluster**

- CfnCluster ("**cloud formation cluster**") is a framework that deploys and maintains high performance computing clusters on Amazon Web Services (AWS)

- **AWS CloudFormation (Basic requirement for CFN cluster)**
  - *AWS Identity and Access Management (IAM)*
  - *Amazon SNS (Amazon Simple Notification Service)*
  - *Amazon SQS (Amazon Simple Queue Service)*
  - *Amazon EC2*
  - *Auto Scaling*
  - *Amazon EBS*
  - *Amazon S3*
  - *Amazon DynamoDB*


Region
VPC
Availability Zone
AWS ParallelCluster
Public subnet
Private subnet
Head Node
SSM
connection
Job Queue
SLURM
GPU queue
G4dn
G4dn
G4dn
Amazon EC2
Auto Scaling
DCV
Compute queue
hpc6a
hpc6a
hpc6a
Case
data
S3 bucket
Amazon FSx
for Lustre
High memory queue
R5
R5
R5

Components in **cfncluster**

- AWS **Systems Manager Agent (SSM Agent)** is Amazon software that runs on Amazon Elastic Compute Cloud (Amazon EC2) **instances, edge devices, and on-premises**.
- DCV is a high-performance **remote display protocol** that provides customers with a **SECURE WAY** to deliver **remote desktops and application** streaming from any cloud or data center to any device, over varying network conditions.
- **SLURM WORKLOAD MANAGER**, formerly known as **Simple Linux Utility for Resource Management (SLURM), or simply Slurm**, is a free and open-source job scheduler for Linux and Unix-like kernels, used by many of the world's supercomputers and computer clusters
- **AWS ParallelCluster** is tested with Slurm configuration parameters, which are provided by default **creation, management, and scaling of high-performance computing (HPC) clusters in the AWS cloud**

Deploying an HPC Cluster on Amazon

- CloudFormation service, which enables the automated deployment of complex *collections of related services, such as*
- ***multiple EC2 instances,***
- ***load balancers,***
- ***special subnetworks connecting these components,***
- ***security groups* that apply across the collection.**

Deploying an HPC Cluster on Amazon

- **CfnCluster (CloudFormation Cluster)** Python scripts that you can install and run on your Linux, Mac, or Windows computer to invoke CloudFormation, as follows, to build a private, custom HPC cluster
- ***`sudo pip install cfncluster`***
- ***`cfncluster configure`***

Deploying an HPC Cluster on Amazon

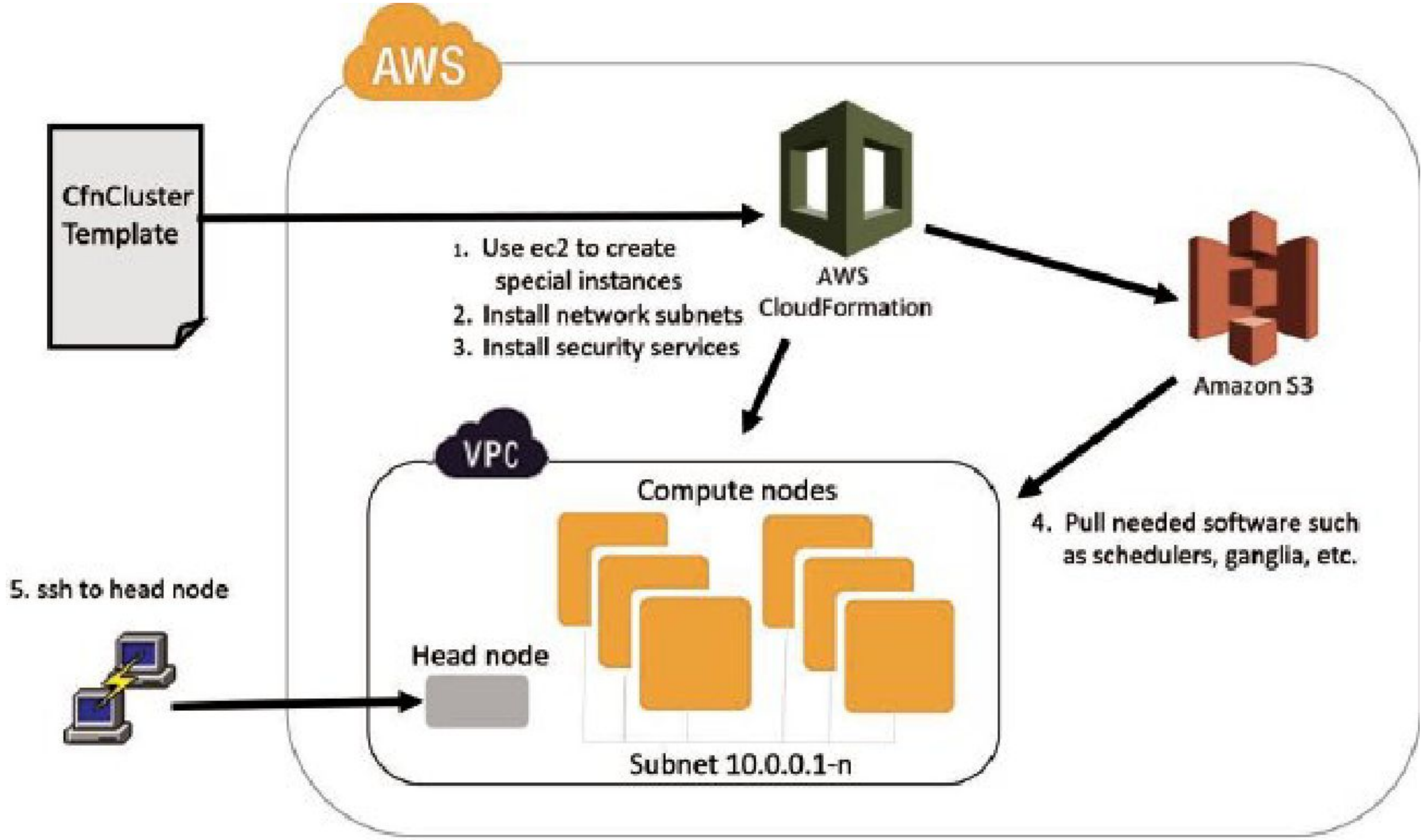
Yet Another Markup Language
instance types, networking options, the scheduler to use, and storage configurations

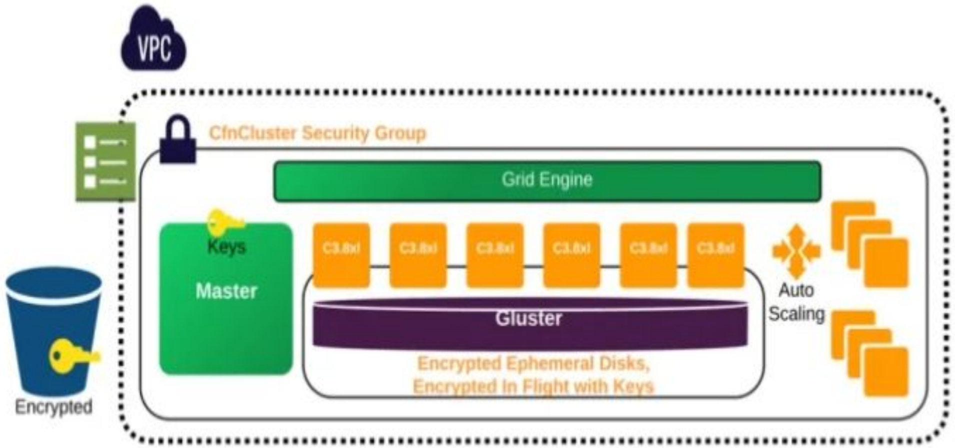
Cluster Creation
,
such as VPCs, subnets, EC2 instances, and storage

Job Submission

Cluster Management
manage the cluster, including scaling, updating configurations, and deleting the cluster

cfncluster **CREATE MYCLUSTER**





- CloudFormation steps involved in launching a private HPC cloud from a **CfnCluster template**.
- The **"create"** command returns a **Ganglia URL**.
- Ganglia is a well-known and frequently **USED CLUSTER MONITORING TOOL.**
- Following that link takes you to a Ganglia view of your HPC cluster.
- The default settings for a new cluster are **AUTOSCALE** compute nodes and the gridEngine scheduler.
- With **AUTOSCALE**, compute nodes are shut down when not in use, and new nodes are started when load increases.

- The Ganglia open source project is a ***scalable, distributed system designed to monitor clusters*** and grids while minimizing the impact on their performance.
- When you enable Ganglia on your cluster, you can **generate reports and view the performance of the cluster** as a whole, as well as inspect the performance of individual node instances.
- Ganglia is also configured to **ingest and visualize Hadoop and Spark metrics**

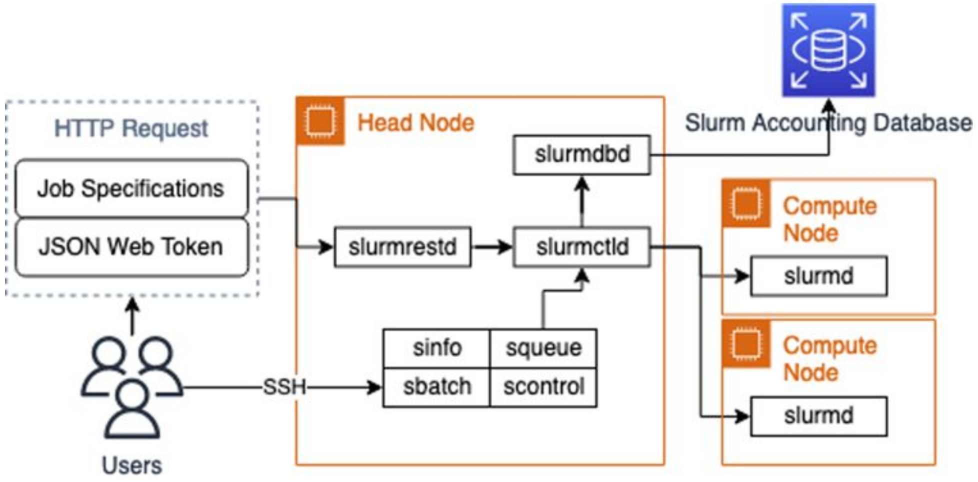
- Deploy a new cluster with better compute nodes and a better scheduler.
    - `cfncluster delete mycluster`
- To create a new and improved deployment that you can use for MPI programs, find the directory **~/.cfncluster** on your PC and edit the file config.

Instance Family	Current Generation Instance Types
General purpose	t2.nano | t2.micro | t2.small | t2.medium | t2.large | m4.large | m4.xlarge | m4.2xlarge | m4.4xlarge | m4.10xlarge | m3.medium | m3.large | m3.xlarge | m3.2xlarge
Compute optimized	c4.large | c4.xlarge | c4.2xlarge | c4.4xlarge | c4.8xlarge | c3.large | c3.xlarge | c3.2xlarge | c3.4xlarge | c3.8xlarge
Memory optimized	r3.large | r3.xlarge | r3.2xlarge | r3.4xlarge | r3.8xlarge
Storage optimized	i2.xlarge | i2.2xlarge | i2.4xlarge | i2.8xlarge | d2.xlarge | d2.2xlarge | d2.4xlarge | d2.8xlarge
GPU instances	g2.2xlarge | g2.8xlarge

Slurm Workload Manager

- The Slurm Workload Manager by SchedMD is a popular HPC scheduler and is supported by **AWS ParallelCluster**, an elastic HPC cluster management service offered by AWS.
- It's designed to **schedule, allocate, and manage computational tasks across clusters of computers**.
- Traditional HPC workflows involve ***logging into a head node and running shell commands to submit jobs to a scheduler and check job status***.
- Modern distributed systems often use **representational state transfer (REST) API operations** to programmatically communicate between system components.

The key objectives of the process



- Job Scheduling
- Resource Allocation
- Scalability
- **Fault Tolerance**

- Supercomputing Centers
- Enterprise with large-scale data processing needs

compute_instance_type = c3.xlarge **initial_queue_size = 4** **maintain_initial_size = true** **scheduler = slurm**	• **c3.xlarge instance type** supports what Amazon calls enhanced networking, which means that it runs on hardware and with software that support **Single-Root I/O Virtualization (SR-IOV).** • VM image because the default contains all libraries needed for HPC MPI-style computing compute nodes to stay around and not be managed by autoscale, and that you want Slurm to be the scheduler
cfncluster create mycluster	• The cluster's head node, using **the key pair that we used to create the cluster.** • On a pc you can use **PUTTY**, and on a mac you can use ssh from the command line. • The user is **EC2-USER**. • First you need to set up some **PATH INFORMATION**.

```
export PATH=/usr/lib64/mpich/bin:$PATH
export LD_LIBRARY_PATH=/usr/lib64/mpich/lib
export I_MPI_PMI_LIBRARY=/opt/slurm/lib/libpmi.so
```

#include <stdio.h>
#include <mpi.h>
#include <stdlib.h>
main(int argc, char **argv)
{
char hostname[1024];
gethostname(hostname, 1024);
printf("%s\n", hostname);
}
	to know the local IP addresses of your compute nodes.
	Create a file called **ip-print.c**
	
	if your cluster has **16 nodes, run**
	1024 size of the buffer , safe, generous allocation that ensures compatibility
	The output file machines should then contain **multiple IP addresses** of the form 10.0.1.x, where x is a number between 1 and 255, one for each of your compute nodes.

mpicc ip-print.c **srun -n 16 /home/ec2-user/a.out >** machines	the **Slurm command srun** to run **copies of this program across the entire cluster**
mpicc ring.c mpirun -np 7 -machinefile ./machines /home/ec2-user/a.out	• MPI program **processes send a message around in a "ring" pattern** • Program starts with MPI node **0 and sends the number -1 to MPI node 1.** • MPI node 1 sends 0 to node 2, node 2 sends 1 to node 3, and so on.

Command	What it does
sinfo	reports the state of partitions and nodes
squeue	reports the state of jobs in the batch queue
sbatch *script*	submits a job script
scancel *jobid*	cancels a pending or running job *jobid*
sacct	reports accounting information about active and completed jobs
srun	submits job for execution or initiate job steps in real time (used inside batch script)

Directive	Description
Partition	#SBATCH --partition=*partitionname* OR #SBATCH –p *partitionname*
Account	#SBATCH --account=*accountname* OR #SBATCH –A *accountname*
Wall time	#SBATCH --time=01:00:00 OR #SBATCH -t 1:00:00
Node count	#SBATCH --nodes=2 OR #SBATCH -N 2
Tasks per node	#SBATCH --ntasks-per-node=12
Constraint	#SBATCH --constraint="c12" OR #SBATCH -C "c12"
Total tasks count	#SBATCH --ntasks=24 OR #SBATCH -n 24
Memory	#SBATCH --mem=24576 (NOTE: memory given in MB)
Mail options	#SBATCH --mail-type=FAIL,BEGIN,END
Mail user	#SBATCH --mail-user=user@mail.com
Job name	#SBATCH --job-name=*jobname* OR #SBATCH –J *jobname*
Stderr	#SBATCH -e slurm-%j.err-%N (where %j is job number and %N is first node name)
Stdout	#SBATCH -o slurm-%j.out-%N

```c
#include <mpi.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

int main(int argc, char** argv) {
  // Initialize the MPI environment
  MPI_Init(NULL, NULL);
  // Find out rank, size
  int rank, world_size, number;
  MPI_Comm_rank(MPI_COMM_WORLD, &rank);
  MPI_Comm_size(MPI_COMM_WORLD, &world_size);
  char hostname[1024];
  gethostname(hostname, 1024);

  // We assume at least two processes for this task
  if (world_size < 2) {
    fprintf(stderr, "World size must be >1 for %s\n", argv[0]);
    MPI_Abort(MPI_COMM_WORLD, 1);
  }

  if (rank == 0) {
    // If we are rank 0, set number to -1 & send it to process 1
    number = -1;
    MPI_Send(&number, 1, MPI_INT, 1, 0, MPI_COMM_WORLD);
  }
  else if (rank > 0 && rank < world_size) {
    MPI_Recv(&number, 1, MPI_INT, rank-1, 0, MPI_COMM_WORLD,
                 MPI_STATUS_IGNORE);
    printf("Received number %d from process %d on node %s\n",
               number, rank-1, hostname);
    number = number+1;
    if (rank+1 < world_size)
      MPI_Send(&number, 1, MPI_INT, rank+1, 0, MPI_COMM_WORLD);
  }
  MPI_Finalize();
}
```

Deploying an HPC Cluster on Azure

- The first is to use Azure's service deployment orchestration service, Quick Start. **(FIRST APPROACH)**
- Like Amazon CloudFormation, it is based on templates.
- The **templates are stored in GitHub and can be invoked directly from the GitHub page**

Deploy a slurm cluster

Deploy to Azure | Visualize

1. Fill in the 3 mandatory parameters - public DNS name, a storage account to hold VM image, and admin user password.
2. Fill in other info and click "OK".

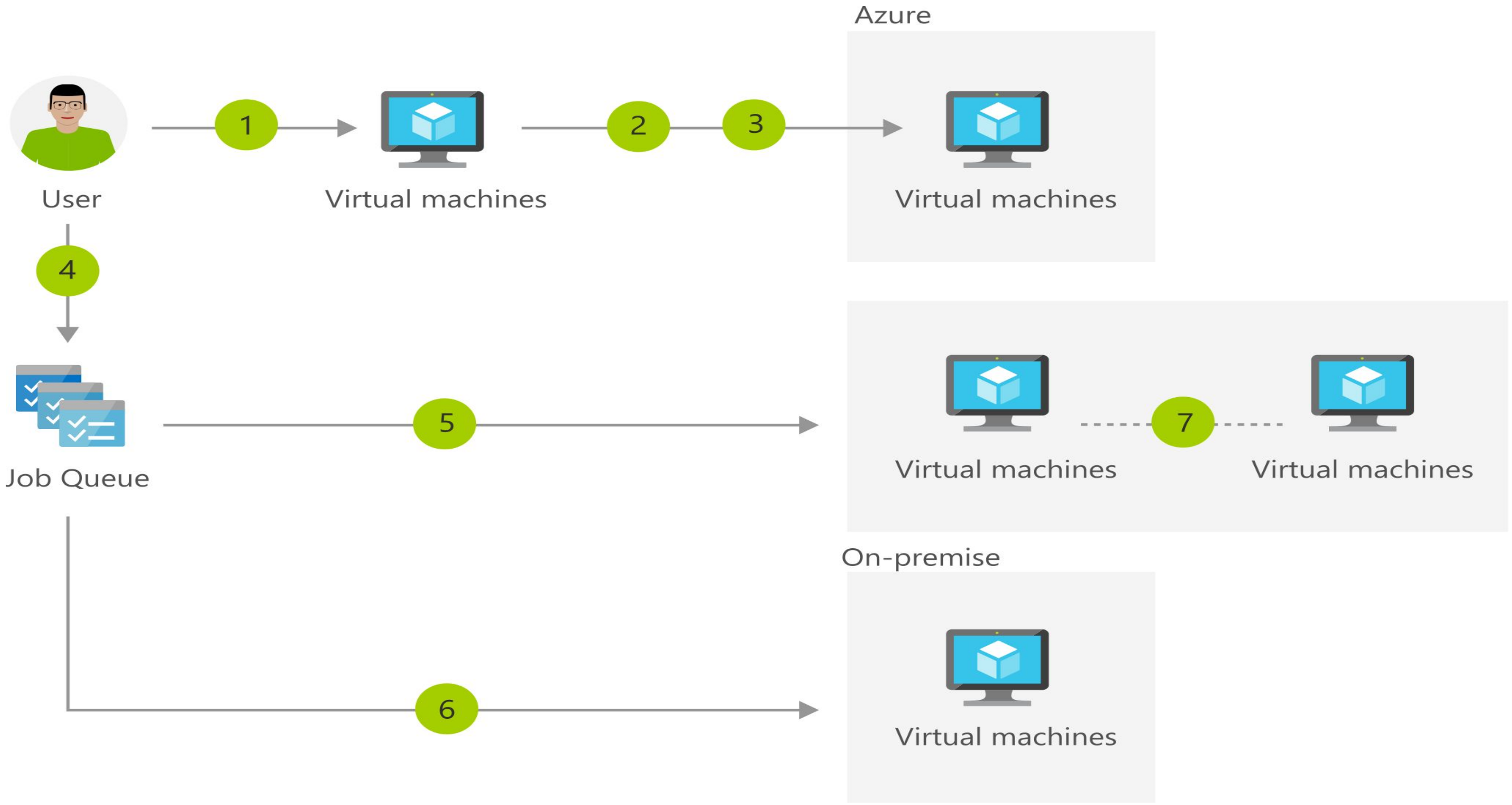
Using the cluster

Simply SSH to the master node and do a srun! The DNS name is ***dnsName.location***.cloudapp.azure.com, for example, yidingslurm.westus.cloudapp.azure.com.

- When you click **Deploy to Azure**, you are taken to the Azure login page
- Then directly to the web form for completing the **SLURM CLUSTER DEPLOYMENT**.
- Enter the names for *the* ***new resource group*** *that defines your cluster*
- *The* ***number and type of compute nodes***
- *Few details* ***about the network***.

- HPC computing on Azure **IS AZURE BATCH**, which supports the management of large pools of VMs that can handle large batch jobs, such as many task-parallel jobs **(SECOND APPROACH)**
- *A managed tool that you can use to **AUTOSCALE DEPLOYMENTS** and **set policies for job scheduling**.*
- *The Azure Batch service **handles provisioning, assignment, runtimes, and monitoring of your workloads.***
- Upload your application binaries and input data to Azure storage.
- Define the pool of **compute VMs** that you want to use, specifying the desired VM size and OS image
- Define a **Job, a container** for tasks executed in your VM pool.
- Create tasks that are loaded into the Job and executed in the VM pool.

Sample Diagram for Cluster formation in Azure



Steps for 2nd Approach

on-premises head node

→ compute nodes to the cluster

→ Start the compute

→ Submit jobs

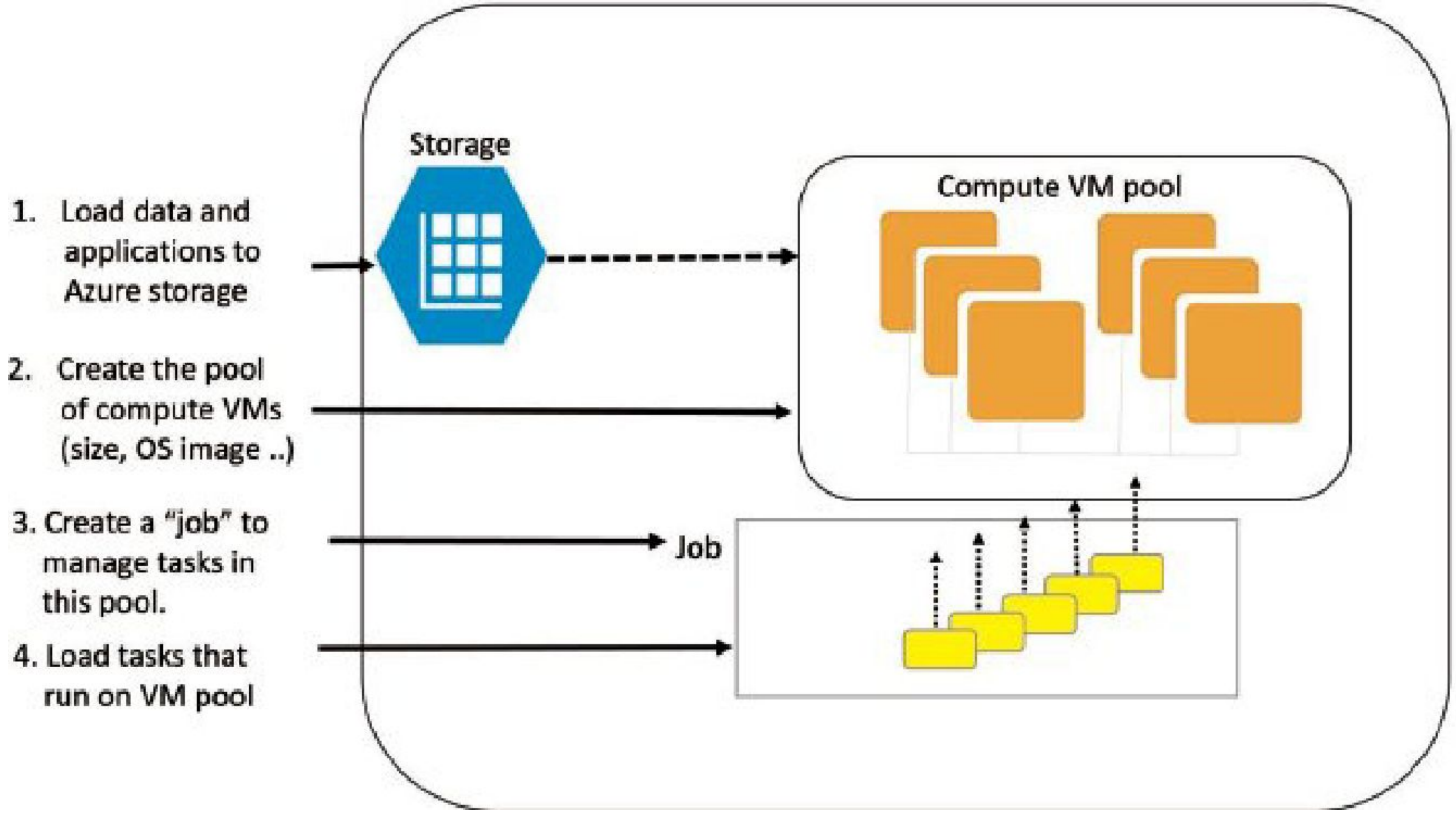
→ on-premises and Azure nodes, based upon the

→ Monitor the job progress

→ Stop the compute nodes

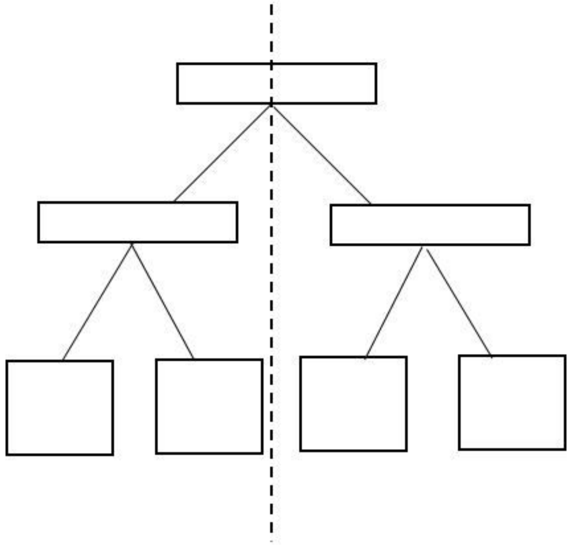
CONFIGURE AUTOSCALING

Steps in creating and executing an Azure batch job



Scaling Further

- A quantity called the network bisection bandwidth is a measure of how much **data traffic can flow from** one half of the supercomputer to the other half in a specified period of time.
- The networks used in supercomputers have an extremely **HIGH BISECTION BANDWIDTH.**
- The first cloud data centers had **LOW BISECTION BANDWIDTH.**

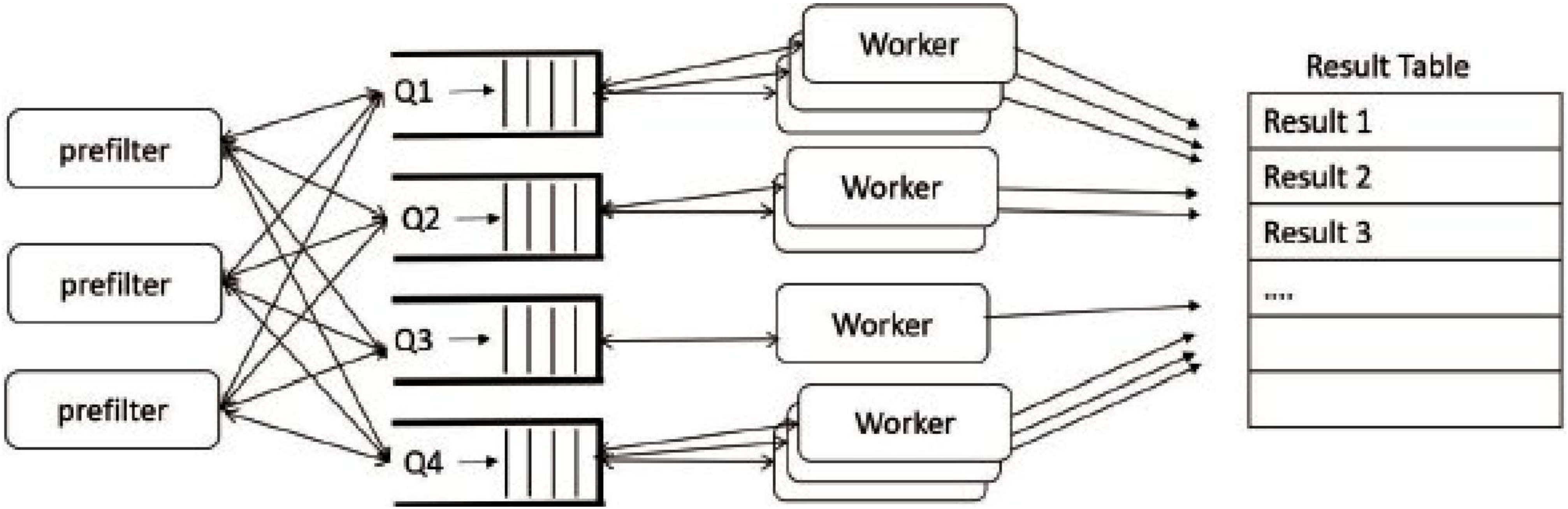


- **Service level agreement (SLA)** that cloud providers make with their users.
- Supercomputers commit to a specific
- ***Processor type***
- ***Network bandwidth***
- ***Bisection width***
- ***Latency***
- ***Allowing the user to predict an application's performance profile with a fair degree of certainty***

Many Task Parallelism

- Analyze **many data samples**.
- Each analysis task can be performed **independently of all the other tasks.**
- You place all **data samples in a queue** in the cloud, and then start a **large number of worker VMs or containers**.
- We refer to this as many task parallelism, but it is also known as **bag of tasks parallelism and manager worker parallelism.**

Simple many task execution model



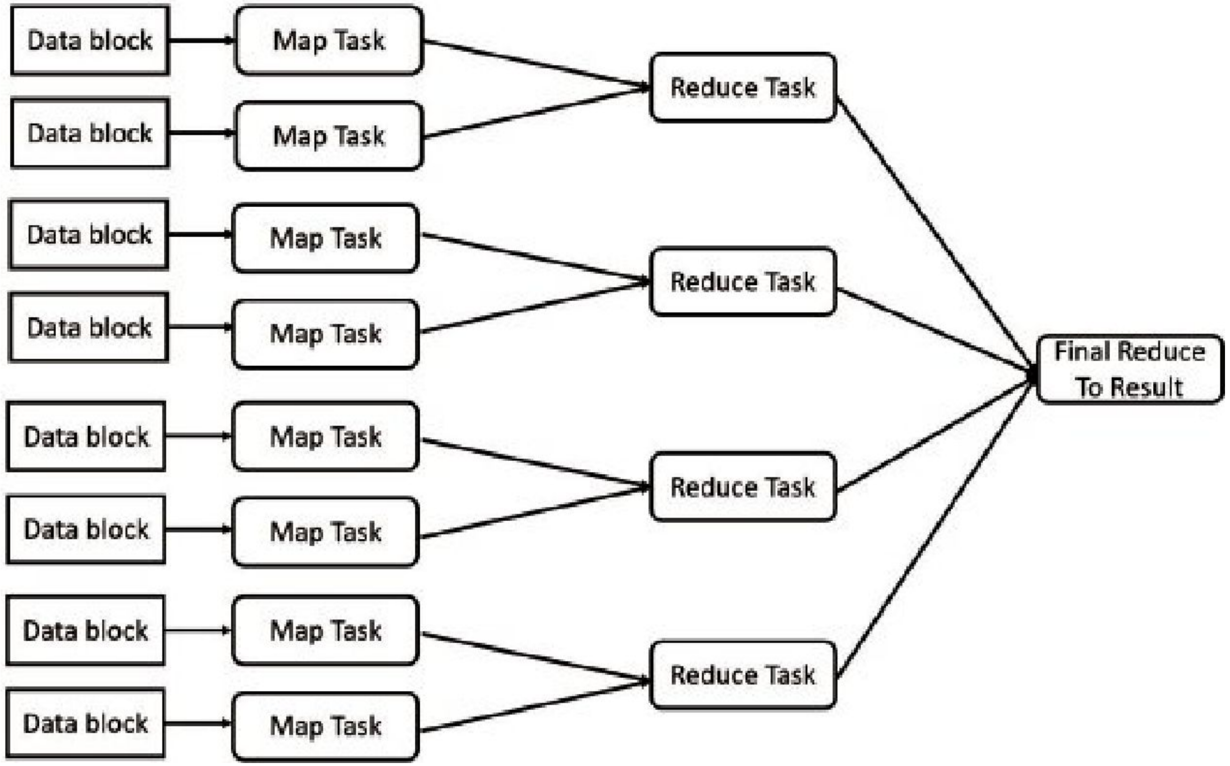
MapReduce and Bulk Synchronous Parallelism

- A slightly more sophisticated approach to parallelism is based on a concept called bulk-synchronous parallelism (BSP).
- This is important when **worker tasks must periodically synchronize and exchange data with each other**.
- The point of synchronization in a BSP computation is called a **BARRIER**
- Because **no computation is allowed to proceed until all computations reach the synchronization point**

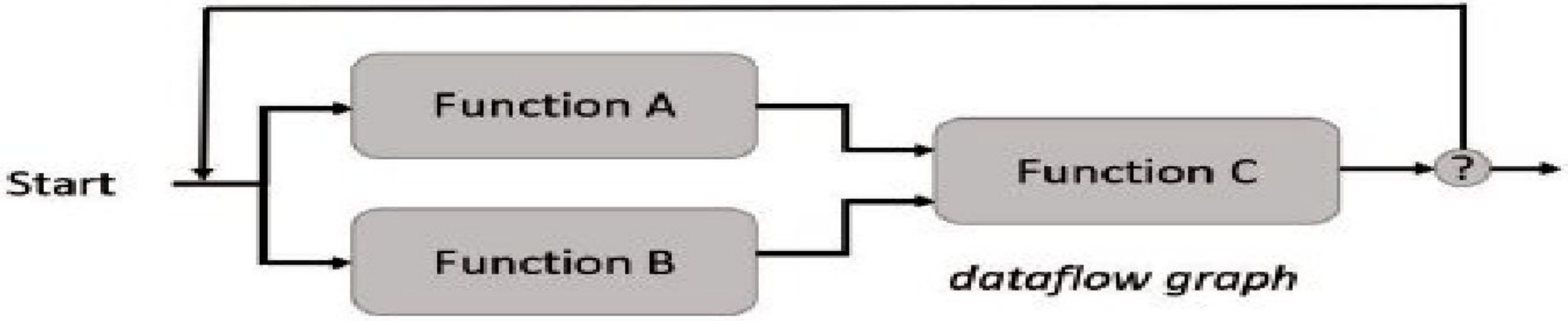
- MapReduce is a special case of BSP computing.
- Say you have a sequence of data objects Xi for i = 1..n and you want to apply a function f(x) to each element.
- Assume the result is a value in an associative ring like the real numbers, for which we can compose objects, and we want to compute the sum.

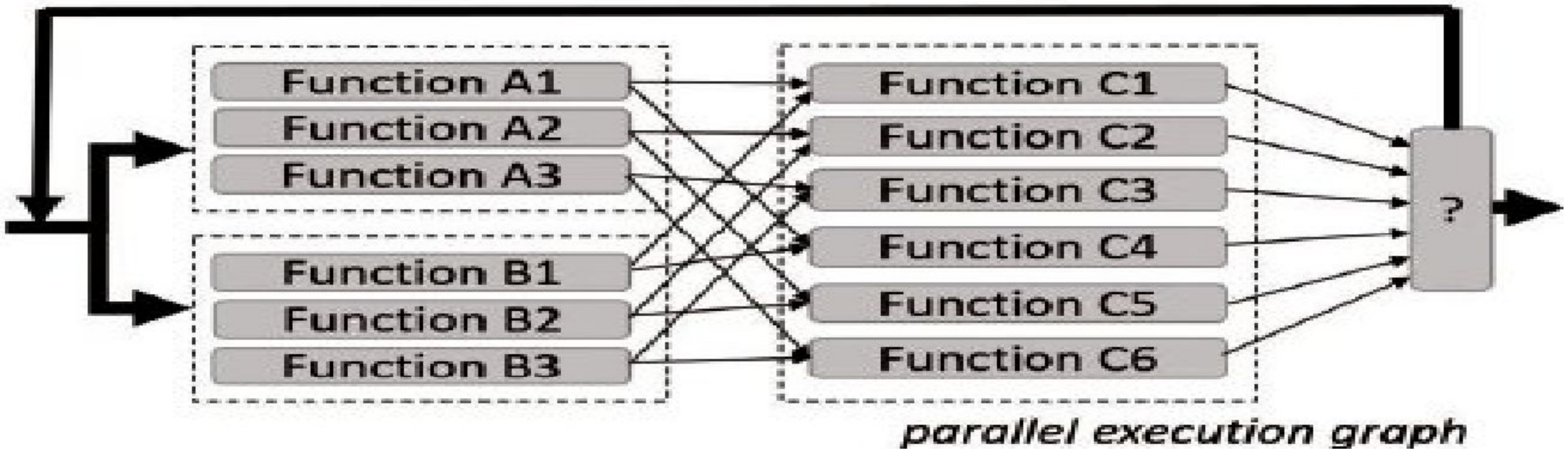
$$\sum_{i=1}^{n} f(x_i)$$

- A MapReduce computation starts with a distributed data collection partitioned into non-overlapping blocks.
- It then maps a supplied function over each block



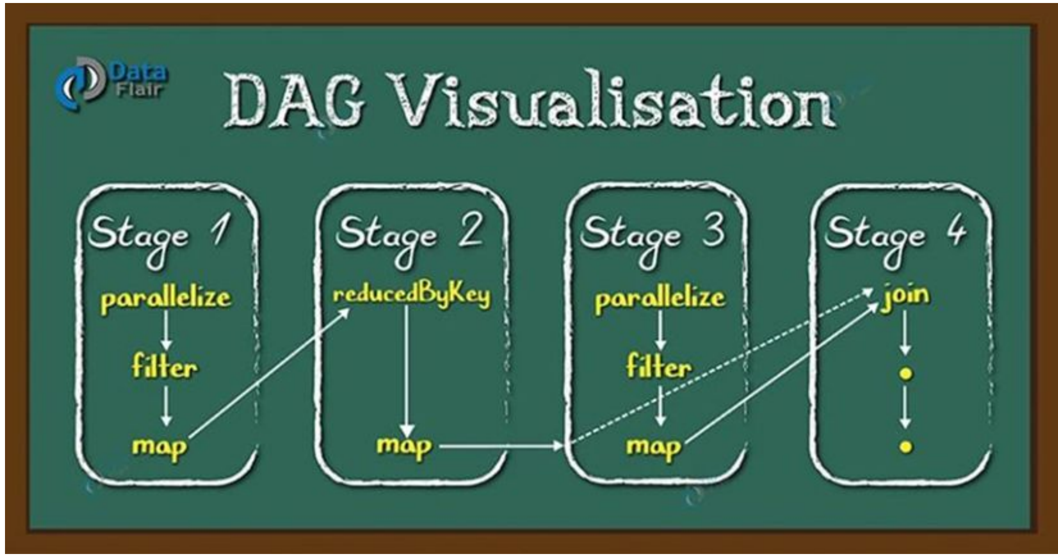
Graph Dataflow Execution and Spark



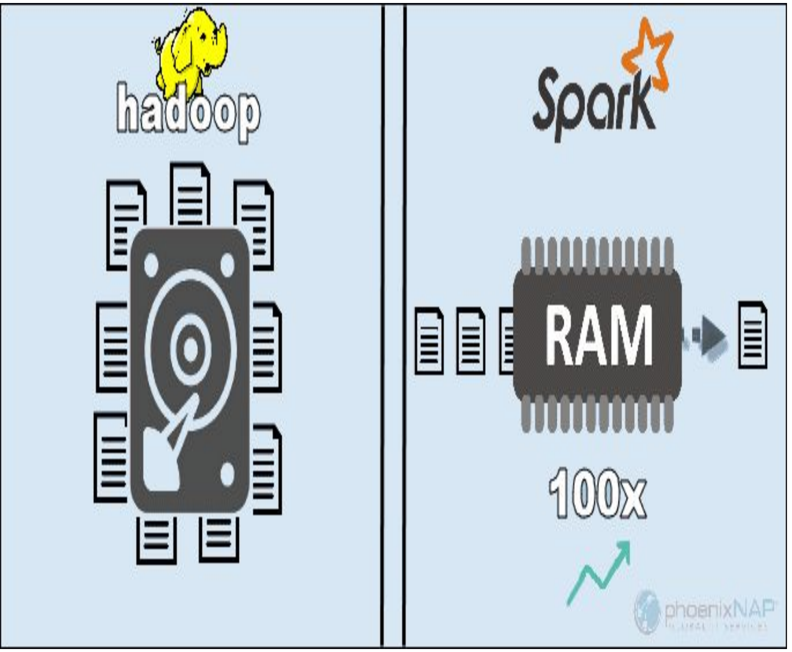


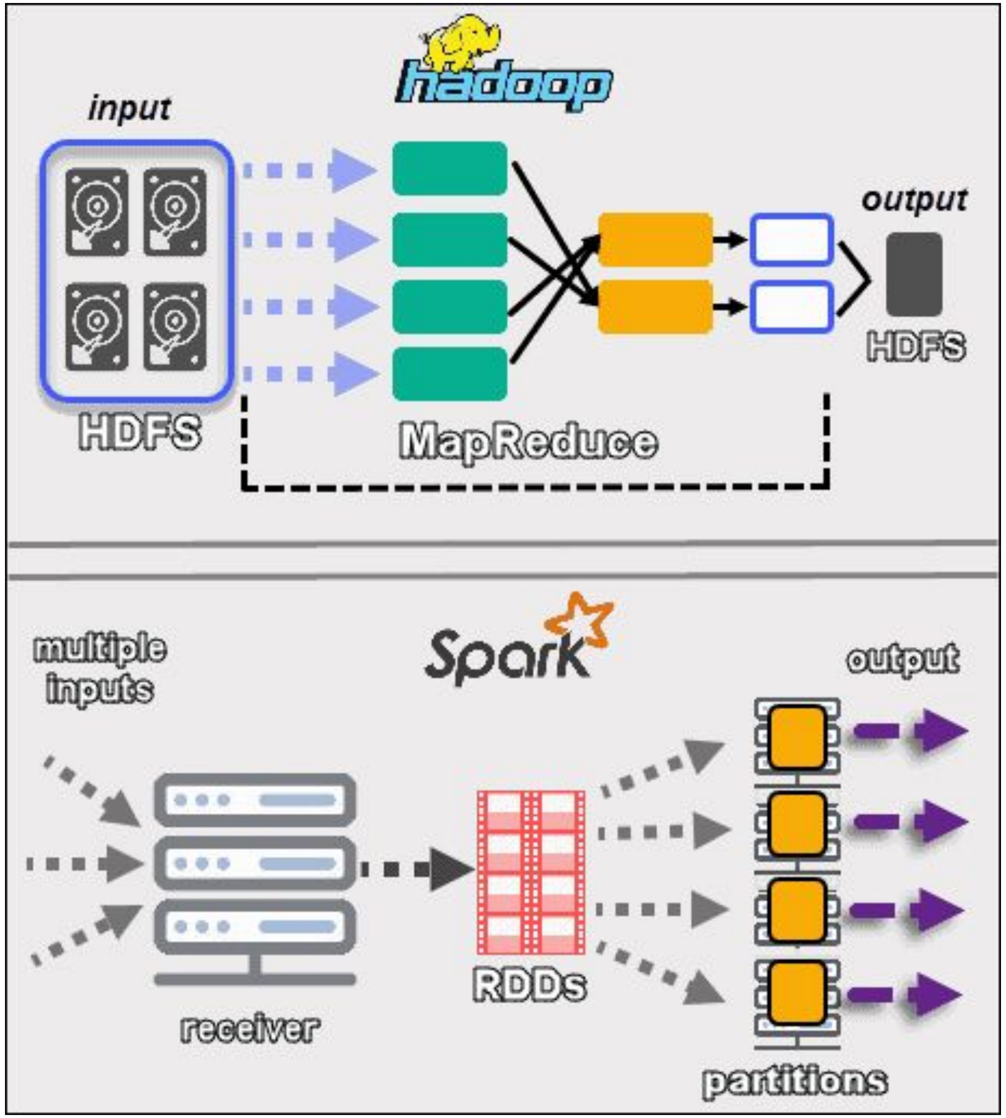
DAG Visualization in SPARK

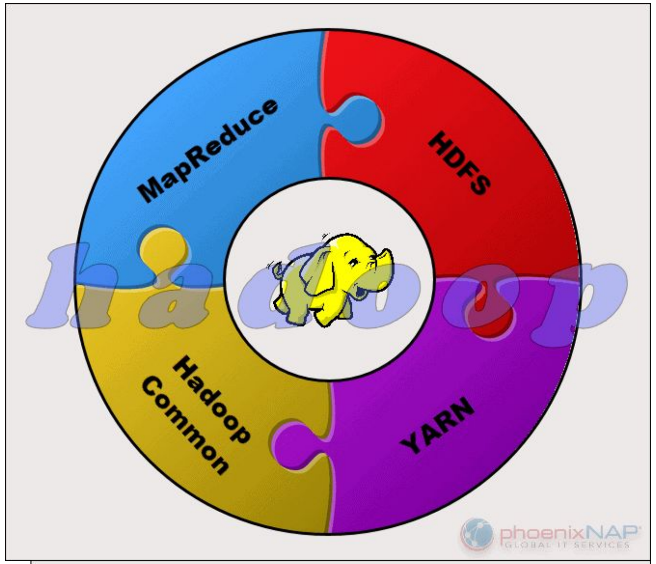
Stage 1	• Parallelize • Filter • Map
Stage 2	• ReducedBykey • Map
Stage 3	• Join

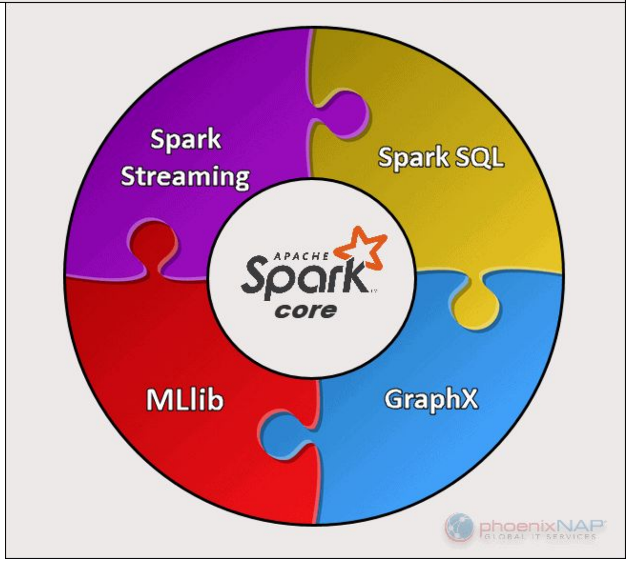


Name	Description
HADOOP	1. Apache Hadoop is a platform that **handles large datasets in a distributed fashion.** 2. The framework uses MapReduce to **split the data into blocks and assign the chunks to nodes across a cluster.** 3. MapReduce then processes the data in parallel on each node **to produce a unique output.** 4. Every machine in a cluster both stores and processes data. 5. Hadoop stores the data to disks using **HDFS**.
SPARK	1. It is designed for **fast performance and uses RAM for caching and processing data.** 2. Spark performs different types of big data workloads. 3. This includes **MapReduce-like batch processing**, as well **as real-time stream processing, machine learning, graph computation, and interactive queries** 4. The data structure that Spark uses is called **Resilient Distributed Dataset (RDD).**

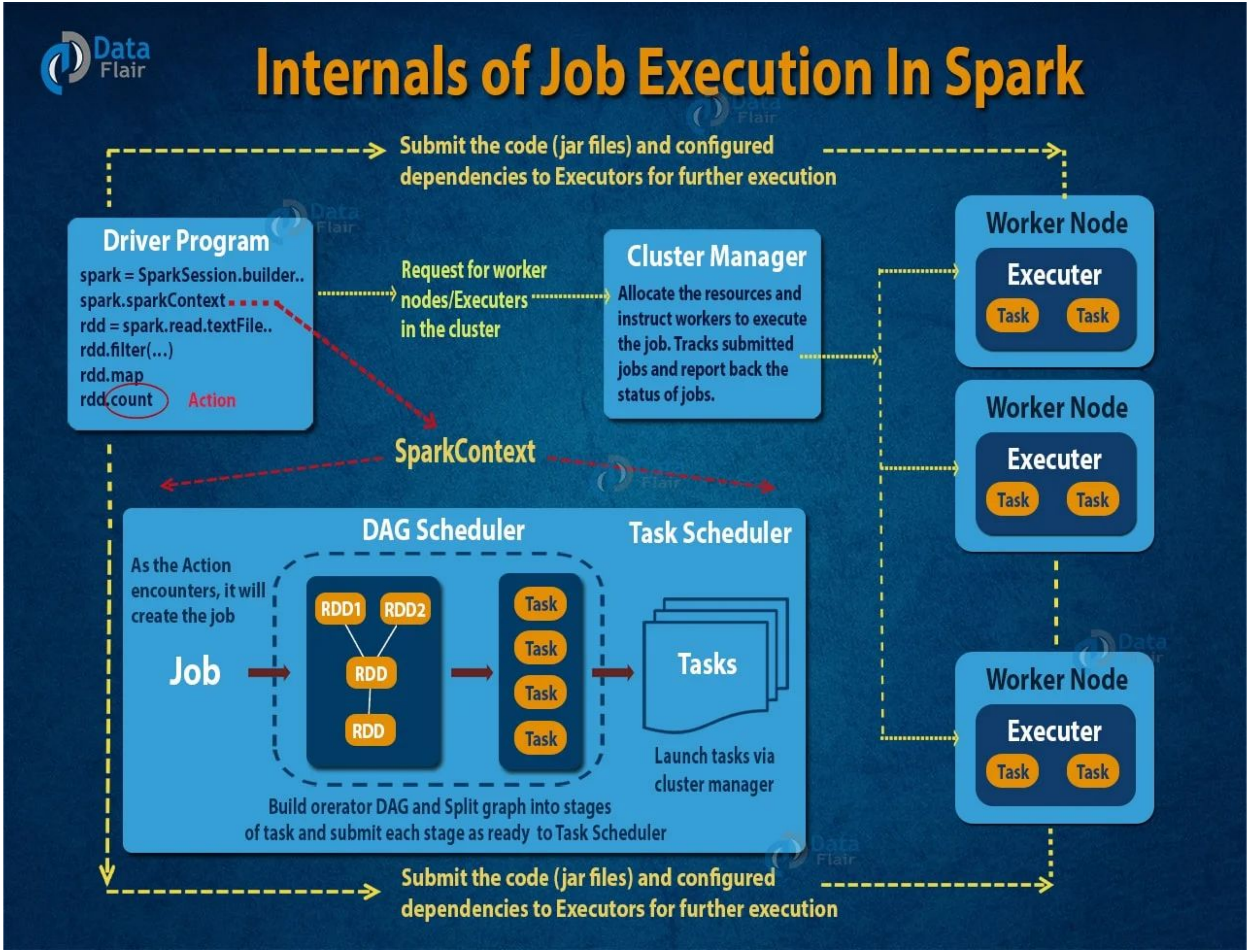

hadoop
Spark
RAM
100x
phoenixNAP


hadoop
input
HDFS
MapReduce
output
HDFS
multiple inputs
Spark
output
receiver
RDDs
partitions


MapReduce
HDFS
Hadoop Common
YARN
hadoop
phoenixNAP


Spark Streaming
Spark SQL
APACHE Spark core
MLlib
GraphX
phoenixNAP

Key Features	Apache Spark	Hadoop MapReduce
Speed	10–100 times faster than MapReduce	Slower
Analytics	Supports streaming, Machine Learning, complex analytics, etc.	Comprises simple Map and Reduce tasks
Suitable for	Real-time streaming	Batch processing
Coding	Lesser lines of code	More lines of code
Processing Location	In-memory	Local disk


Data Flair
Internals of Job Execution In Spark
Submit the code (jar files) and configured dependencies to Executors for further execution
Driver Program
spark = SparkSession.builder..
spark.sparkContext
rdd = spark.read.textFile..
rdd.filter(...)
rdd.map
rdd.count
Action
Request for worker nodes/Executers in the cluster
Cluster Manager
Allocate the resources and instruct workers to execute the job. Tracks submitted jobs and report back the status of jobs.
Worker Node
Executer
Task
Task
Worker Node
Executer
Task
Task
SparkContext
DAG Scheduler
Task Scheduler
As the Action encounters, it will create the job
Job
RDD1
RDD2
RDD
RDD
Task
Task
Task
Task
Tasks
Launch tasks via cluster manager
Build orerator DAG and Split graph into stages of task and submit each stage as ready to Task Scheduler
Worker Node
Executer
Task
Task
Submit the code (jar files) and configured dependencies to Executors for further execution

Why we need DAG with spark

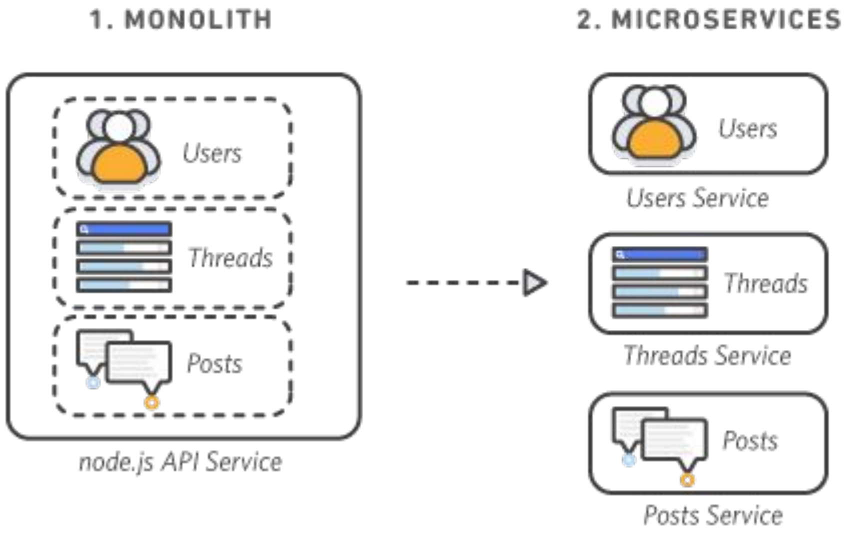
- Each MapReduce operation is **independent of each other** and HADOOP has no idea of which Map reduce would come next.
- Sometimes for some iteration, it is irrelevant to read and write back the immediate result between two map-reduce jobs.
- In such case, **the memory in stable storage (HDFS) or disk memory gets wasted**.
- In multiple-step, till the completion of the previous job all the jobs block from the beginning.
- The DAG in Spark is essential for **managing, optimizing, and executing distributed data processing workflows**.

- It allows Spark to efficiently **plan and execute complex data transformations by understanding the dependencies** between different stages of computation, ensuring fault tolerance, and optimizing resource usage across the cluster
- As a result, **COMPLEX COMPUTATION CAN REQUIRE** a long time with small data volume.
- While in Spark, a **DAG (Directed Acyclic Graph) of consecutive computation stages is formed**.
- In this way, we **optimize the execution plan**, e.g. to minimize shuffling data around.
- In contrast, it is done **manually in MapReduce** by tuning each MapReduce step.

- Spark **spark.apache.org** is a popular example of this style of computation.
- In Spark the control flow program is a **version of SQL, Scala, or Python and, consequently**, you can easily execute Spark programs from a Jupyter notebook.
- Dataflow graph as defined by the program and after the **parallelism is unrolled during execution**.
- Spark is also part of **Microsoft's HDInsight toolkit and is supported on Amazon Elastic MapReduce**.
- Microsoft documentation describes how to deploy Spark on Azure with Linux or Windows and from a Jupyter notebook.
- Because Spark is used **primarily for data analytics**

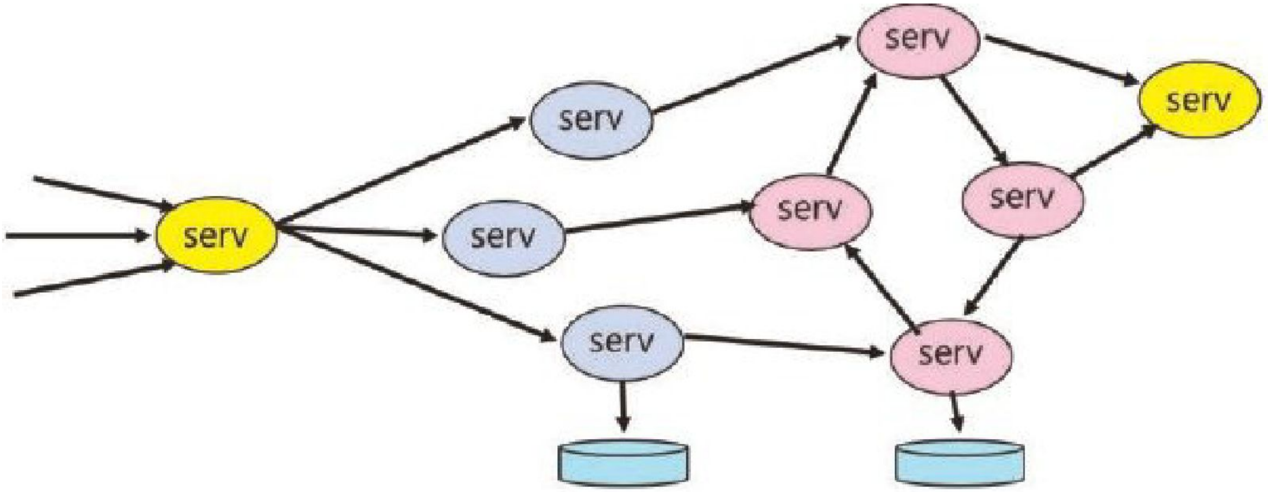
Agents and Microservices

- The cloud is designed to host applications that are realized **as scalable services**, such as a **web server or the backend of a mobile app.**
- Such applications accept connections from remote clients and, based on the client request, perform some computation and return a response



- **SCENARIO:** Another scenario is an application that processes events from **REMOTE SENSORS**, with the goal of informing a control system on how to respond, as when geosensors detecting ground motion tremors occurring in a significant pattern sound an earthquake warning.
- **PROBLEM:** *Multiple components:* ***sensor signal decoders, pattern analysis integrators, database searches, and alarm system interfaces.***
- **SOLUTION:** **PARALLEL PROGRAMMING** is like an asynchronous swarm of communicating processes or services distributed over a virtual network in the cloud.
- The individual processes **may be stateless, such as a simple web service, or stateful, as in the actor programming model**

Conceptual view of a swarm of communicating microservices or actors.

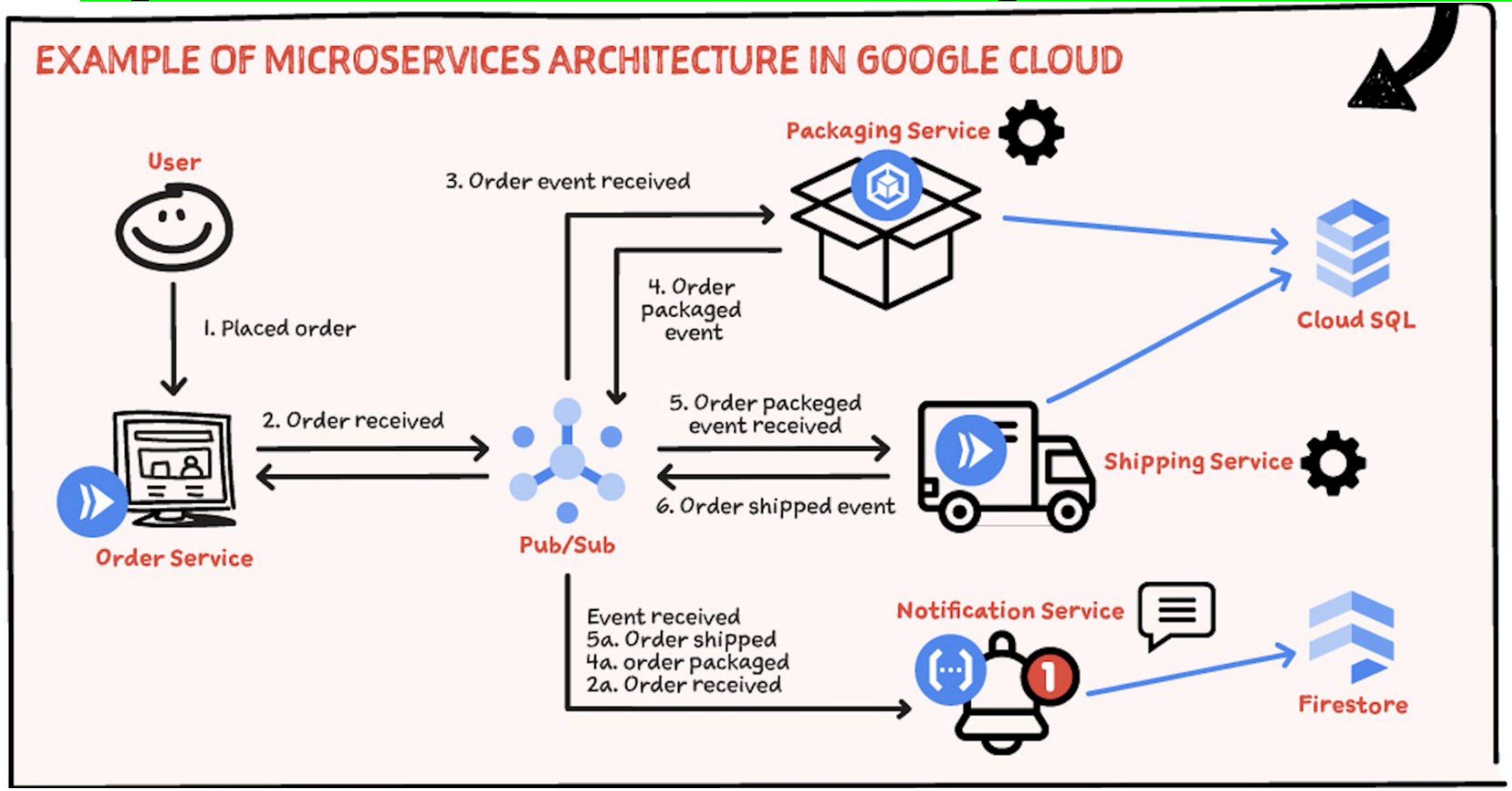


- The problem addressed by microservices is that of **HOW TO DESIGN AND BUILD A LARGE, HETEROGENEOUS APPLICATION THAT IS SECURE, MAINTAINABLE, FAULT TOLERANT, AND SCALABLE**.
- It is particularly important for **LARGE ONLINE SERVICES** that need to support thousands of concurrent users
- The microservice solution to this challenge is to **partition the application into *small, independent service components communicating with simple, lightweight mechanisms*.**
- The microservice paradigm design rules dictate that each microservice must be able to be *managed, replicated, scaled, upgraded, and deployed independently of other microservices.*

- Each microservice must have a single function and operate in a bounded context
- It has limited responsibility and limited dependence on other services.
- The communication mechanisms used by microservice systems are varied
- **REST web service calls**
- **RPC mechanisms such as google's swift**
- **Advanced Message Queuing Protocol (AMQP).**

Microservices and Container Resource Managers

- Amazon ECS container service, Google Kubernetes, Apache Mesos, and Mesosphere on Azure



Ex diagram for Microservices , it can be used for GKE

Managing Identity in a Swarm

- **Scenario:** If some microservices need to **access a queue** of events that you own, and others need to **interact with a database that you created**, then you need to pass part of your authority to those services so that they may make invocations on your behalf. **(SECURITY)**
- **Solution 1:** To pass these values as runtime parameters through a **secure channel to your remotely running application or microservice**
- **Problem in Solution 1:** First, microservices are designed to be **shut down when not needed and scaled up in number when the demand is high.**
- Need to automate the process of **fetching the credentials for each microservice reboot**

- **Solution 2:** Second, by **PASSING THESE CREDENTIALS**, you endow your remote service with all of your authority prefer to pass only a limited authority.

- **ROLE-BASED SECURITY:** This means is that you can **create special secure entities, called roles**
- That authorize **individuals, applications, or services** to access various cloud resources on your behalf.

A Simple Microservices Example

Scenario:

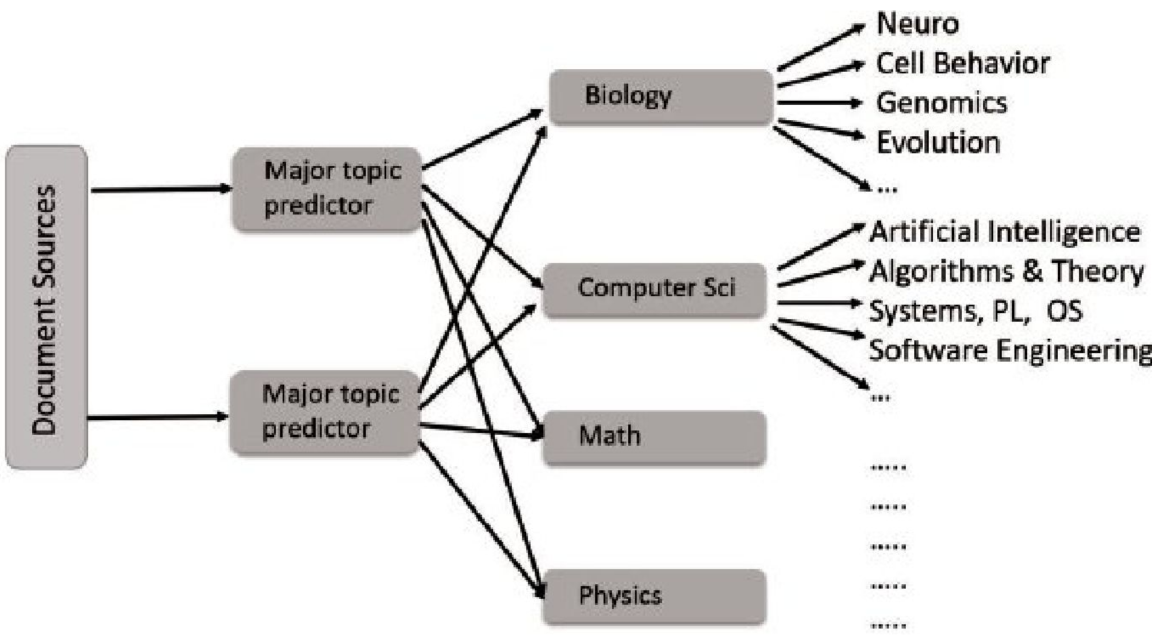
- When scientists send technical papers to scientific journals, the abstracts of these papers often make their way onto the Internet as a stream of news items, to which one can subscribe via RSS feeds.
- A major source of **high-quality streamed science data is arXiv arxiv.org, a collection of more than one million open-access documents.**
- Other sources include the Public Library of Science (PLOS one), Science, and Nature, as well as online news sources.
- We have downloaded a small collection of records from arXiv, each containing a paper title, an abstract, and, for some, a scientific topic as determined by a curator.

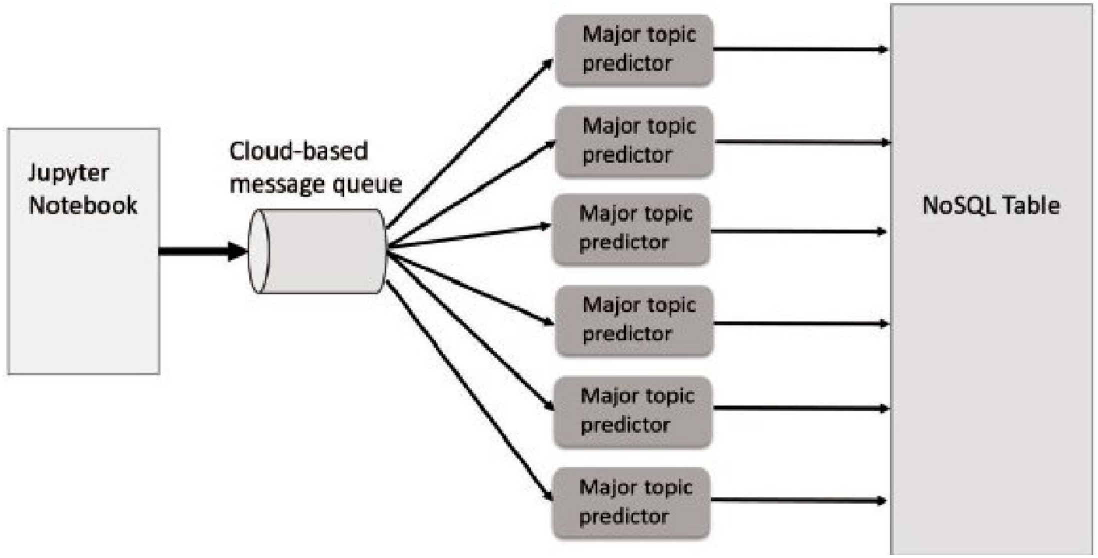
Solution: Our goal is to **build a system that pulls document abstracts from the various feeds and then uses one set of microservices** to classify those abstracts into the major topics of physics, biology, math, finance, and computer science, and a second set to classify them into subtopic areas

A Simple Microservices Example

Requirements:

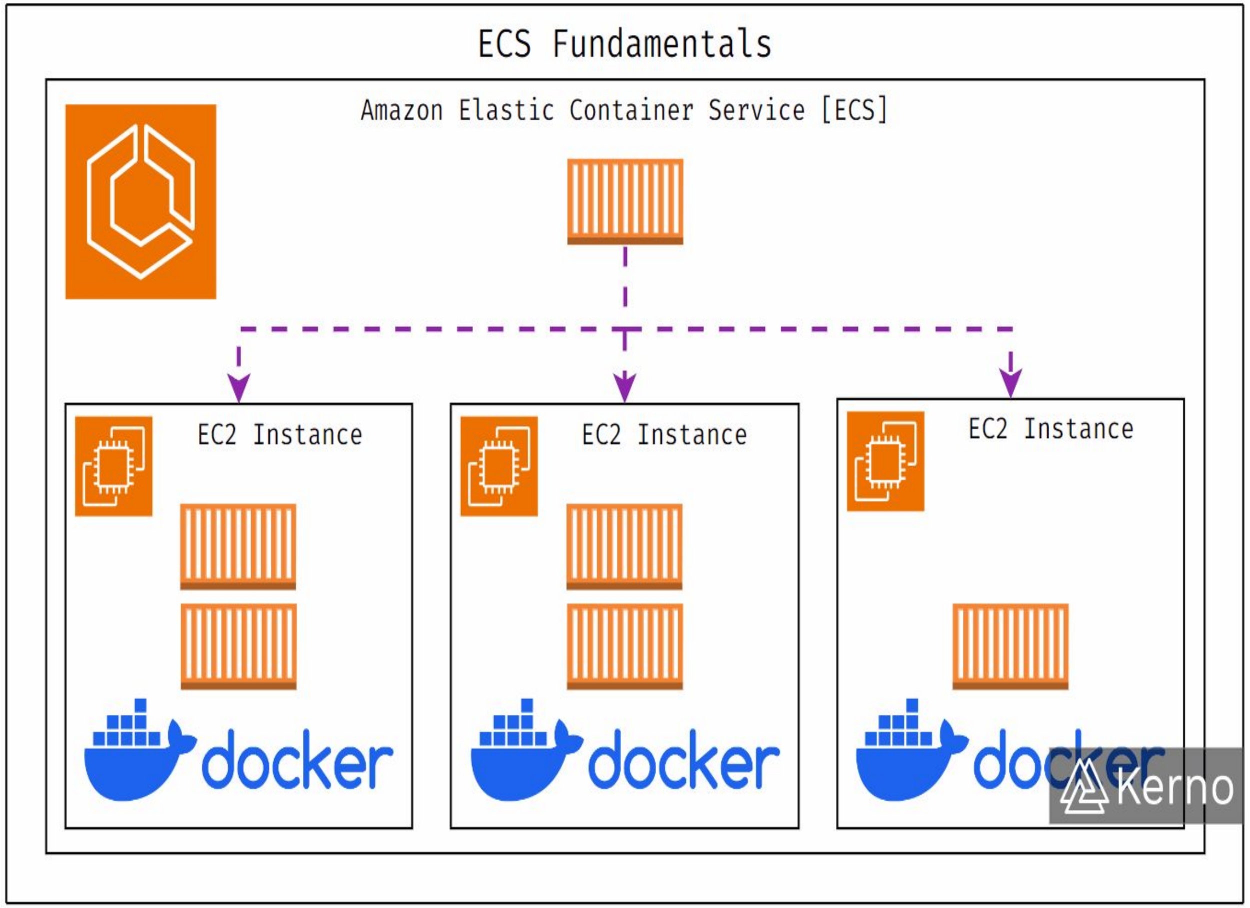
- Documents from a **Jupyter notebook** into a cloud-based message Queue
- **NoSQL table**
- Online scientific document classifier example, showing two levels and subcategories for biology and computer science.
- Document classifier version 1, showing the multiple predictor microservices





Amazon EC2 Container Service

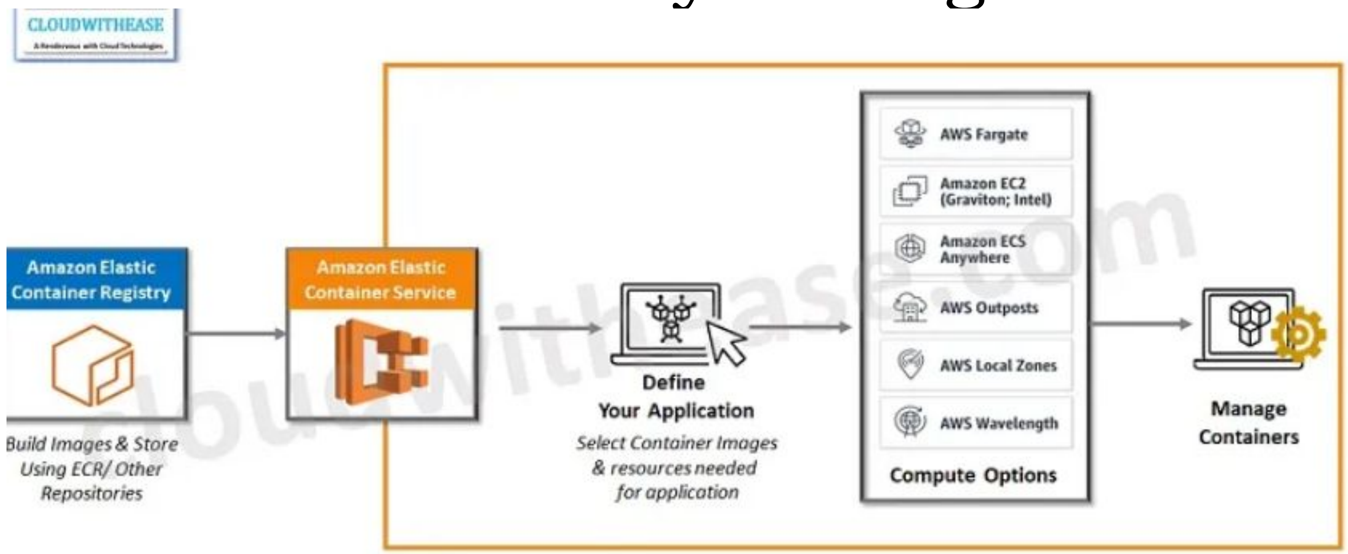
- The **Amazon EC2 Container Service (ECS)** is a system to **manage clusters of servers devoted to launching and managing microservices based on Docker containers**.
- One or more sets of **EC2 instance**s, each a **logical unit called a cluster.**
- One Default cluster can be formed if it requires we can **add more.**
- Task definitions, which specify information about the containers in your application, such as *how many containers are part of your task, what resources they are to use, how they are linked, and which host ports they are to use*.



Amazon Elastic Container Service (ECS), also known as Amazon EC-2 Container Service, is a managed service that allows users to run Docker-based applications packaged as containers across a **cluster of EC2 instances.**

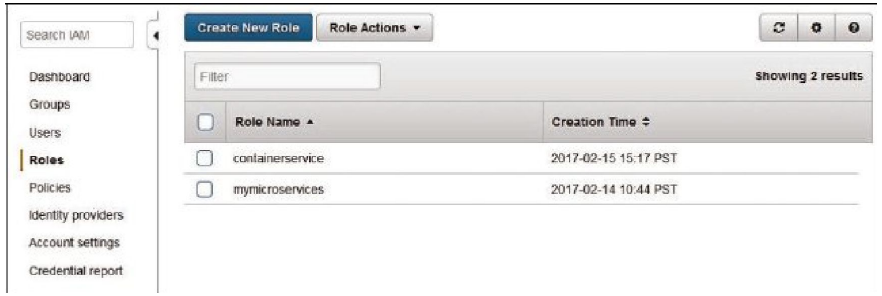
Running simple containers on a single EC-2 instance is simple but running these applications on a cluster of instances and managing the cluster is being administratively heavy process.

- **Amazon-hosted Docker image repositories**.
- Storing your images here may **make them faster to load when needed**, but you can also use the public Docker Hub repository.
- Amazon refers to the **EC2 VM instances in a cluster as container instances.**
- **Amazon Identity and Access Management (IAM)** system to address the identity management issues.



- The IAM link in the **Security subarea of the AWS** management console takes you to the IAM Dashboard.
- Name it container service, and then select the role type.
- You need two roles: **one for the container service** (which actually refers to the VMs in our cluster) and one for the **actual Docker services that we deploy**.
- Scroll down the list of role types and look for Amazon **EC2 Container Service Role and Amazon Container Service Task Role.**
- Select the **Container Service Role** for container service.
- Save this, and now create a second role and call it my microservices

- On the panel on the left is the link **Roles**.
- Select your container **service role, and click Roles**.
- You should now be able to attach **various access policies** to your role.
- The attach policy button exposes a list of over 400 access policies that you can attach.

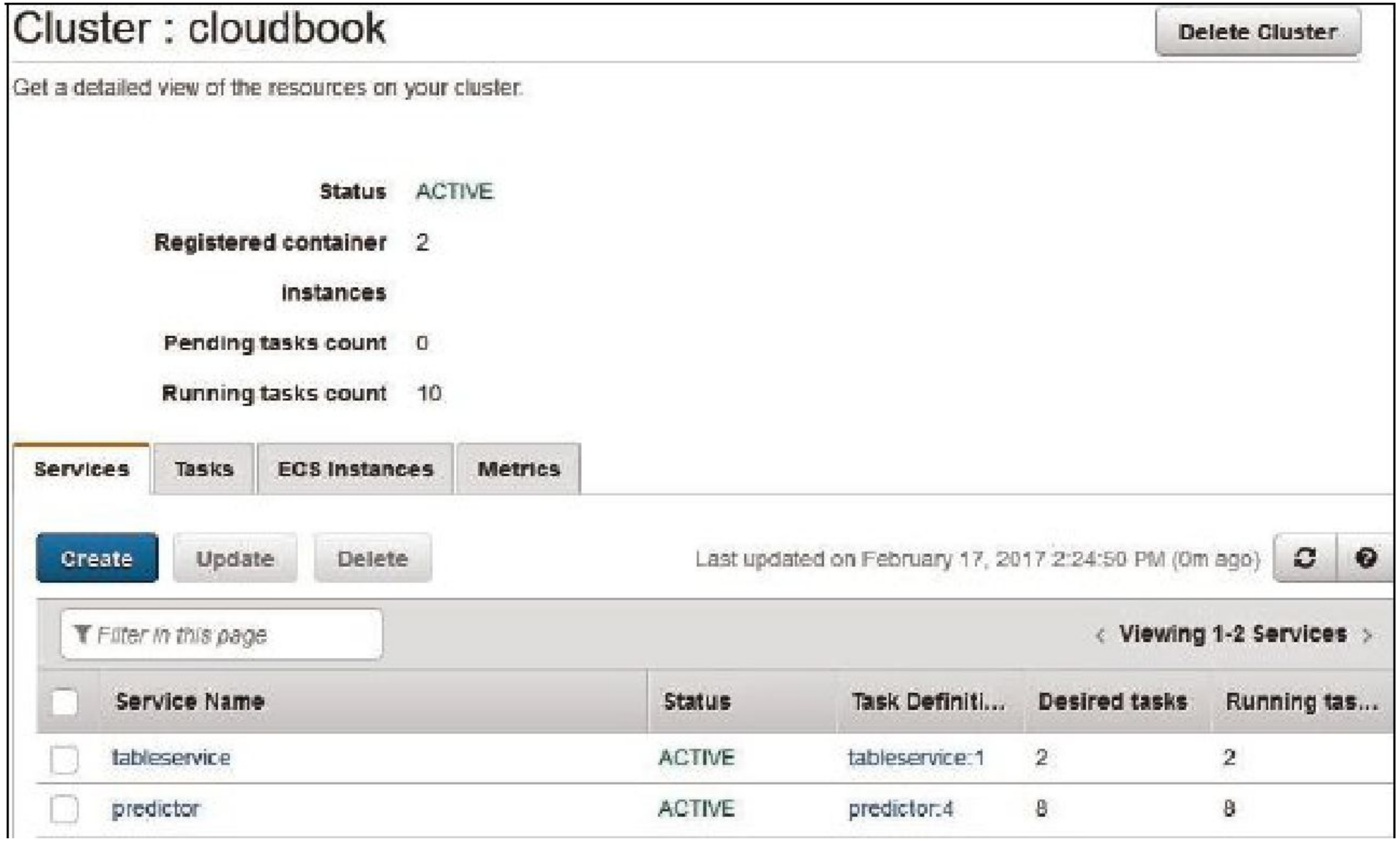


- Add three policies:
  - *AmazonBigtableServiceFullAccess,*
  - *AmazonEC2ContainerServiceforEC2role*
  - *AmazonEC2ContainerServiceRole.*

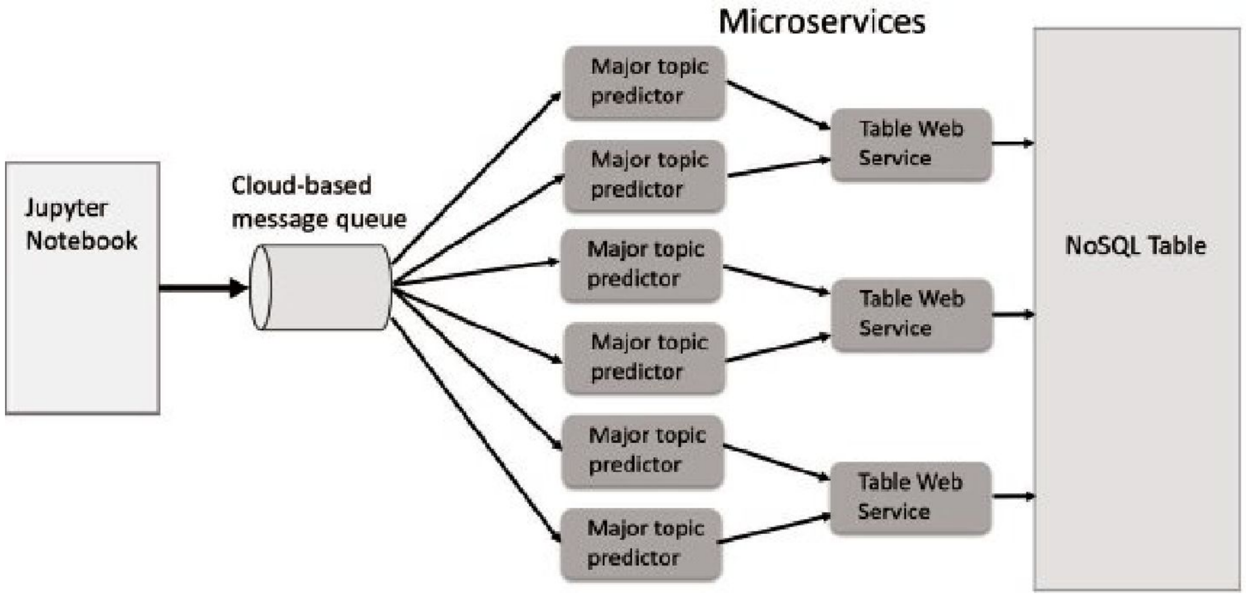
- The mymicroservices role, for the amazon simple queue service (SQS) and dynamodb:
  - Amazonsqsfullaccess and amazondynamodbfullaccess
  - Creating a **cluster** is now easy.
  - From the **amazon ECS console**, simply click create cluster, and then give it a name.
  - Select the **ec2 instance type** you want and provide the number of instances.
  - Container instance **IAM ROLE**, you should see the “containerservice” role.
  - Select this, and select create.
  - The **cluster listed on the cluster console,** with the container instances running

import boto3
client = boto3.client('ecs')
response = client.register_task_definition(
family='predict',
networkMode='bridge',
taskRoleArn =
'arn:aws:iam::01233456789123:role/mymicroservices ',
containerDefinitions=[
{
'name': 'predict',
'image': 'cloudbook/predict',
'cpu': 20,
'memoryReservation': 400,
'essential': True,
},
],
)
	response = client.create_service(
	cluster='cloudbook',
	serviceName='predictor',
	taskDefinition='predict:5',
	desiredCount=8,
	deploymentConfiguration={
	'maximumPercent': 100,
	'minimumHealthyPercent': 50
	}
	)

Eight instances of the predictor and two instances of the table service running



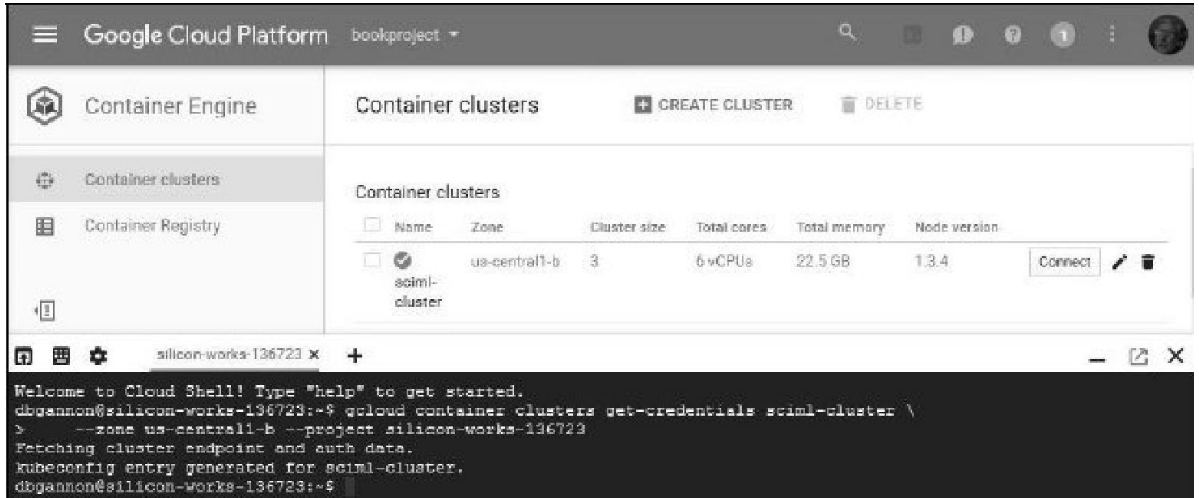
```
queue = sqs.get_queue_by_name(QueueName='bookque')
abstracts, sites, titles = load_docs("path-todocuments",
"sciml_data_arxiv")
for i in range(1330,1430):
queue.send_message(MessageBody='boto3',
MessageAttributes ={
'Title':{ 'StringValue': titles[i],
'DataType': 'String'},
'Source':{ 'StringValue': sites[i],
'DataType': 'String'},
'Abstract':{ 'StringValue': abstracts[i],
'DataType': 'String'}
```



PartitionKey	RowKey	Answer	Date	Predicted	Title
e0bfabe3d880	0	gr-qc	148…	Physics	Superconducting dark eng …
e0bfabe3d880	1	physics.optics	148…	Physics	Directional out-coupling of il …
e0bfabe3d880	2	q-bio.PI	148…	Bio	A guide through a family of p …
e0bfabe3d880	4	math.PR	148…	Math	Critical population and error …
e0bfabe3d880	5	physics.comp	148…	Phys	Coupling all-atom molecular …
e0bfabe3d880	7	hep-th	148…	Pyysics	Nonsingular Cosmology from …

Google's Kubernetes

- This service can be both **installed on a third-party cloud and accessed within the Google Cloud**.
- Creating a Kubernetes cluster on the Google Cloud is easy.
- Google **CLOUD PUB/SUB**, that supports both push and pull subscribers.
- The monitoring for **Kubernetes involves the collection and analysis of metrics, logs, and events**. They provide insights into the health and performance of the systems or applications running on the clusters.



- **Google Kubernetes Engine** (also known as GKE) is a **managed, production-ready environment for running Docker containers in the Google cloud**.
- It permits you to form **multiple-node clusters** whereas conjointly providing access to any or all Kubernetes options.

CONTAINERIZATION

Allows to run and manage containers (K8)

Allows to create and manage containers(DC)

SELF-HEALING

Automatically replaces failed containers with new ones

It relies on third-party tools like Docker Compose or Docker Swarm

LOAD BALANCING

Provides internal load balancing

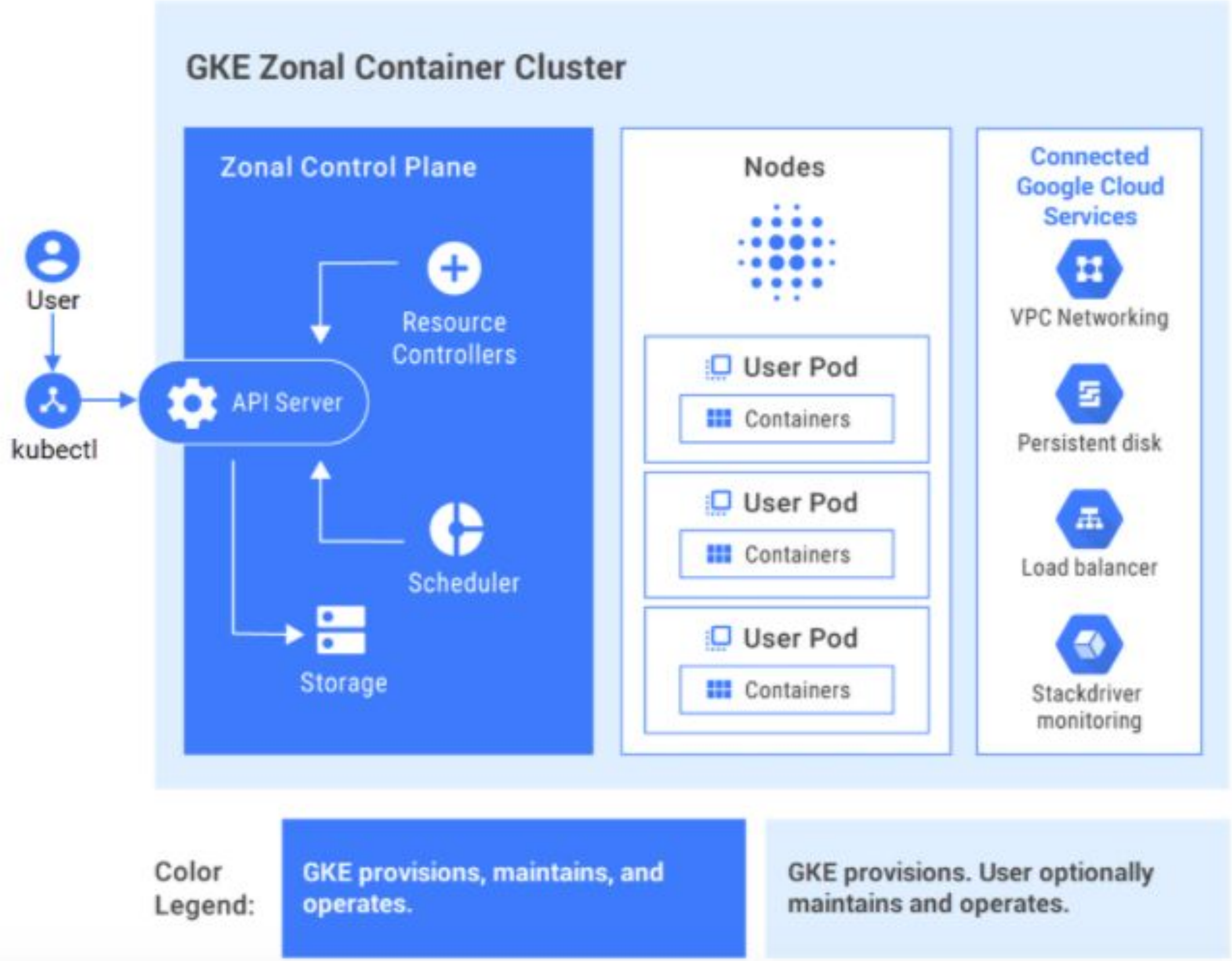
It relies on third-party tools like Docker Swarm

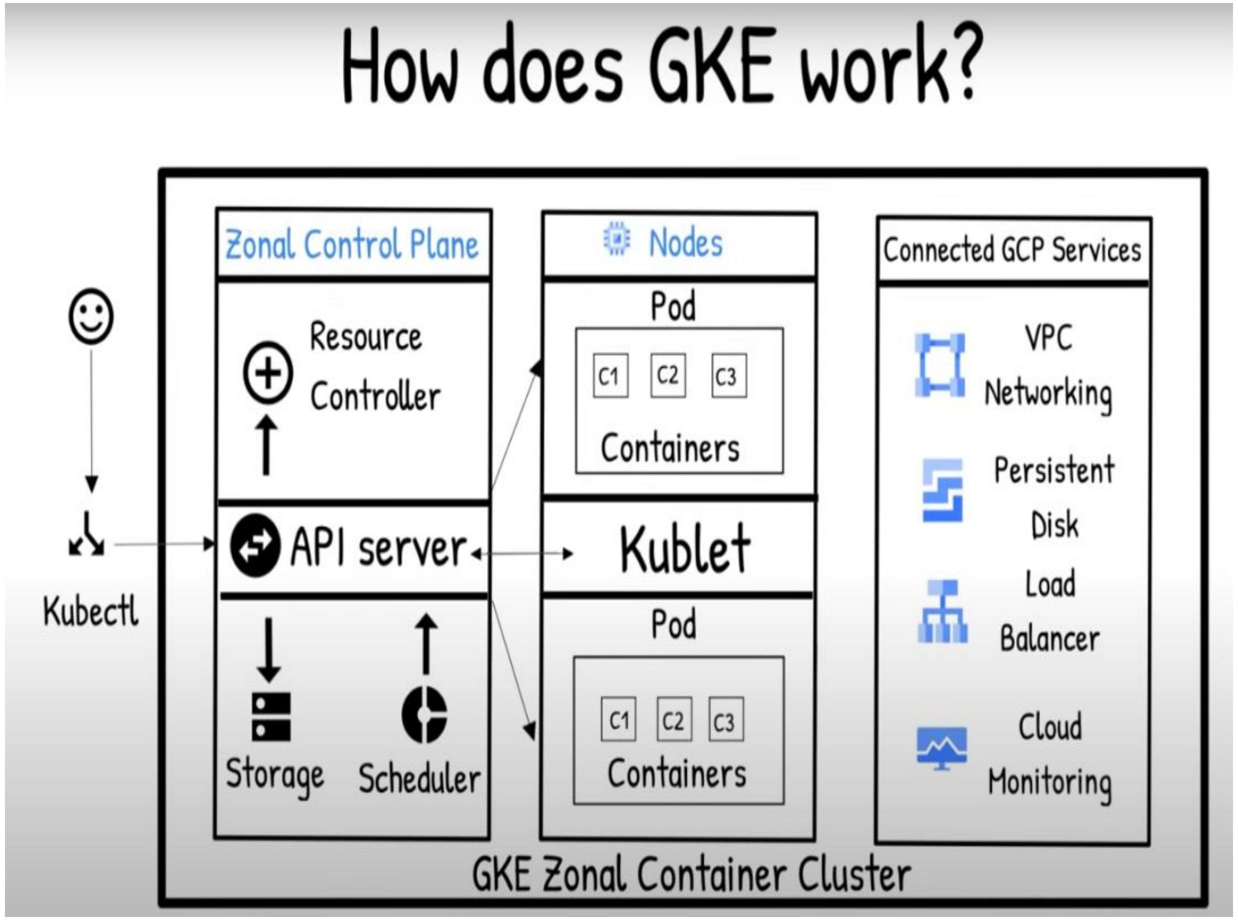
Components of Kubernetes

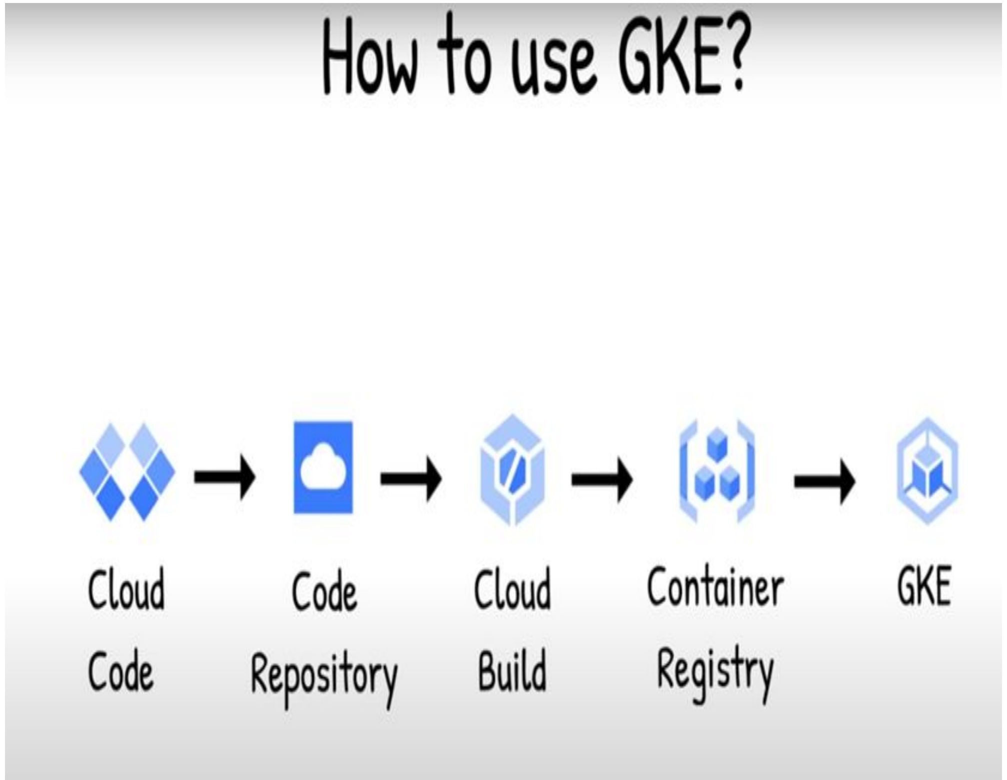
SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

- **NODES**
- Node is the core of any **Kubernetes cluster**.
- It hosts *system Pods, application Pods, and maybe controller Pods*.
- It is important to know if nodes are working as expected.
- **PODS**
- Pods are what you run.
- You need to know for sure what is ***happening on your applications.***
- Cluster-level metrics determine the **scalability, and application-level metrics provide insights**.

- **INGRESS**
- Ingresses **handle traffic towards Pods or Services**.
- Getting statistics **helps with troubleshooting network-related issues.**
- As the door of your applications, it should be working fine all the time with detailed records of who entered.
- **PERSISTENT STORAGE**
- You should make sure the volumes are well-planned and its **resources are utilized correctly**.
- In some cases, the unavailability of storage brings downtime to your applications.


GKE Zonal Container Cluster
Zonal Control Plane
User
kubectl
API Server
Resource Controllers
Scheduler
Storage
Nodes
User Pod
Containers
User Pod
Containers
User Pod
Containers
Connected Google Cloud Services
VPC Networking
Persistent disk
Load balancer
Stackdriver monitoring
Color Legend:
GKE provisions, maintains, and operates.
GKE provisions. User optionally maintains and operates.


How does GKE work?
Zonal Control Plane
Resource
Controller
API server
Storage
Scheduler
Kubectl
Nodes
Pod
C1
C2
C3
Containers
Kublet
Pod
C1
C2
C3
Containers
Connected GCP Services
VPC
Networking
Persistent
Disk
Load
Balancer
Cloud
Monitoring
GKE Zonal Container Cluster


How to use GKE?
Cloud
Code
Code
Repository
Cloud
Build
Container
Registry
GKE

- Deploy an **instance of the open source queue service** rabbitmq rabbitmq.Com on a VM running on jetstream.
- Python package called **CELERY** to communicate with the queue service.
- Celery is a distributed **remote procedure call system** for python programs.
- The celery view of the world is that you have a set of **worker processes running on remote machines** and a client process that invokes functions that are executed on the remote machines.
- **ADVANCED MESSAGE QUEUING PROTOCOL (AMQP)**

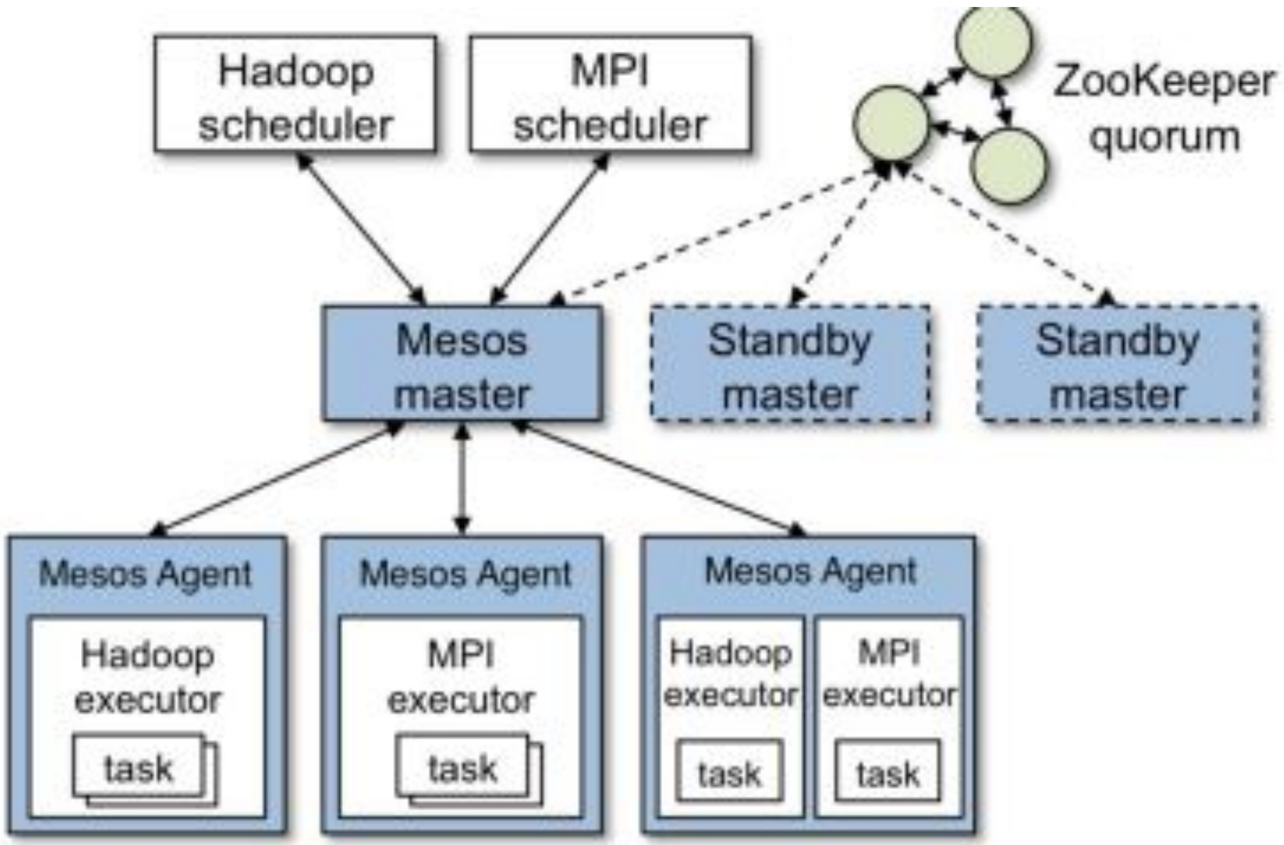
- >celery worker -A predictor -b
- 'amqp://guest@brokerIPaddr
- **from celery import Celery**
- app = Celery('predictor',
- broker='amqp://guest@brokerIPaddr',\
- backend='amqp')
- @app.task
- def predict(statement):
- return ["stub call"]
- res = predict.apply_async(["this is a science
- document..."])
- print(res.get())

Mesos and Mesosphere

- **Managing and running distributed systems**.
- It provides a set of tools and services that make it easy to ***deploy, scale, and manage containerized applications across a cluster of machines.***
- **Mesosphere is based on Apache Mesos**, an open-source cluster manager that provides a distributed resource abstraction layer.

Mesos and Mesosphere

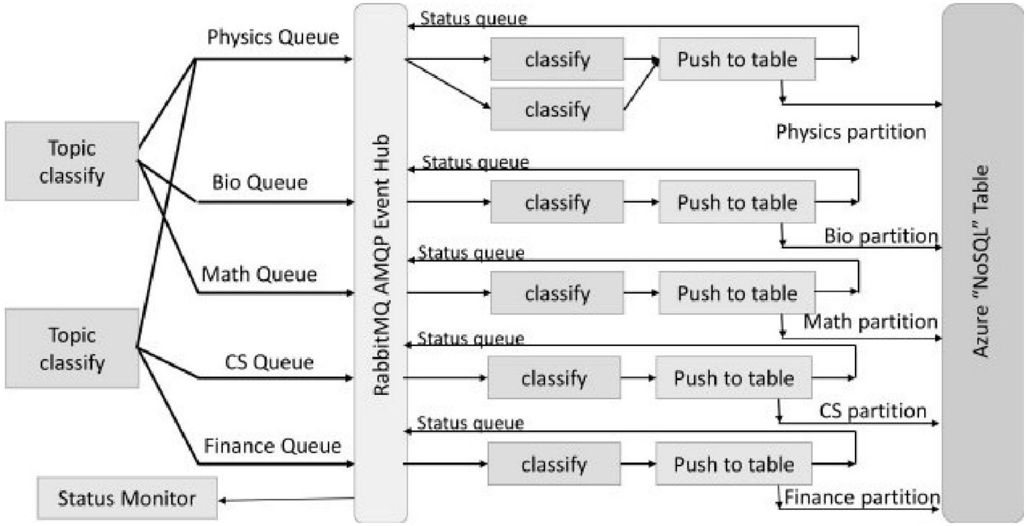
- Mesosphere (from Mesosphere.com) is a **DATA CENTER OPERATING SYSTEM (DCOS)** based on the original Berkeley Mesos system for managing clusters.
- The **Apache Mesos** distributed system **kernel**.
- The **Marathon init system**, which monitors applications and services and, like Amazon ECS and Kubernetes, automatically **heals any failures**.
- **Mesos-DNS**, a service discovery utility.
- **ZooKeeper**, a **high-performance coordination service** to manage the installed DCOS services.

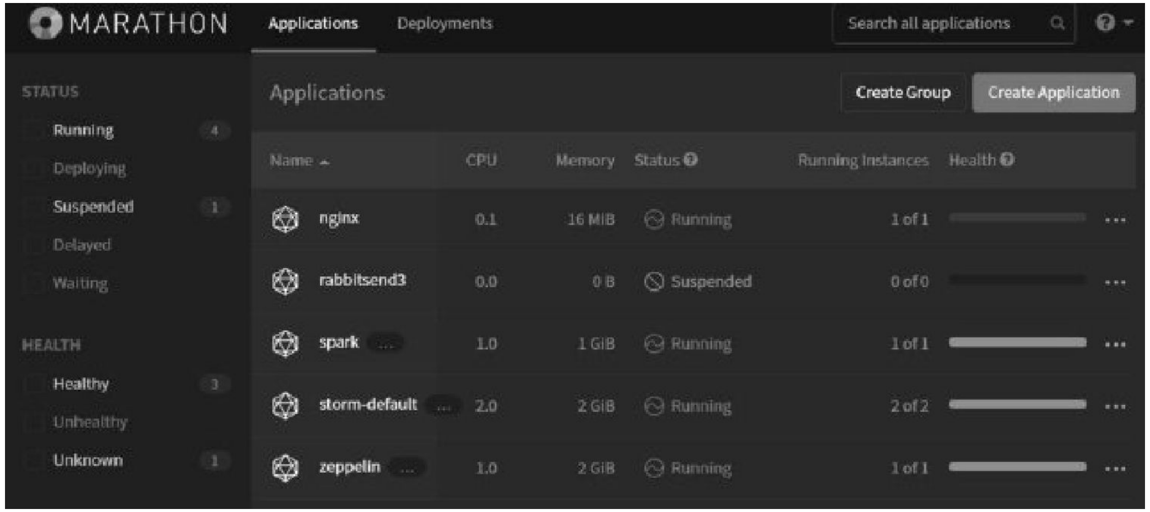

Hadoop scheduler
MPI scheduler
ZooKeeper quorum
Mesos master
Standby master
Standby master
Mesos Agent
Hadoop executor
task
Mesos Agent
MPI executor
task
Mesos Agent
Hadoop executor
MPI executor
task
task

- Mesosphere is deployed, it has **A MASTER NODE**, a **backup master, and a set of workers that run the service containers.**
- **Azure supports the Mesosphere** components listed above as well as another container management service called **DOCKER SWARM**.
- Azure also provides a set of **DCOS command** line tools.

```
> dcos marathon app list
 ID              MEM    CPUS   TASKS   HEALTH   DEPLOYMENT   CONTAINER   CMD
 /nginx           16    0.1    1/1     ---      ---          DOCKER      None
 /rabbitsend3    512    0.1    0/0     ---      ---          DOCKER      None
 /spark         1024     1     1/1     1/1      ---          DOCKER      /sbin/init.sh
 /storm-default 1024     1     2/2     2/2      ---          DOCKER      ./bin/run-
                                                                         with-marathon.sh
 /zeppelin      2048     1     1/1     1/1      ---          DOCKER      sed ...
```

- Mesosphere also provides excellent interactive service management consoles.
- When you bring up Mesos on Azure through the Azure **Container Services, the console presents a view of your service health, current CPU and memory allocations**, and current failure rate. (**Kind of MASHUP ARCH)**



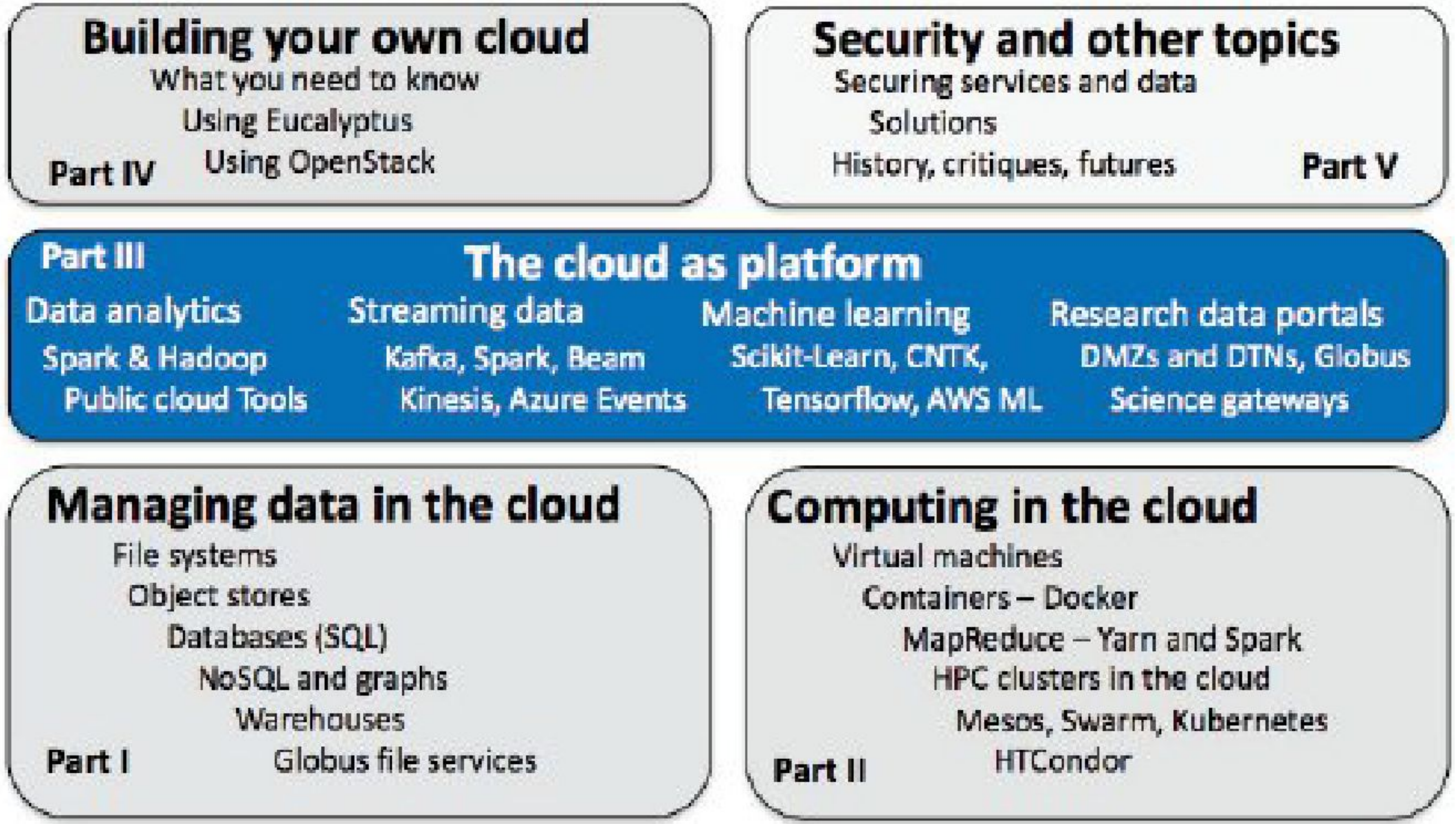


HTCondor

- HTCondor research.cs.wisc.edu/htcondor **high-throughput computing system** is a particularly mature technology for scientific computing in the cloud.
- **Globus Genomics system** uses HTCondor to schedule large numbers of bioinformatics pipelines on the **Amazon cloud**
- **GeoDeepDive geodeepdive**.org, part of the NSF EarthCube project, is an infrastructure for text and data mining that uses HTCondor for **large analyses of massive text collections**

- **PEGASUS** is a workflow system for **managing large scientific computations on top of HTCondor**.
- **HEPCLOUD PROJECT** used HTCondor to process data from a high energy physics experiment

UNIT III


Building your own cloud
What you need to know
Using Eucalyptus
Part IV
Using OpenStack
Security and other topics
Securing services and data
Solutions
History, critiques, futures
Part V
Part III
The cloud as platform
Data analytics
Spark & Hadoop
Public cloud Tools
Streaming data
Kafka, Spark, Beam
Kinesis, Azure Events
Machine learning
Scikit-Learn, CNTK,
Tensorflow, AWS ML
Research data portals
DMZs and DTNs, Globus
Science gateways
Managing data in the cloud
File systems
Object stores
Databases (SQL)
NoSQL and graphs
Warehouses
Part I
Globus file services
Computing in the cloud
Virtual machines
Containers – Docker
MapReduce – Yarn and Spark
HPC clusters in the cloud
Mesos, Swarm, Kubernetes
Part II
HTCondor

The Cloud as Platform

Key concepts	Description
DATA ANALYTICS	• As implemented with the **HADOOP AND YARN TOOLS** including spark. • **AMAZON ELASTIC MAPREDUCE** and **AZURE HDINSIGHT** and **GOOGLE'S CLOUD DATALAB**. • Data warehouse tools such as **AZURE DATA LAKE AND AMAZON ATHENA**.
STREAMING DATA	• Services, which have become a **FULLY INTEGRATED PART** of the public cloud landscape. • **AMAZON KINESIS** and its analytics tools, along with azure event hubs and stream analytics, are easily used and powerful
MACHINE LEARNING	• Services, which **combine open source libraries** and interactive cloud-based development environments to provide exciting new capabilities. • **Deep learning is revolutionizing** the field because of the availability of extremely large data collections and powerful computing platforms
GLOBUS PLATFORM SERVICES	• Which provide **identity, group, and research data management capabilities** that simplify the development of applications and systems that integrate • **People and data at disparate locations**, such as research data management portals.

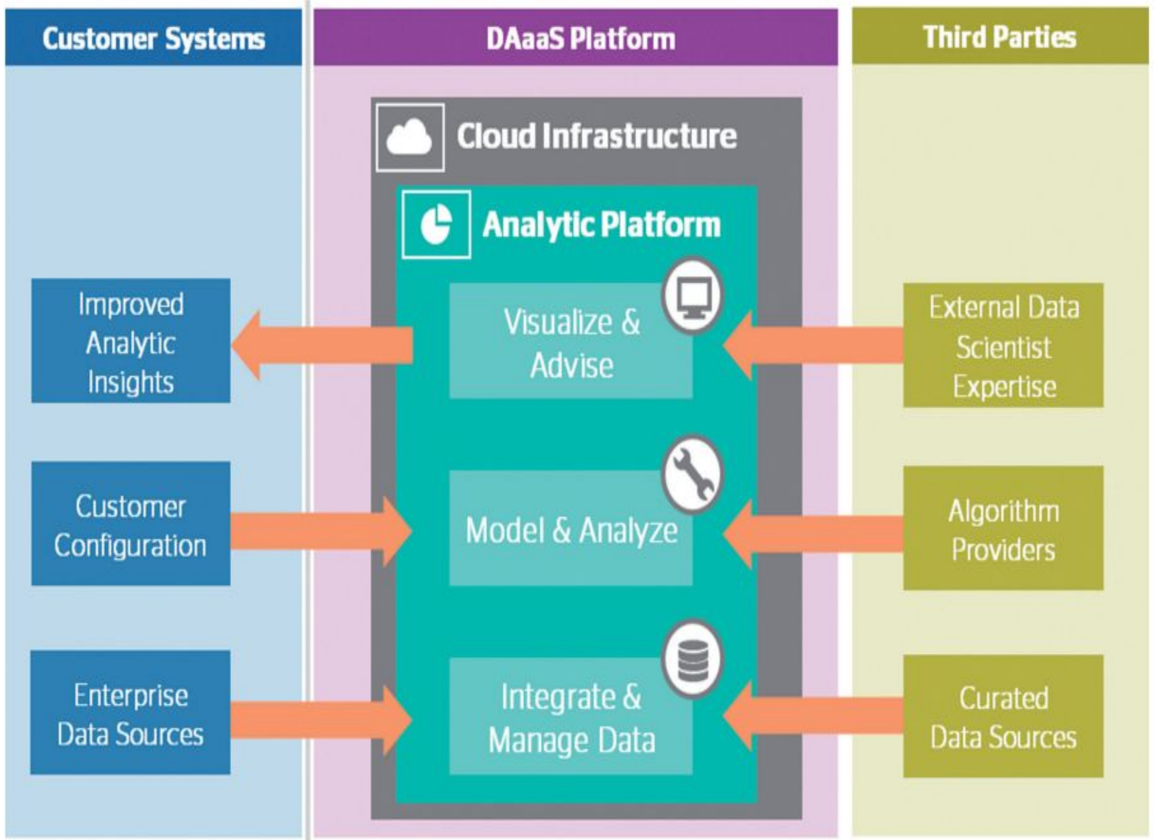
Data Analytics in the Cloud

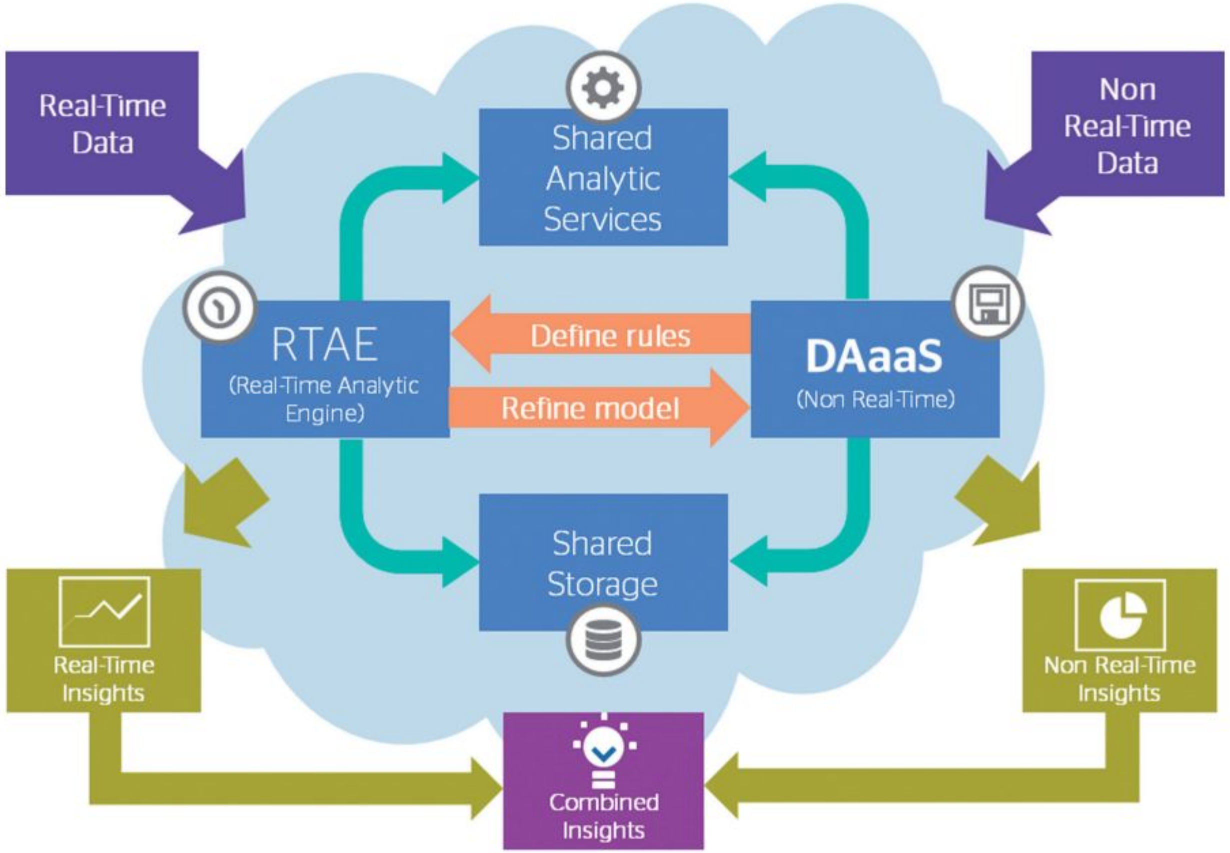
- Cloud analytics is the process of storing and analyzing data in the cloud and using it to extract actionable business insights.
- Similar to on-premises data analytics, cloud analytics algorithms are applied to large data collections.
- To **identify patterns, predict future outcomes and produce other information useful to business decision makers**.

Table1. Comparison of Cloud Based Big Data Enterprise Solution Frameworks

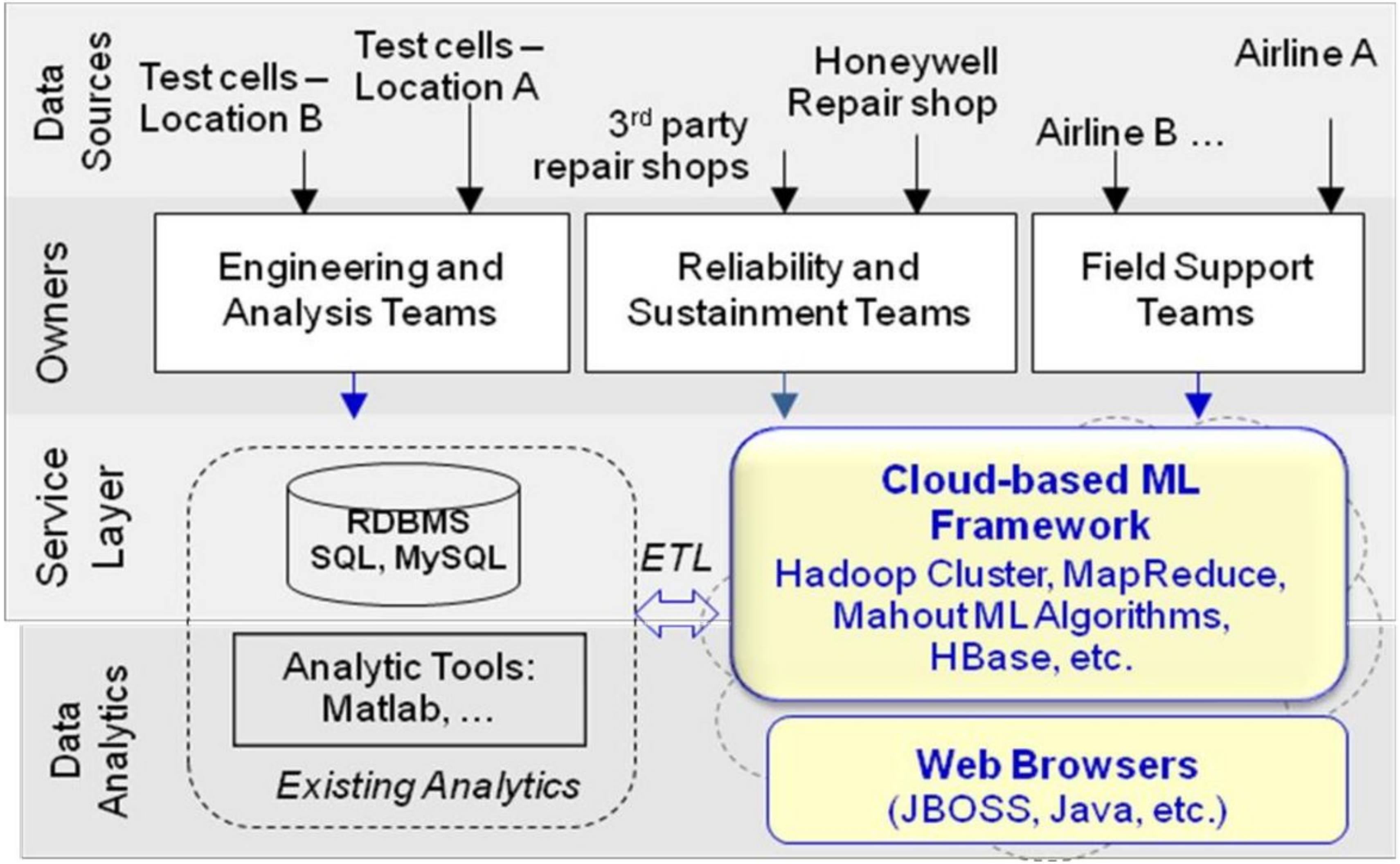
S No	Framework Type and Features	AWS	GCP	IBM Cloud	MS Azure
1.	Big Data Analytics	Amazon ElasticSearch Service	Google Cloud Dataproc	IBM Analytics Engine	Azure HDInsight
	Mode of Software	Open-Source	Open-Source	Open-Source	Open-Source
	Types of Data	Structured, semi-structured and unstructured	Structured, semi-structured and unstructured	Unstructured	Unstructured
	Data Sources	Amazon S3, Amazon Kinesis Firehose, and Amazon DynamoDB	Google Bigtable, Google Cloud Storage, and Google BigQuery	IBM Cloud Object Storage	Blob Storage
	Supported Operating System	CentOS, Ubuntu, and Amazon Linux	Debian 8	CentOS 7	Ubuntu 14, Ubuntu 16, and Windows Server 2012 R2
	Applications	Logs analytics, real-time applications monitoring, and clickstream analytics	Batch processing, querying, streaming, and machine learning	Data analytics, enterprise solution for various Big data problems, and analytics applications development and deployment	Stream and Batch data analytics
	Service Integration	Yes	Yes	Yes	Yes
	Deployment	Zonal	Zonal	Regional	Regional

Data Analytics as a Service (DAaaS)





rmatted across the three nodes.



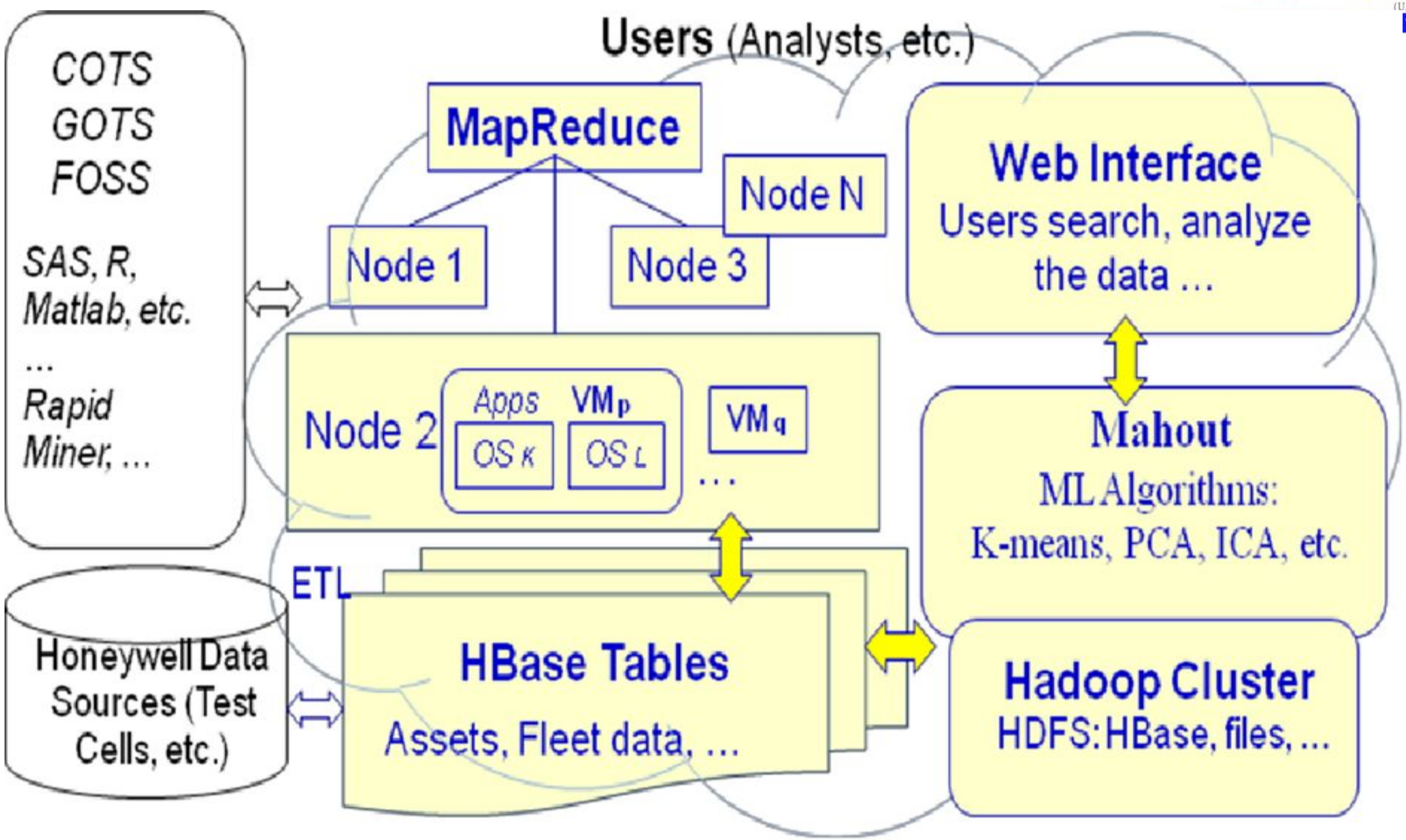
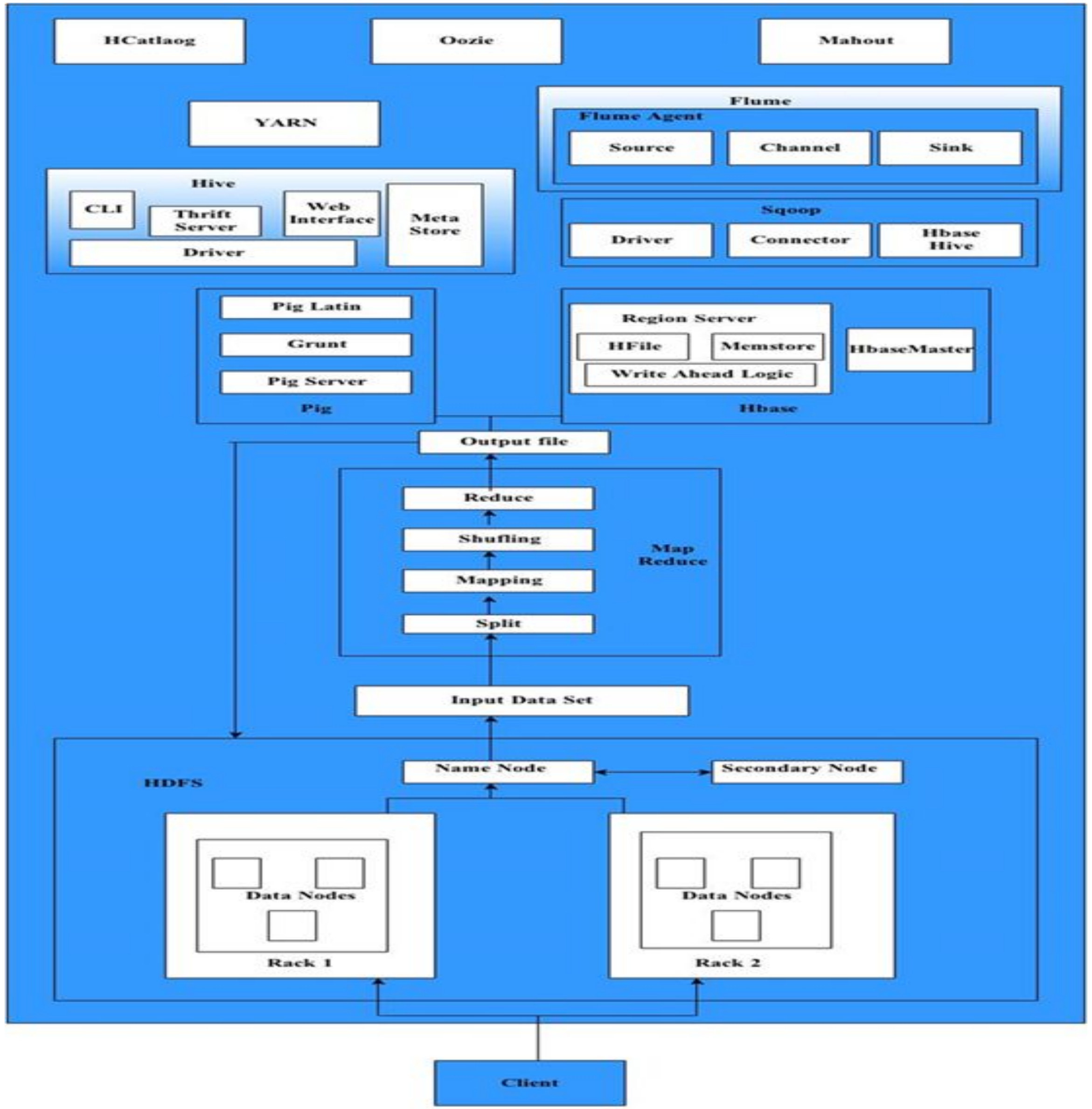


Figure 1.
Architecture and Components of our Cloud-based ML Framework.

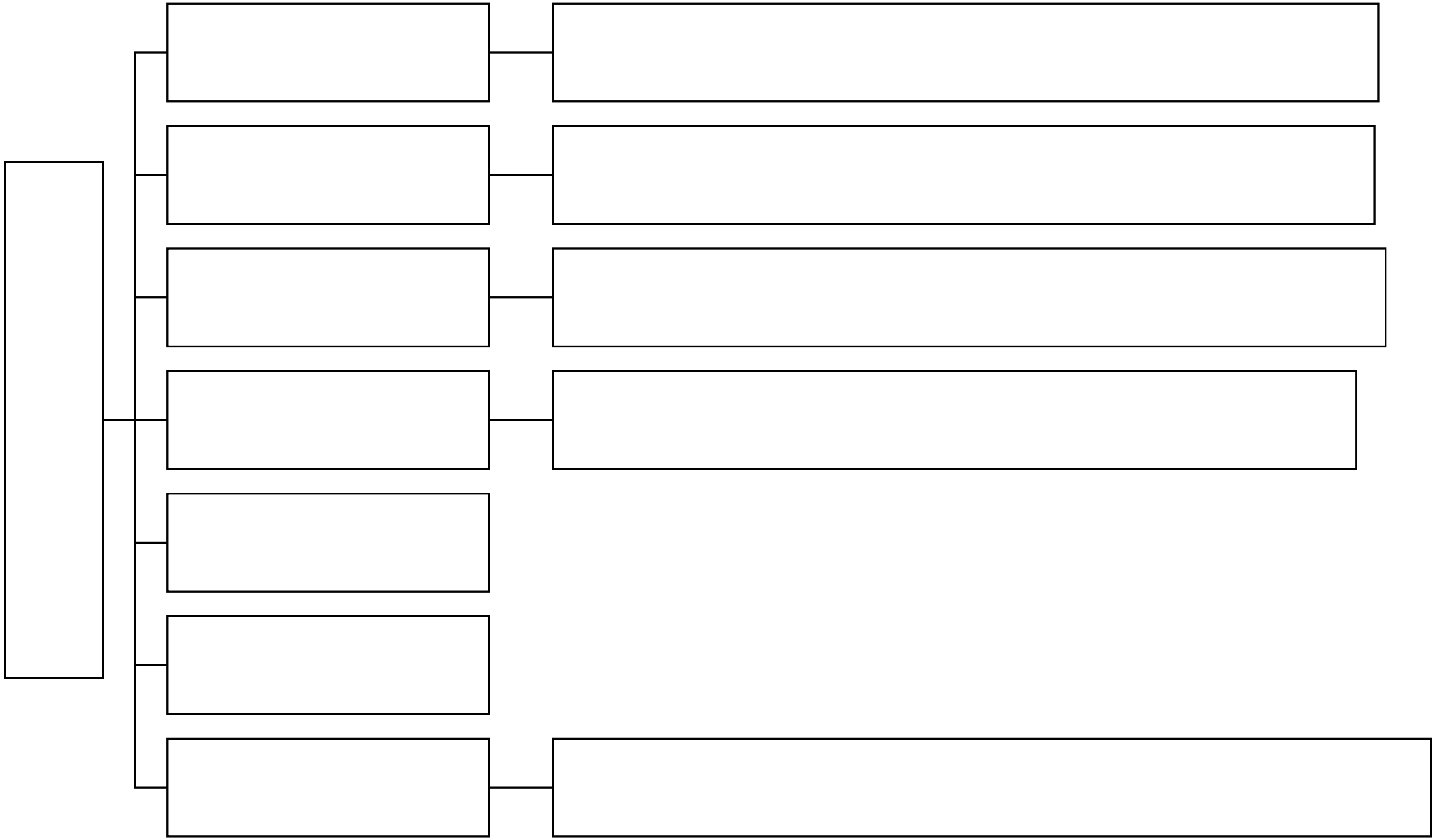
Hadoop and YARN



Hadoop Distributed File System (HDFS)

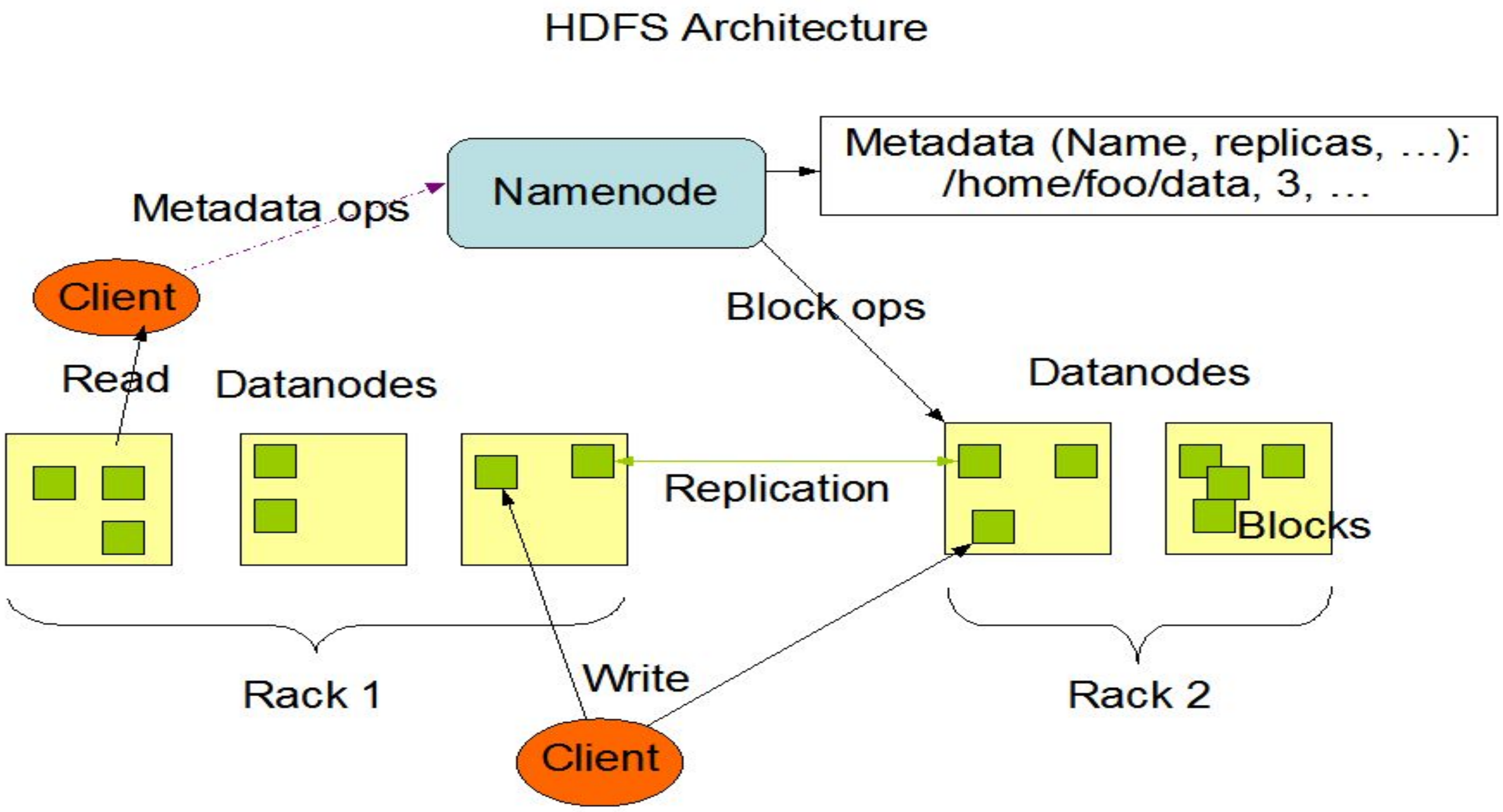
- A distributed filesystem is a filesystem that allows us to store data across multiple machines **or nodes in a cluster and allows multiple users to access data**.
- One of the biggest challenges in **DFS is to tolerate node failure without suffering data loss**.
- HDFS can store data of any size generated from any source in any formats, either structured or unstructured.

- HDFS stores data across the commodity hardware due to which there is **no need for high-end machines for storing big data.**
- Thus provides economical storage for storing big data.
- HDFS follows the most efficient data processing pattern that is **Write-Once-Read-Many-Times pattern**.
- A dataset generated from various sources are **copied, and then the various analysis is performed on that dataset over time.**
- So, it is best for **BATCH PROCESSING**.



HDFS Architecture

- Hadoop DFS follows **master-slave** architecture



Components in HDFS

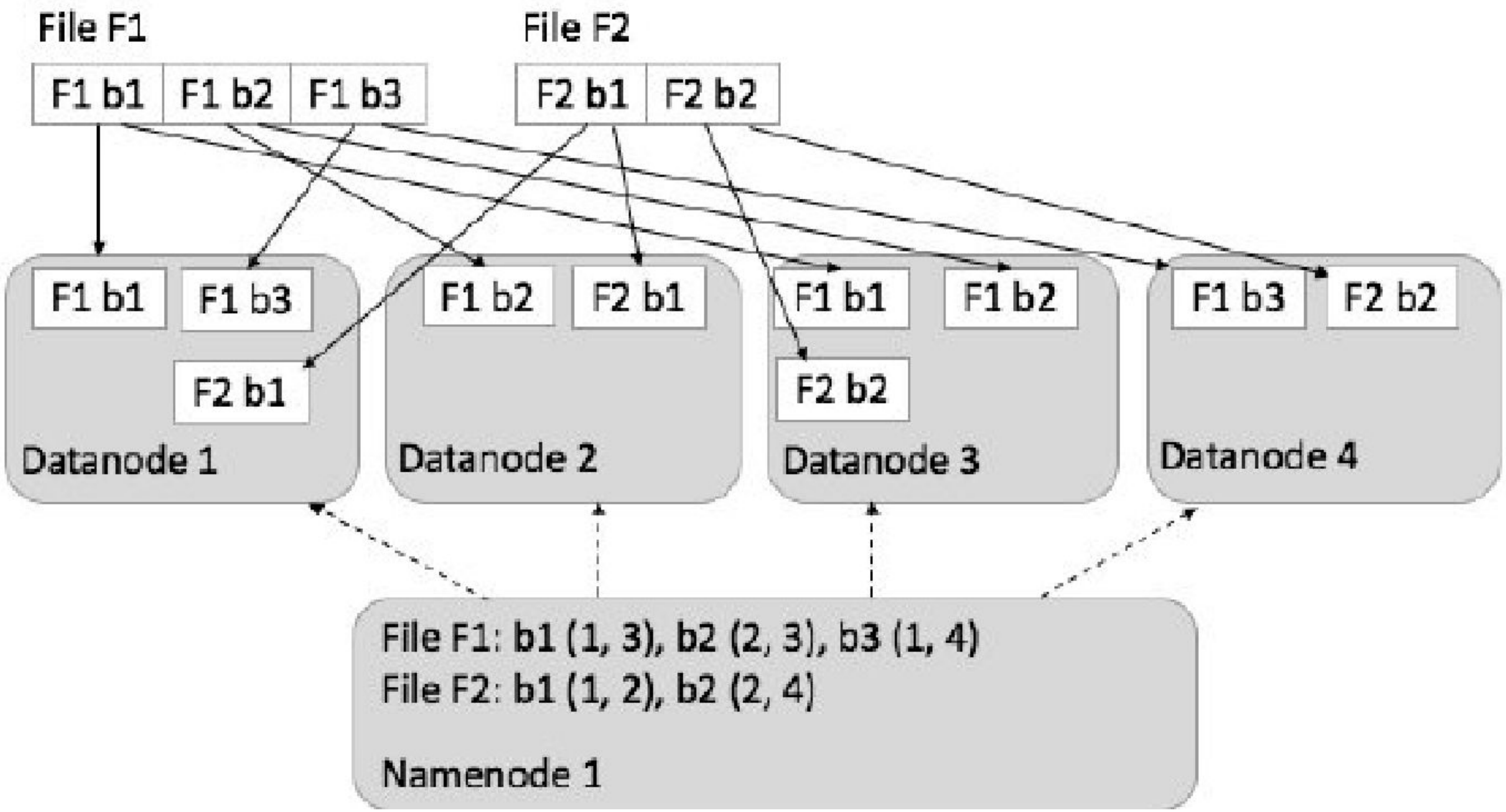
Name	Features
HDFS Master	• Master in HDFS is the centerpiece of **Hadoop HDFS**. • They are the high-end machines that **store metadata** related to all the files stored in HDFS. • It manages and maintains the filesystem namespace and provides instructions to the slave nodes. • **NameNode** is the master node in Hadoop HDFS
HDFS Slave	• Slave Nodes are responsible for storing the actual business data. • They are the normal configuration machines (commodity hardware) that stores and processes the datasets upon instruction from the master node. • The **DataNodes** are the slave nodes in Hadoop HDFS
HDFS NameNode	• **NameNode** is the master node. • It manages filesystem namespace operations like opening/closing, renaming files, and directories. • NameNode maps data blocks to DataNodes and records each change made to the filesystem namespace.
HDFS DataNode	• DataNodes are the slave nodes that handle **read/write requests from HDFS clients**. • DataNodes **creates, deletes, and replicates data blocks as per the instructions from the governing name node**.

Components in Master Node

Name	Features
IMAGE	• The **Image in hdfs** is used for keeping the entire namespace in RAM. • The information of the blocks are collected in the list, this **list with inode defines the metadata.** • The name of the metadata is known as **IMAGE**
JOURNAL	• It is also known as **EDIT LOG**, the modification log of the image is stored by using the ***master node and this is known as journal***. • The process carried in the journal is ,if a client perform the write operation in the file system, the journal record the information or the process in the edit log. • The master node contains some in memory, **which updates the alteration carried in the edit log file.**
CHECKPOINT	• The checkpoint is also known **as fsimage (file system image).** • The fsimage is stored by the master node in the local file system; this process is known as checkpoint. • The location of the block replication is dynamic, where they can be changed over time

Components in Data Node

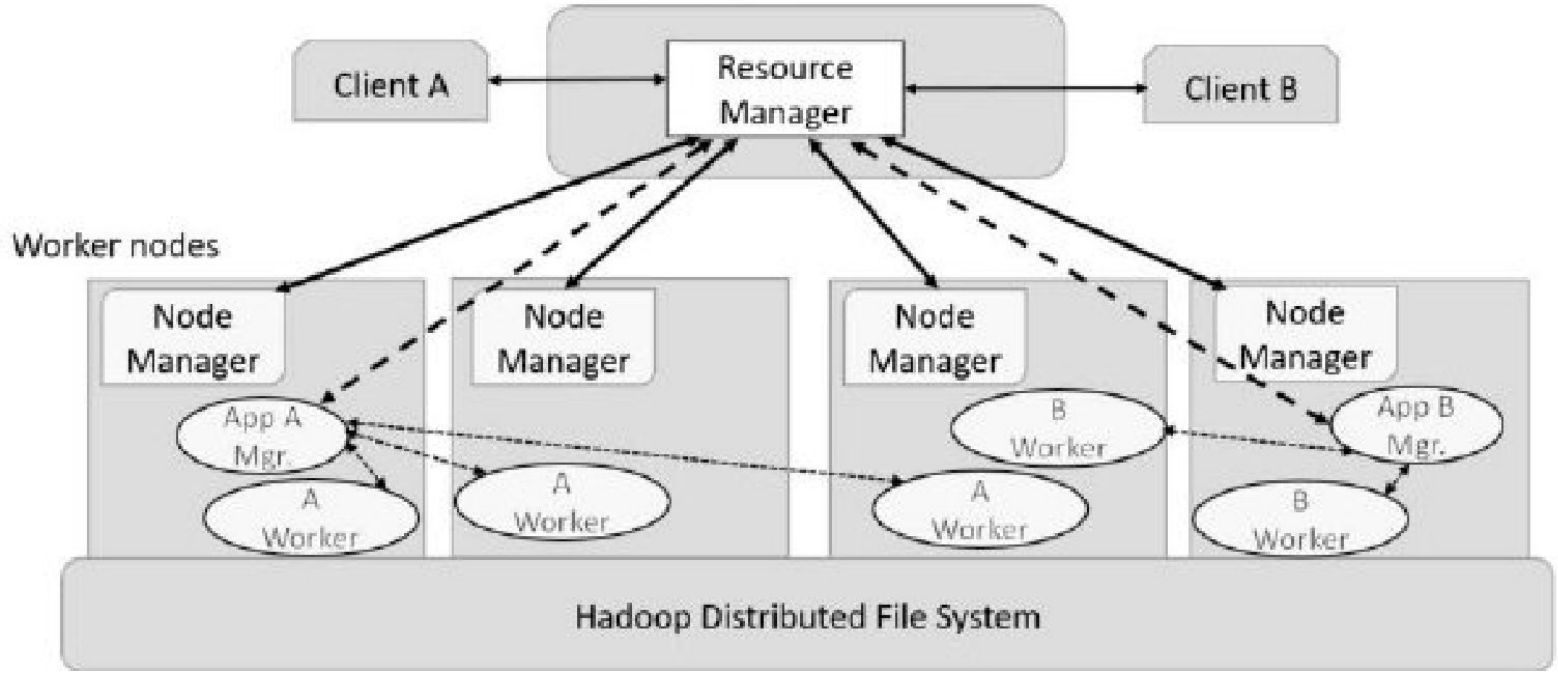
Name	Features
Files	• In hdfs the **data nodes is consider as the slave nodes**, here the slave nodes are used for storing **the replication of each block** and it is represented by two files in the local native file system. • The first file contains the **information about the data** • The second file is used for **making records about the block's** metadata. • It also includes information about the **checksums for the data and the generation stamp.**
Mechanism	• The **heart beat mechanism** is used for checking the status of the data node, by knowing whether the data node is active or inactive


File F1
F1 b1
F1 b2
F1 b3
File F2
F2 b1
F2 b2
F1 b1
F1 b3
F2 b1
Datanode 1
F1 b2
F2 b1
Datanode 2
F1 b1
F1 b2
F2 b2
Datanode 3
F1 b3
F2 b2
Datanode 4
File F1: b1 (1, 3), b2 (2, 3), b3 (1, 4)
File F2: b1 (1, 2), b2 (2, 4)
Namenode 1

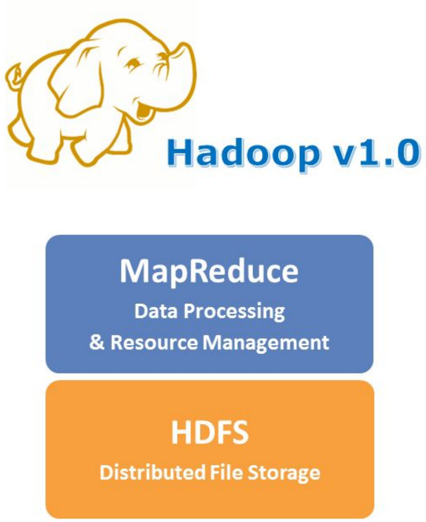
Apache YARN

- Apache YARN (**Yet Another Resource Negotiator**) represents the evolution of the Hadoop ecosystem into a **full distributed job management system**.
- It has a **resource manager and scheduler that communicates with node manager processes in each worker node.**
- Applications connect to the resource manager, which then spins up an application manager for that application instance.

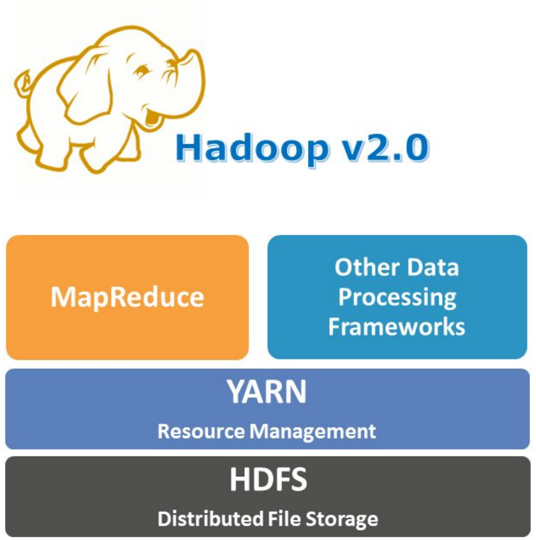
- **Resource Manager**: Runs on a **master daemon** and **manages the resource allocation** in the cluster.
- **Node Manager**: They run on the **slave daemons** and are responsible for the **execution of a task on every single Data Node.**
- **Application Master**: Manages the user **job lifecycle** and resource needs of individual applications. It works along with the Node Manager and **monitors the execution of tasks.**
- **Container**: Package of resources including RAM, CPU, Network, HDD etc on a single node.



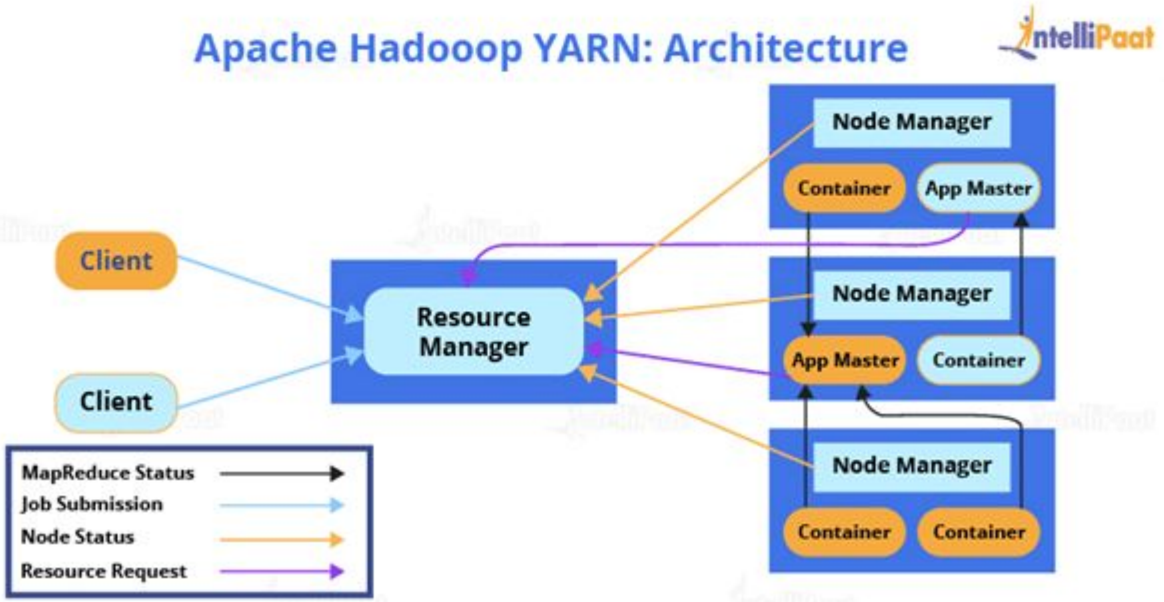
Client A
Resource Manager
Client B
Worker nodes
Node Manager
Node Manager
Node Manager
Node Manager
App A Mgr.
A Worker
A Worker
B Worker
A Worker
App B Mgr.
B Worker
Hadoop Distributed File System



Hadoop v1.0
MapReduce
Data Processing & Resource Management
HDFS
Distributed File Storage



Hadoop v2.0
MapReduce
Other Data Processing Frameworks
YARN
Resource Management
HDFS
Distributed File Storage



Apache Hadooop YARN: Architecture
intelliPaat
Client
Client
Resource Manager
Node Manager
Container
App Master
Node Manager
App Master
Container
Node Manager
Container
Container
MapReduce Status
Job Submission
Node Status
Resource Request

Spark

- Spark's design addresses limitations in the original Hadoop MapReduce computing paradigm.
- In Hadoop's
    - **LINEAR DATAFLOW STRUCTURE**
    - **PROGRAMS READ INPUT DATA FROM DISK**
    - **MAP A FUNCTION ACROSS THE DATA**
    - **REDUCE THE RESULTS OF THE MAP**
    - **STORE REDUCTION RESULTS ON DISK**
    - **The main concern is to maintain speed in processing large datasets in terms of waiting time between queries and waiting time to run the program.**

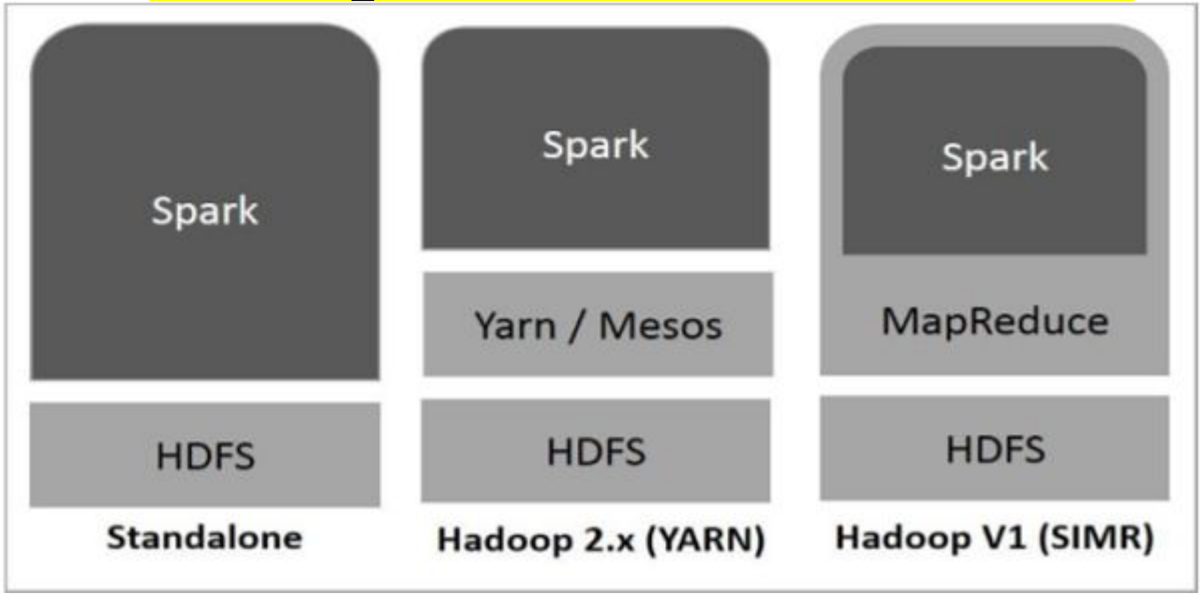
- Spark supports a more **general graph execution model** that allows for iterative MapReduce as well as more **efficient data reuse**.
- Spark is also **interactive and much faster** than pure Hadoop.
- **In-memory cluster computing** that increases the processing speed of an application.

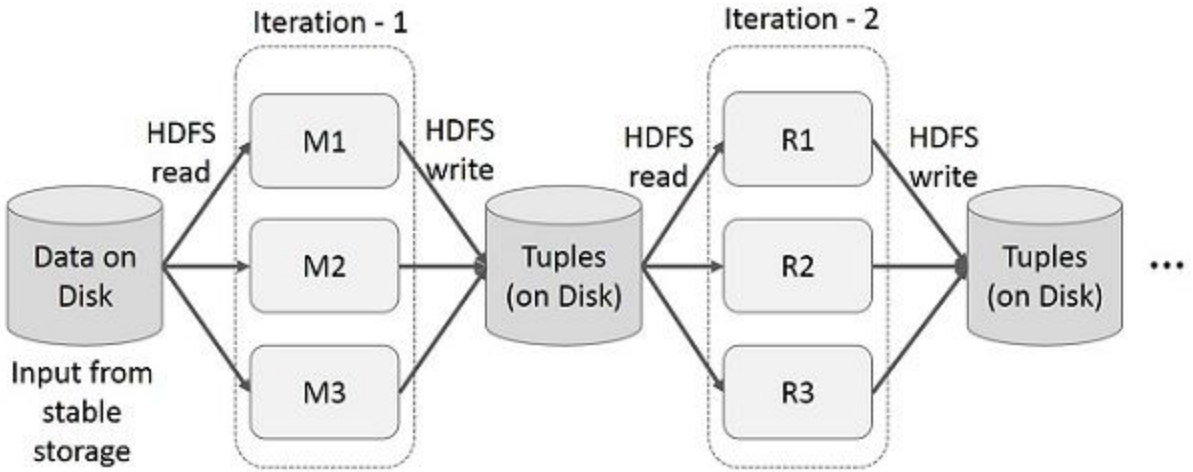
- **Resilient Distributed Dataset (RDD),** a data collection that is distributed across servers and **mapped to disk or memory,** providing a restricted form of distributed shared memory.
- Spark is implemented in **SCALA**, **an interpreted, statically typed object-functional language**.
- Spark has a library of Scala parallel operators, similar to the **Map and Reduce operations used in Hadoop**, that perform transformations on RDDs
- Spark has two types of operations
  - ***Transformations that map RDDS into new RDDS***
  - ***Actions that return values to the main program***

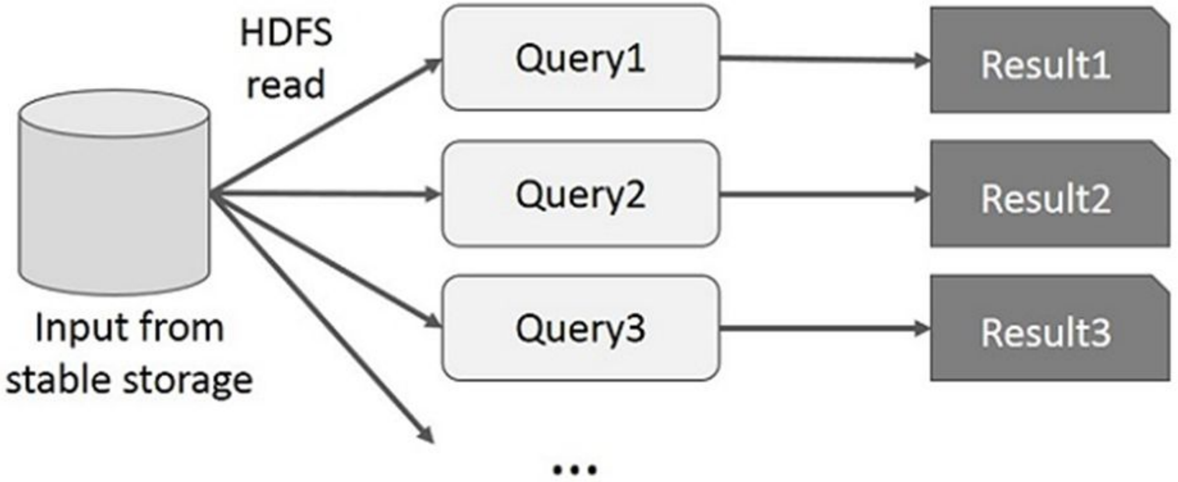
A Simple Spark Program

- **Uses a map operation** to compute for each of **n values** of i, and a reduce operation to sum the results of those computations.
- The program creates a **one-dimensional array** of integers that we then convert to an **RDD partitioned into two pieces.**

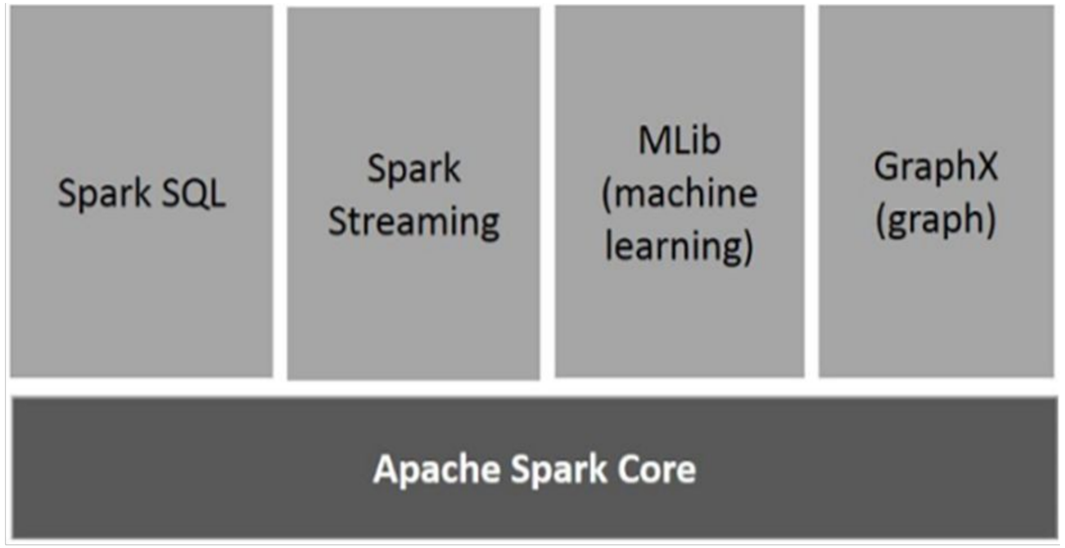
$$\lim_{n->\infty} \sum_{i=1}^{n} \frac{1}{i^2} = \frac{\pi^2}{6}$$

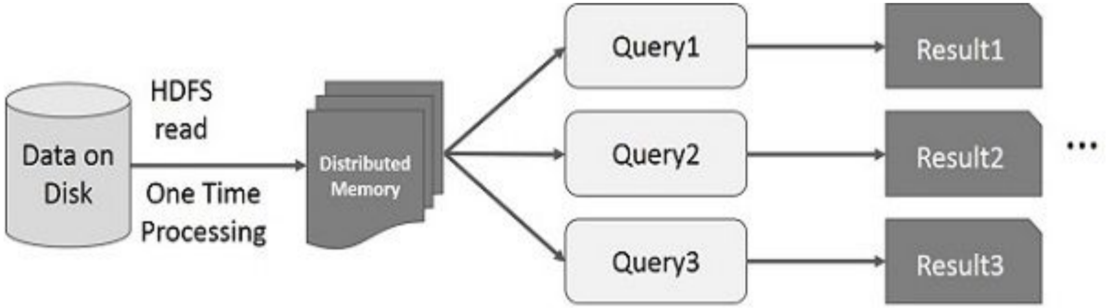


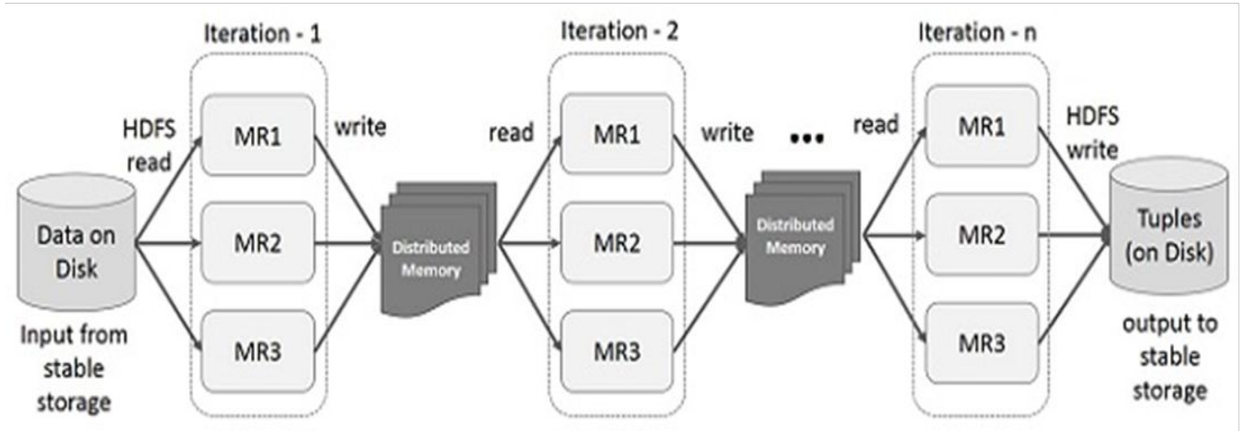




- **Data sharing** is slow in MapReduce due to **replication, serialization**, and **disk IO**.
- Most of the Hadoop applications, they spend more than **90% of the time doing HDFS read-write operations**.

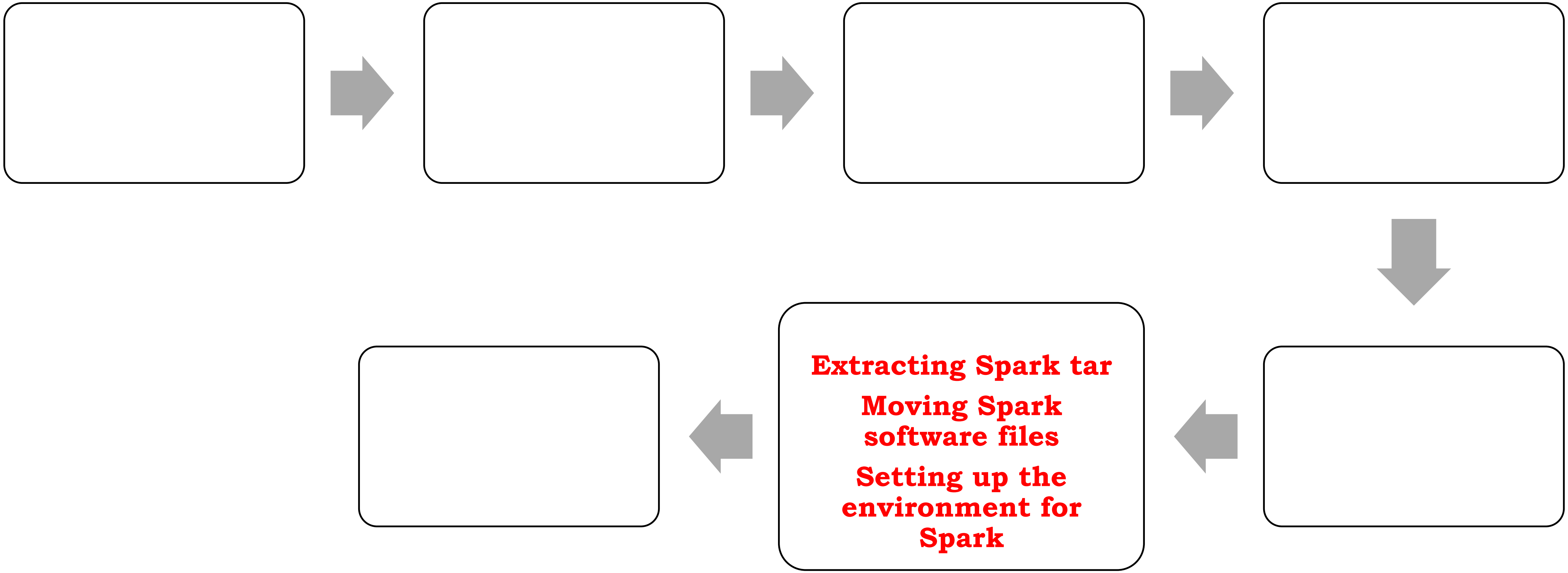

Spark SQL
Spark Streaming
MLib (machine learning)
GraphX (graph)
Apache Spark Core


HDFS read
Data on Disk
One Time Processing
Distributed Memory
Query1
Query2
Query3
Result1
Result2
Result3
...


Iteration - 1
Iteration - 2
Iteration - n
HDFS read
Data on Disk
Input from stable storage
MR1
MR2
MR3
write
Distributed Memory
read
MR1
MR2
MR3
write
...
Distributed Memory
read
MR1
MR2
MR3
HDFS write
Tuples (on Disk)
output to stable storage

Basic Installations in Spark

- Spark is Hadoop’s sub-project. Therefore, it is better to install. Spark into a Linux based system



Computing with Spark

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Map and Reduce is the general term using in parallel processing. Splitting and reducing

Start

- Begin the process using scala or python or java
- **SC (Spark Context):** It is mainly used for connecting to the **cluster, and managing the distributed data and operations**

Read Input File

- **Input:** Text file.
- **Operation**: Use **sc.textFile**("path/to/your/inputfile.txt") to read the file into an RDD. **read a text file and convert it into an RDD (Resilient Distributed Dataset**
- **Output:** RDD of lines from the text file.

Hi how are are you

FlatMap **Transformation**

- **Input**: RDD of lines.
- **Operation**: Apply **flatMap(line => line.split(" "))** to **split each line into words.**
- **Output**: RDD of words.

Hi
how
are
are
you

Computing with Spark

Map
Transformation

- **Input:** RDD of words.
- **Operation:** Apply **map(word => (word, 1))** to create **key-value pairs for each word. And one tells at least 1 occurrence of the word**
- **Output:** RDD of key-value pairs (word, 1).

Hi ,1
How, 1
are ,1
are ,1
You ,1

ReduceByKey
Transformation

- **Input**: RDD of key-value pairs.
- **Operation**: Apply **reduceByKey(_ + _)** to aggregate word counts by key (word). It will check all words and store in list and add the count
- **Output:** RDD of (word, count) pairs

Hi ,1
How, 1
are ,1,1 => 2
You ,1

Save As Text File
Action

- **Input:** RDD of (word, count) pairs.
- **Operation**: **Use saveAsTextFile**("path/to/output/directory") to save the result to a text file.
- **Output:** Directory containing the word count results in text files.

End

- The process is complete

Apache Spark – A sample Programming

- Spark provides an interactive shell – a powerful tool **to analyze data interactively**.
- It is available in either **Scala or Python language**.
- Spark's primary abstraction is a distributed collection of items called a **Resilient Distributed Dataset (RDD).**
- RDDs can be created from **Hadoop Input Formats (such as HDFS files) or by transforming other RDDs.**

Open Spark Shell	$ spark-shell
Create simple RDD	scala> val inputfile = sc.textFile("input.txt") *inputfile: org.apache.spark.rdd.RDD[String] = input.txt MappedRDD[1] at textFile at <console>:12*
RDD Transformations	• **RDD transformations** returns pointer to new RDD and allows you to **create dependencies between RDDs**. • Each RDD in dependency chain (String of Dependencies) has a **function for** calculating its data and has a pointer (dependency) to **its parent RDD**. Therefore, RDD transformation is not a set of data but is a step in a program (might be the only step) telling **Spark how to get data and what to do with it.**

Programming with RDD

Consider a word count example – It counts each word appearing in a document. Consider the following text as an input and is saved as an **input.txt** file in a home directory. **input.txt** – input file "**people are not as beautiful as they look, as they walk or as they talk. they are only as beautiful as they love, as they care as they share**".	
Open Spark-Shell	
Create an RDD *1. Read the input file* *2. Create an RDD* *3. Select absolute path for the input file name*	scala> val inputfile = **sc.textFile("input.txt")** **// read a text file and convert it into an RDD (Resilient Distributed Dataset)**
Execute Word count Transformation 1. Create a flat map for splitting each line into words (**flatMap(line ⇒ line.split(" "))**). 2. read each word as a key with a value **'1'** (<key, value> = <word,1>)using map function (**map(word ⇒ (word, 1)**)). 3. Finally, reduce those keys by adding values of similar keys (**reduceByKey(_+_)**)	scala> val counts = inputfile.flatMap(line => line.split(" ")).map(word => (word, 1)).reduceByKey(_+_); #*Spark* ***flatMap*()** transformation flattens the RDD/DataFrame column after applying the function on every element and returns a new RDD/DataFrame

Programming with RDD

Consider a word count example – It counts each word appearing in a document. Consider the following text as an input and is saved as an **input.txt** file in a home directory. **input.txt** – input file "**people are not as beautiful as they look, as they walk or as they talk. they are only as beautiful as they love, as they care as they share**".	
Create a flat map for splitting each line into words (**flatMap(line ⇒ line.split(" "))**).	This transformation **splits each line into words**. If there are multiple words in a line, **flatMap will return them as individual elements in the resulting RDD**.
map(word => (word, 1))	Here, each word is mapped to a **key-value pair where the key is the word itself and the value is 1.**
Finally, reduce those keys by adding values of similar keys (**reduceByKey(_+_)**)	**reduceByKey(_ + _)**: This transformation **aggregates the values** by key (word), summing up the counts for each word across the entire RDD

Programming with RDD

Current RDD scala> counts.toDebugString	It will show you the description about **current RDD and its dependencies for debugging**
Caching the Transformations scala> counts.cache()	• **Persistence** is an optimization technique in which saves the result of **RDD evaluation**. • Cache() and persist() is that using **cache() the default storage level is memory only** while using persist() we can use various storage levels **Memory and Replicate and Serialize in Memory and Disk**
Applying the Action store all the transformations, results into a text file	scala> counts.saveAsTextFile("output")
Checking the Output	[hadoop@localhost ~]$ cd output/ [hadoop@localhost output]$ ls -1 part-00000 part-00001 _SUCCESS
Output **Part-00000** **Part-00001**	(people,1) (are,2) (not,1) (as,8) (beautiful,2) (they, 7) (look,1) (walk, 1) (or, 1) (talk, 1) (only, 1) (love, 1) (care, 1) (share, 1)

Computing with Spark

jupyter spark-euler (autosaved)

File Edit View Insert Cell Kernel Help | Python [Root]

In [2]:
```
n = 10000000
ar = np.arange(n)
```

In [3]:
```
ar
```

Out[3]:
```
array([      0,       1,       2, ..., 9999997, 9999998, 9999999])
```

grab the spark context convert ar into a Spark RDD with 16 partitions.

In [4]:
```
sc
```

Out[4]:
```
<pyspark.context.SparkContext at 0x100678710>
```

This is where we create our first RDD called 'dat' from our array. In this simple case the RDD will have 2 partitions.

In [5]:
```
numpartitions = 2
dat = sc.parallelize(ar, numpartitions)
```

For fun let's compute $\sum_{i=1}^{n}(1/i^2)$ that Euler in 1735 proved:

$$\lim_{n \to \infty} \sum_{i=1}^{n} \frac{1}{i^2} = \frac{\pi^2}{6}$$

now we will apply a simple map operation to compute $\frac{1}{i^2}$ for each i. Note that we do not actually execute this operation until we resolve the data back to the shell. That will come with the reduce step below.

In [6]:
```
sqrs = dat.map(lambda i: 1.0/(i+1)**2)
```

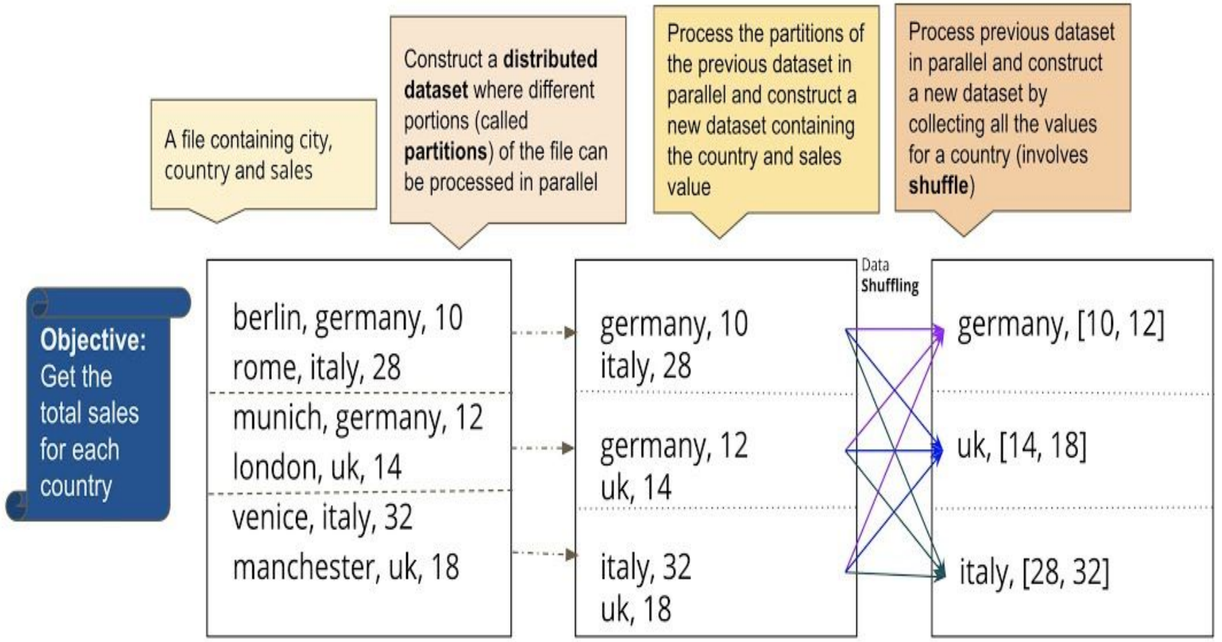
In [7]:
```
t0 = time.time()
x = sqrs.reduce(lambda a,b: a+b)
t1 = time.time()
```

In [8]:
```
print("x = %f"%x)
print("time=%f"%(t1-t0))
```

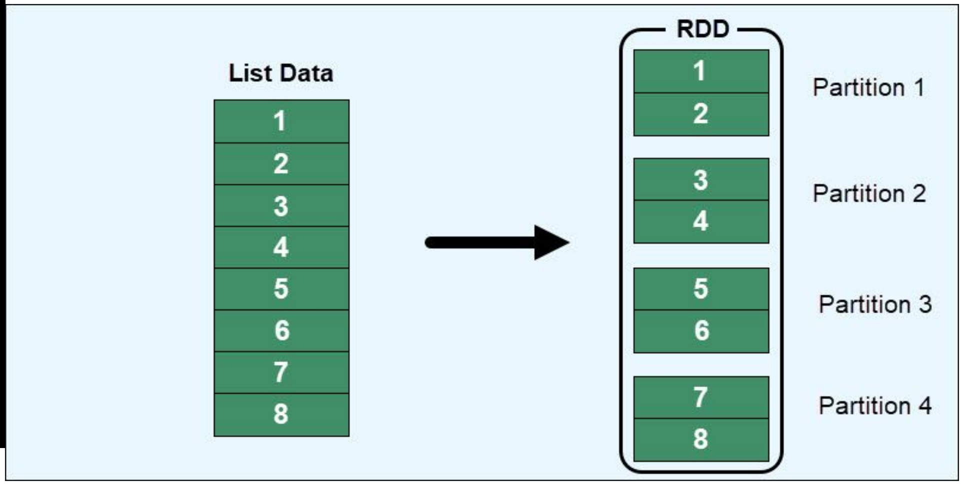
```
x = 1.644934
time=5.151475
```

- In Spark, **the partitions are distributed to the workers**.
- Parallelism is achieved by applying the computational parts of Spark operators on each partition in parallel, using, furthermore, multiple threads per worker.
- For actions, such as a reduce, **most of the work is done on each partition and then across partitions as needed**
- **Resilient Distributed Datasets (RDD)** is a fundamental **data structure of Spark.**
- It is an immutable distributed collection of objects.
- **Immutability** means that once an RDD is created, it cannot be altered. It designed to perform parallel operations across a cluster efficiently.
- **Creates a new RDD rather than modifying the original one.**
- Each dataset in RDD is divided into **LOGICAL PARTITIONS**, which may be computed on **different nodes of the cluster**.
- RDDs can contain any type of Python, Java, or Scala objects, including user-defined classes.

Resilient Distributed Dataset (RDD)



Transformations	Actions
map (func)	reduce(func)
flatMap(func)	collect()
filter(func)	count()
groupByKey()	first()
reduceByKey(func)	take(n)
mapValues(func)	saveAsTextFile(path)
sample(...)	countByKey()
union(other)	foreach(func)
distinct()	...
sortByKey()	
...	



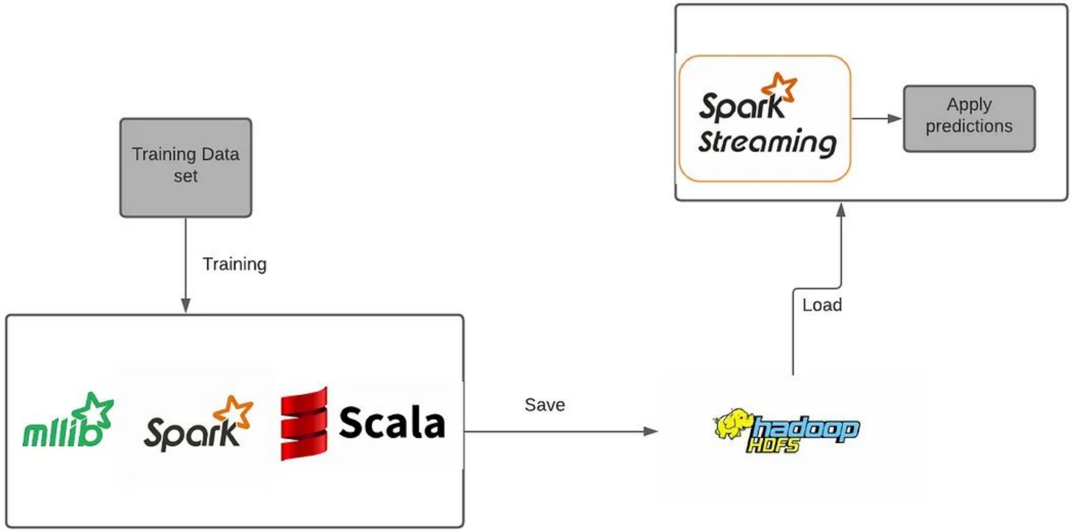
In Spark, transformations (like map, filter, flatMap, etc.) **are not executed immediately when** they are called.

Instead, they are "lazy," meaning that they build up a logical execution plan but do not actually perform any computations until an **action (like count, collect, saveAsTextFile, etc.)** is invoked.

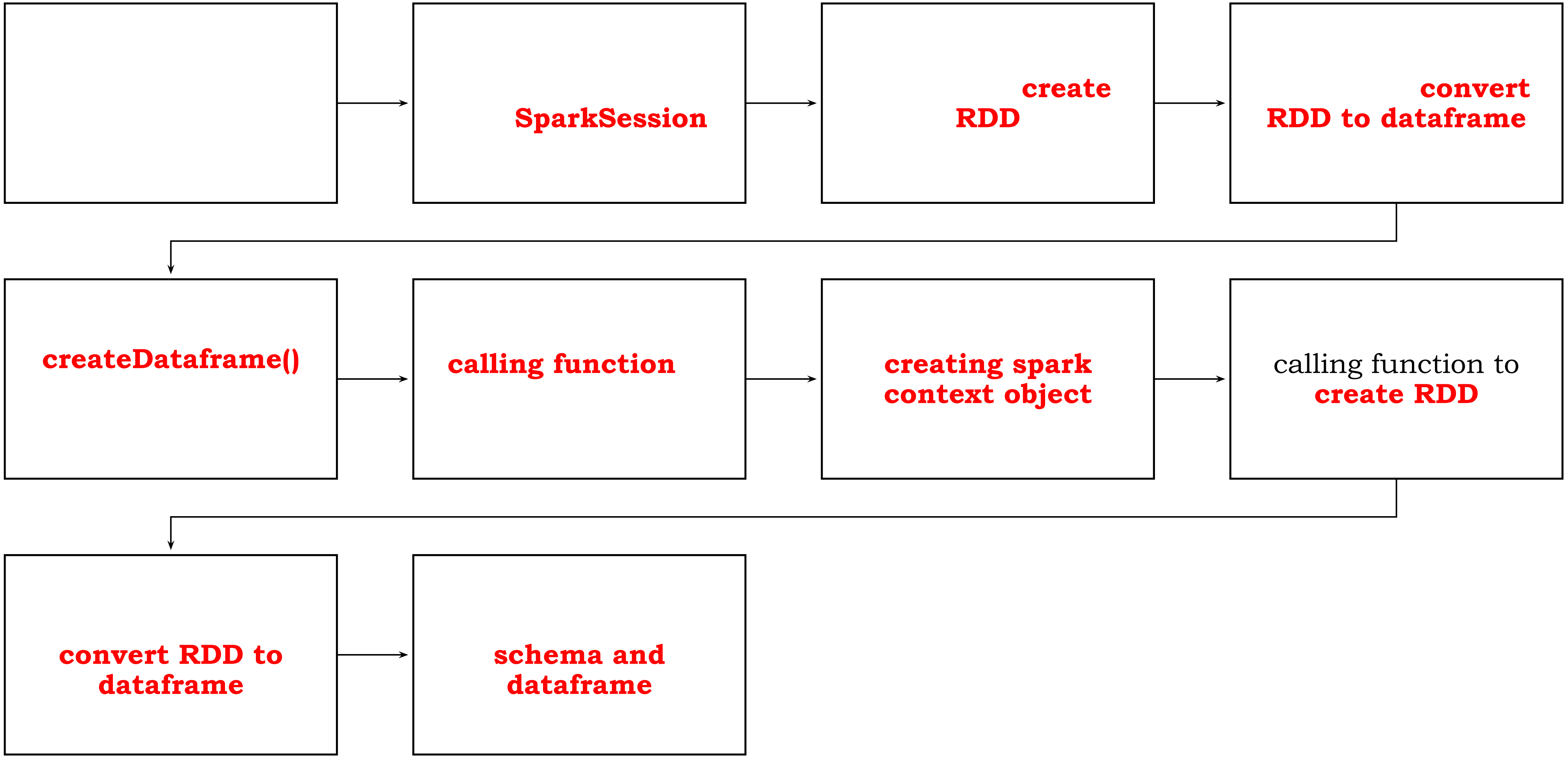
Directed Acyclic Graph, **Execution Triggered by Actions,** Optimizations, to optimize and manage large-scale data processing efficiently

Steps in K means with Spark

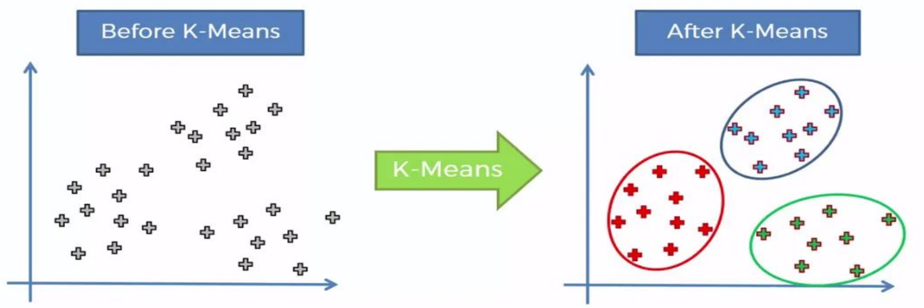
- Train a **k means cluster model using mllib** which is developed as a part of **APACHE SPARK** project.
- Save our trained **model in HDFS**.
- Use this trained model for making predictions.



Converting normal Data frame to RDD



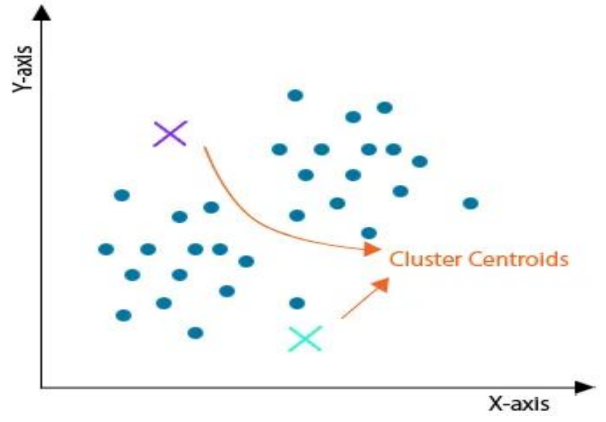
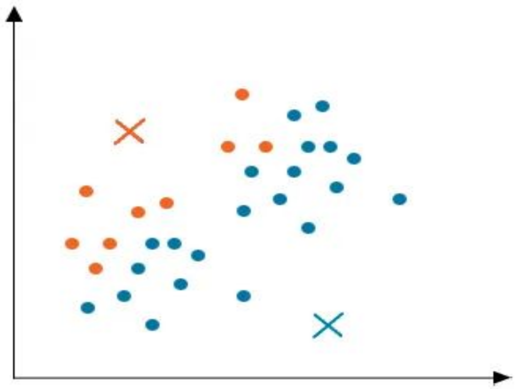
- **CLUSTERING :** Clustering is an ***unsupervised learning*** method, in which we are trying to find the **relation between n observations**.
- In the example we are trying to find the relation between the **three feature** and based on these features all the customers will fall in one of the **two groups**.
- **K MEANS ALGORITHM**
- K means is an **unsupervised clustering algorithm** which the **given data points are partitioned into a cluster** in which the **squared distance between the data points and centroid is minimum**.



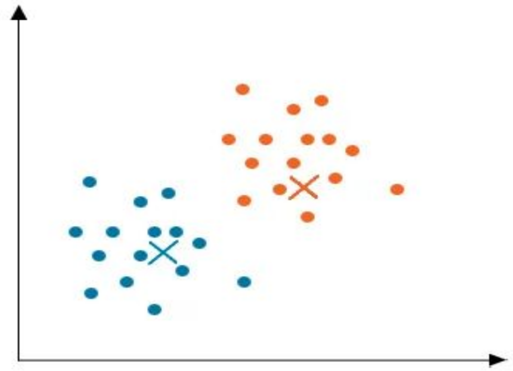
Scenario Based Question

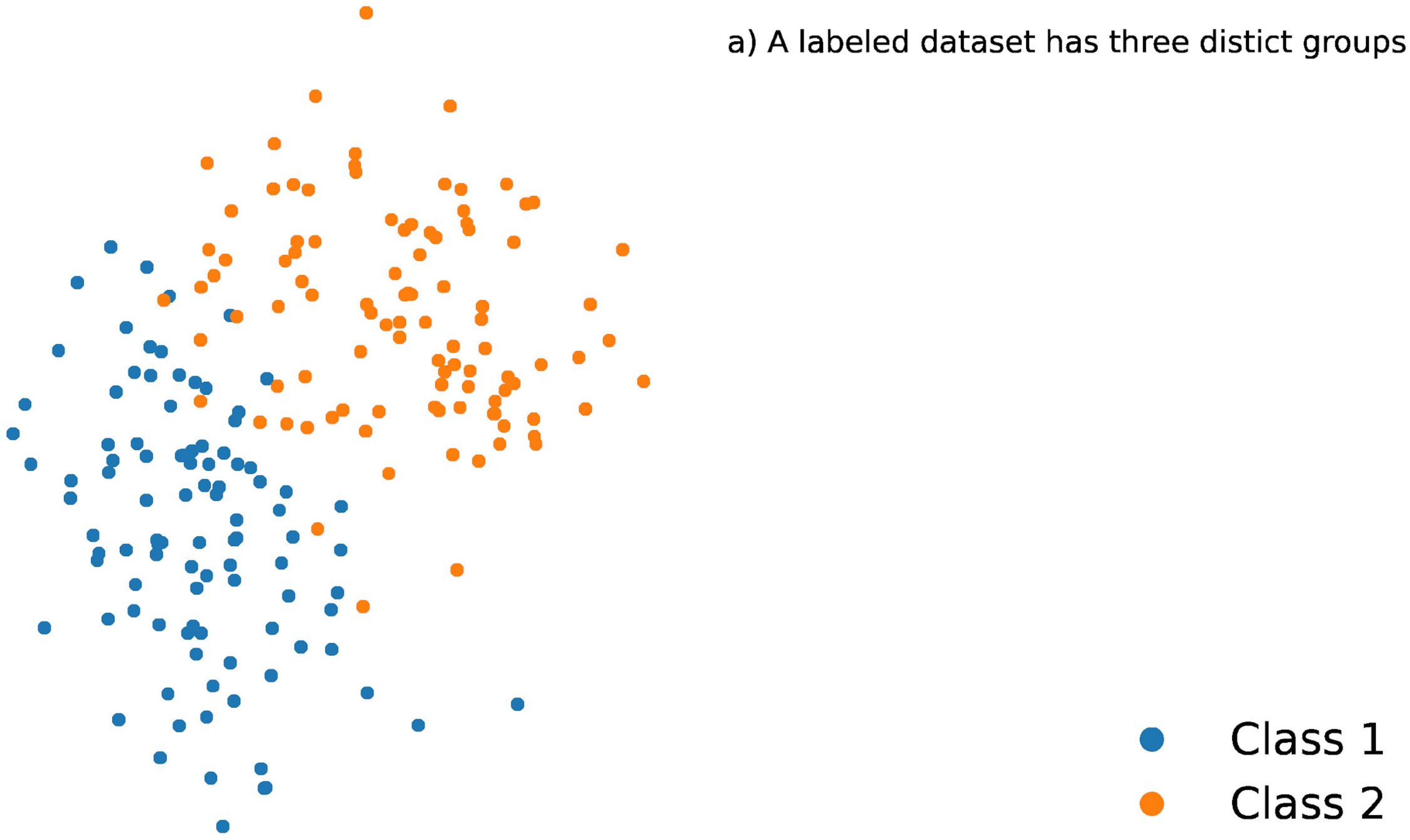
- Consider the following example : For real estate firm we need to categorize their potential customers into **two different classes**. **Class A**: Willing to pay **high amount** for a property. **Class B**: Do not want to pay a **large amount for a property**. To solve the above problem (Categorize all customers into two groups)we were able to gather the following data.
- Annual income of family members.
- Size of the house.
- Distance form urban area.
- Depending on the above features (data gathered) all customers will fall into one of the two class A or B.

Steps in K means clustering

Initialization	Initialize any **random points as the centroid of the cluster**	
Cluster Assignment	After initializing centroid Assign data points in a way such that the **distance between the centroid and the data points are minimum.**	

Steps in K means clustering

Moving centroid	• Now the centriod we calculated may **not be optimized**. • Calculate **the average of data points and move the centroid**. • Repeat cluster assignment until the **centroid stops moving**, if the centriod movement has stopped we can say our algorithm is **optimized**.	
Convergence	Now we can say the algorithm is converged and can give **a clear result**.	



a) A labeled dataset has three distict groups

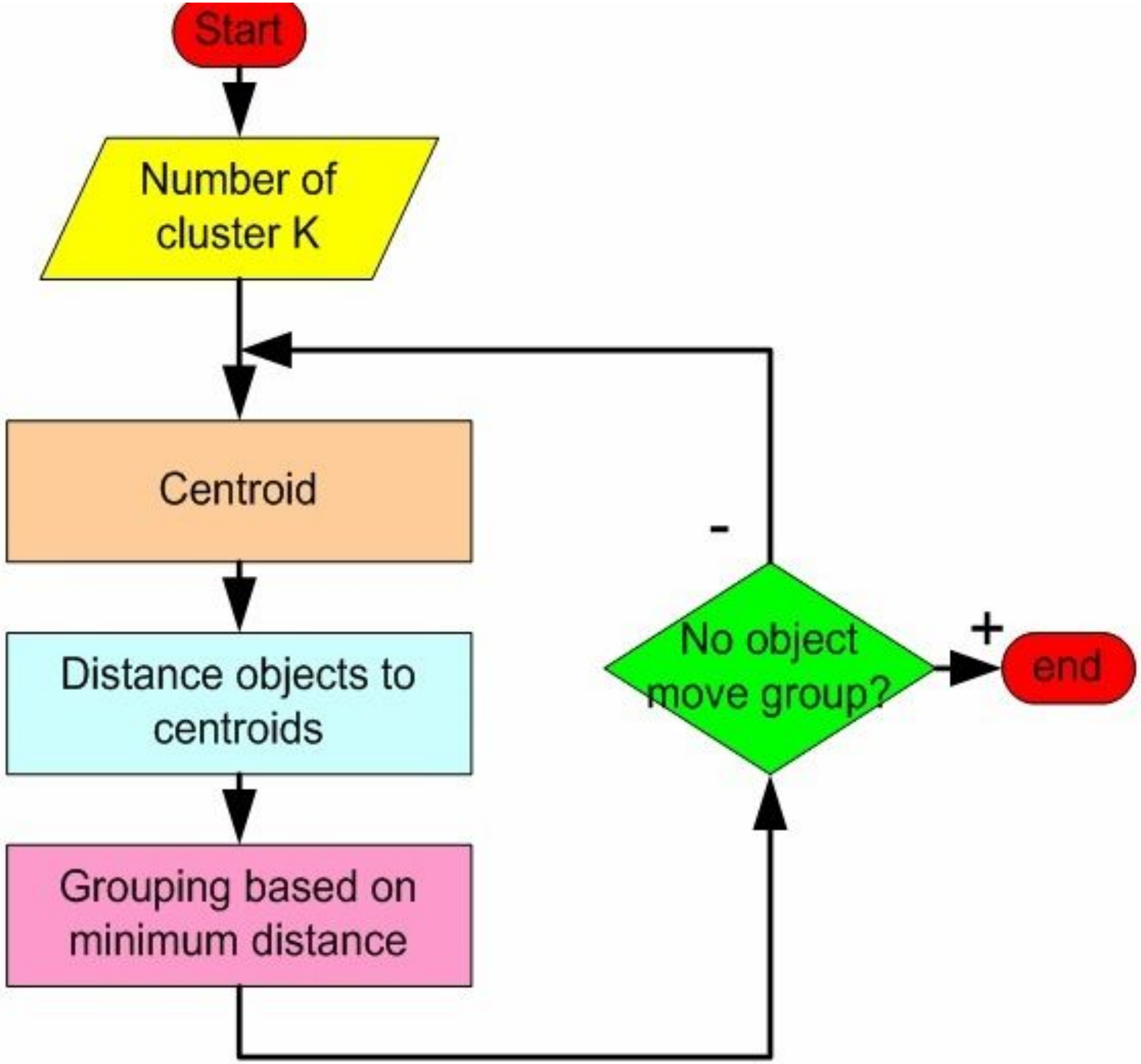
Steps in K means clustering

- **Training and saving a model using spark and Mlllib**

1. **Extract all the features** required.
2. From the training set we can use the **data for training our k means cluster model.**
3. Take **the number of clusters** required for solving the problem.(**Elbow Method, Silhouette Score, Domain Knowledge, Trial and Error**)
4. Here assume the cluster is **2** because we got class a and b

- Import K means from **mllib library**.
- Convert our **extracted features into a resilient distributes dataset(rdd).(refer slide no : 33 for conversion steps)**
- **Set number of clusters as two** (during training k means algorithm will partition the training data into any of these two clusters).
- Train the **model using training set**.
- Apply sample predictions
- **Saving the trained model in HDFS and loading for future use**

A	B	C
Income	Size	Distance from Urban area
100000	15000	10
1500000	30000	2
1500000	30000	2
1500000	30000	2
1500000	30000	2
1500000	30000	2


Start
Number of cluster K
Centroid
Distance objects to centroids
Grouping based on minimum distance
No object move group?
-
+
end

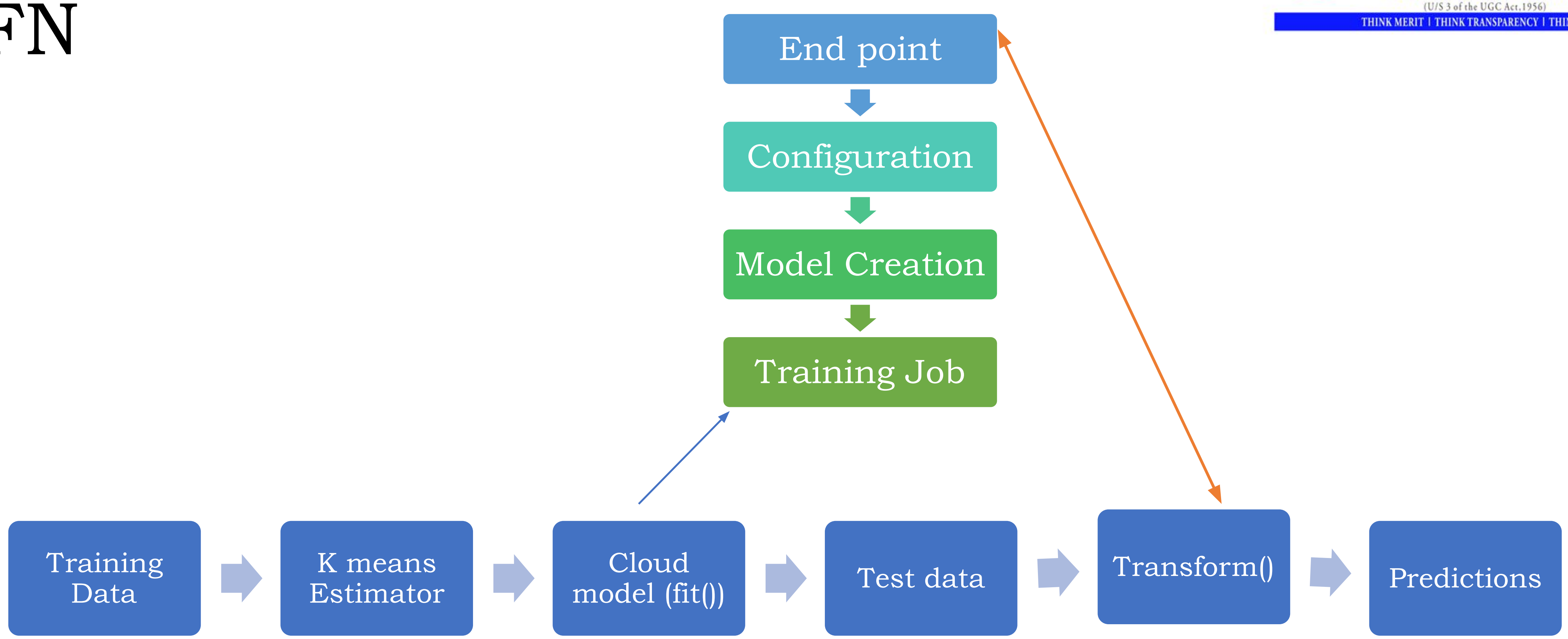
K-means Clustering

- **Scenario:** Suppose you have 10,000 points on a plane, and you want to find **k new points** that are the centroids of k clusters that partition the set.
- **Assumptions:**
  - Points to hold the **k centroids**
  - Initialize this array with **random values**
  - Apply an **iterative MapReduce algorithm**
  - Repeating these two steps until the **centroids have not moved far from their previous position**

- For each point, find the **INDEX OF THE CENTROID** to which it is **nearest and assign the point** to the cluster associated with that nearest centroid.
- For each cluster, **COMPUTE THE CENTROID** of all points in that cluster, and replace the previous **centroid in kPoints with that new centroid**.

Objective: To compute, for a given point p, the index of the centroid in kPoints to which p is nearest

Generic flow for K means cluster using CFN



```
def closestPoint(p, kPoints):
bestIndex = 0
closest = float("+inf")
for i in range(len(kPoints)):
tempDist = np.sum((p - kPoints[i]) ** 2)
if tempDist < closest:
closest = tempDist
bestIndex = i
return bestIndex
data.map( lambda p: (closestPoint(p,
kPoints),
(p, 1)))
reduceByKey(lambda x, y : (x[0] + y[0], x[1] +
y[1]))
tempDist = 1.0
while tempDist > convergeDist:
newPoints = data .map( lambda p:
(closestPoint(p, kPoints), (p, 1))) \
.reduceByKey(lambda x, y : (x[0] + y[0],
x[1] + y[1])) \
.map(lambda x : (x[0], x[1][0]/x[1][1]))
\
.collect()
tempDist = sum(np.sum((kPoints[i] - y) ** 2) \
for (i, y) in newPoints)
for (i, y) in newPoints:
kPoints[i] = y
```

- The **locations of our 10,000 points are in an array called data**.
- **We use the following map expression to create, as a new RDD, the set of tuples (j, (p, 1)) in which**
- **j is the index of the closest centroid** for each **p in data use of tuples of form (p, 1**) is a common **MapReduce idiom**.
- We want to **compute for each j** the sum of all tuples (p, 1), to obtain tuples of the form.
- The sum of the **1s is just the count of the j tuples in our se**t, so we can compute the centroid by dividing the sum of the ps by this count.
- This is an **RDD of size k**, and we can collect it and use it as the **kPoints for the next iteration**

SPARK PROCESS	Kmeans

Syntax 1: data.map(lambda p: (closestPoint(p, kPoints), (p, 1)))
//decide which of your data points belong to the clusters defined by your k centroid points
 	
Syntax 2: reduceByKey(lambda x, y : (x[0] + y[0], x[1] + y[1]))
//choose new centroids based on the average of the points in a cluster
 	
Syntax 1 + Syntax 2=finding the average of all of the points in a cluster.
 	
collect(): The **final 'collect' method call moves the data from an RDD to the local machine** (in preparation of broadcasting the new centroids back out the all of the nodes in your Spark environment).
	• Randomly chose **k points as the centroids of your clusters**
	• Decide which of your data points belong to the clusters defined by your k centroid points(**Elbow Method, Silhouette Score, Domain Knowledge, Trial and Error)**
	• Choose new centroids based on the average of the points in a cluster
	• **Repeat steps 2 & 3 until you decide your are finished**
	• **SPARK PROCESS**
	• Import K means from mllib library.
	• Convert our **extracted features into a Resilient Distributes Dataset(RDD)**.
	  • **clusters.save(sc,"sample_model")**
	• Set number of **clusters as two (During training K means** algorithm will partition the training data into any of these two clusters).
	• train the model using training set.

- Having a map phase
- Followed by a reduce-by-key
- Followed by another map
- Finally a collect that brings the value of newpoints back to the read eval-print-loop.
- Each spark operation is executed on the cluster of cores in which the **RDD** is located.
- In fact, what the python program is doing is compiling a **graph that is then executed by the spark engine**

Spark in a Container

Maps port 8888 on the host to port 8888 in the containers

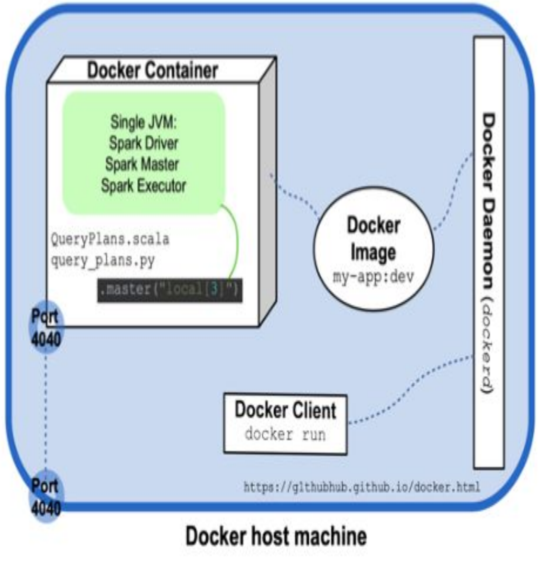
- Easily run Spark on a remote VM with many cores: as long as the VM has Docker installed

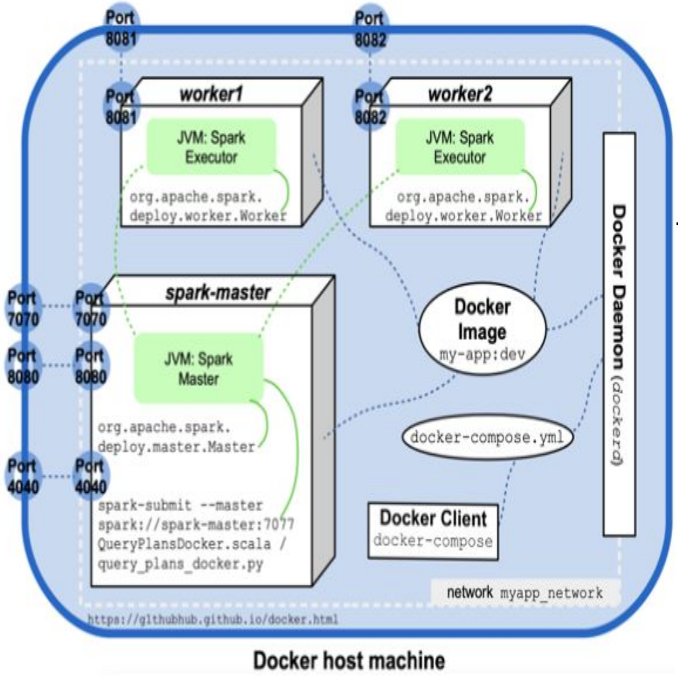
certificate generation

```
docker run -e GEN_CERT=yes -d -p 8888:8888 \
-v /tmp/docmnt:/home/jovyan/work/docmnt \
jupyter/all-spark-notebook startnotebook.
sh \
-- NotebookApp.password='sha1:....'
```

```
docker run -e GEN_CERT=yes -d -p
8888:8888 --
cpuset-cpus 0-3 -m 10G\
-v
/tmp/docmnt:/home/jovyan/work/docmnt
\
jupyter/all-spark-notebook startnotebook.
sh \
--NotebookApp.password='sha1:....'
```

d: runs in the background





https://g1thubhub.github.io/docker.html

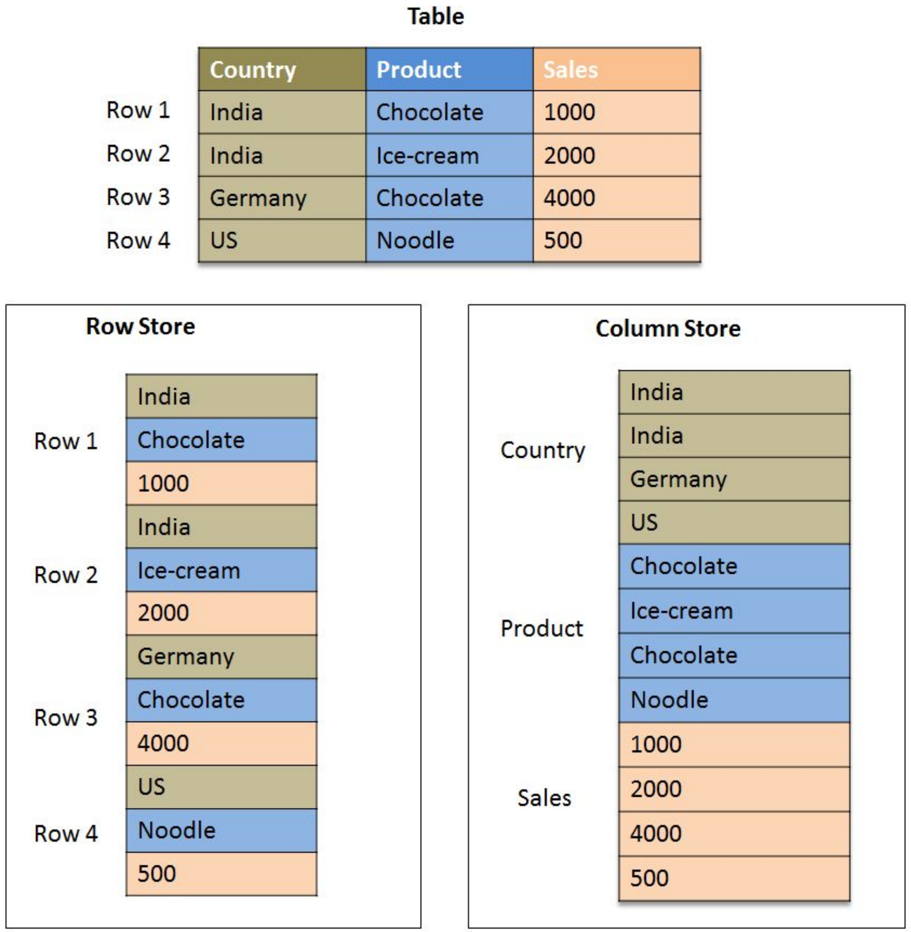
SQL in Spark

- Spark SQL brings native support for SQL to Spark and streamlines the process of querying data stored both in **RDDs (Spark's distributed datasets) and in external sources**.
- Spark SQL conveniently blurs the lines between RDDs and relational tables.
- We have a **comma-separated value (CSV) file**, hvac.csv, with a header and three data lines.
- Import relational data from **PARQUET FILES** and **HIVE TABLES**
- It is a columnar storage file format, designed for efficiency and optimized for analytical workloads, especially in big data environments.
- Run SQL queries over **imported data and existing RDDs**
- Easily write RDDs out to Hive tables or Parquet files

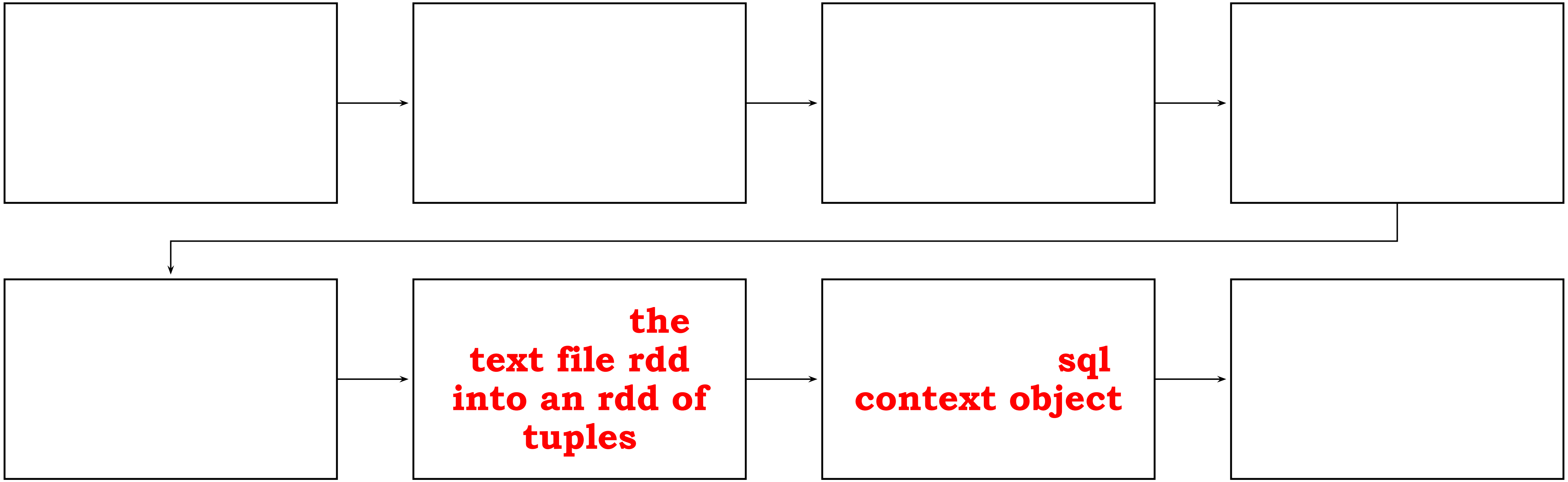
Parquet files

- Apache Parquet is a popular **COLUMN STORAGE FILE FORMAT** used by Hadoop systems, such as Pig, Spark, and Hive.
- The file format is language independent and has a binary representation.
- Parquet is used to efficiently store large data sets and has the extension **.parquet**.

Dataset	*Size on Amazon S3*	*Query Run Time*	*Data Scanned*	*Cost*
Data stored as CSV files	1 TB	236 seconds	1.15 TB	$5.75
Data stored in Apache Parquet Format	130 GB	6.78 seconds	2.51 GB	$0.01
Savings	87% less when using Parquet	34x faster	99% less data scanned	99.7% savings



SQL in Spark



- Load this file into spark and create an RDD by applying the textfile operator to the spark context object.
- **Convert the text file rdd into an rdd of tuples** by stripping off the header and mapping the rest to typed tuples.
- **Create an sql context** object and schema type, and then an sql dataframe
- Ready to execute SQL operations on our data.
- For example, we can use the sql() method to extract a new sql rdd dataframe consisting of the buildingid column

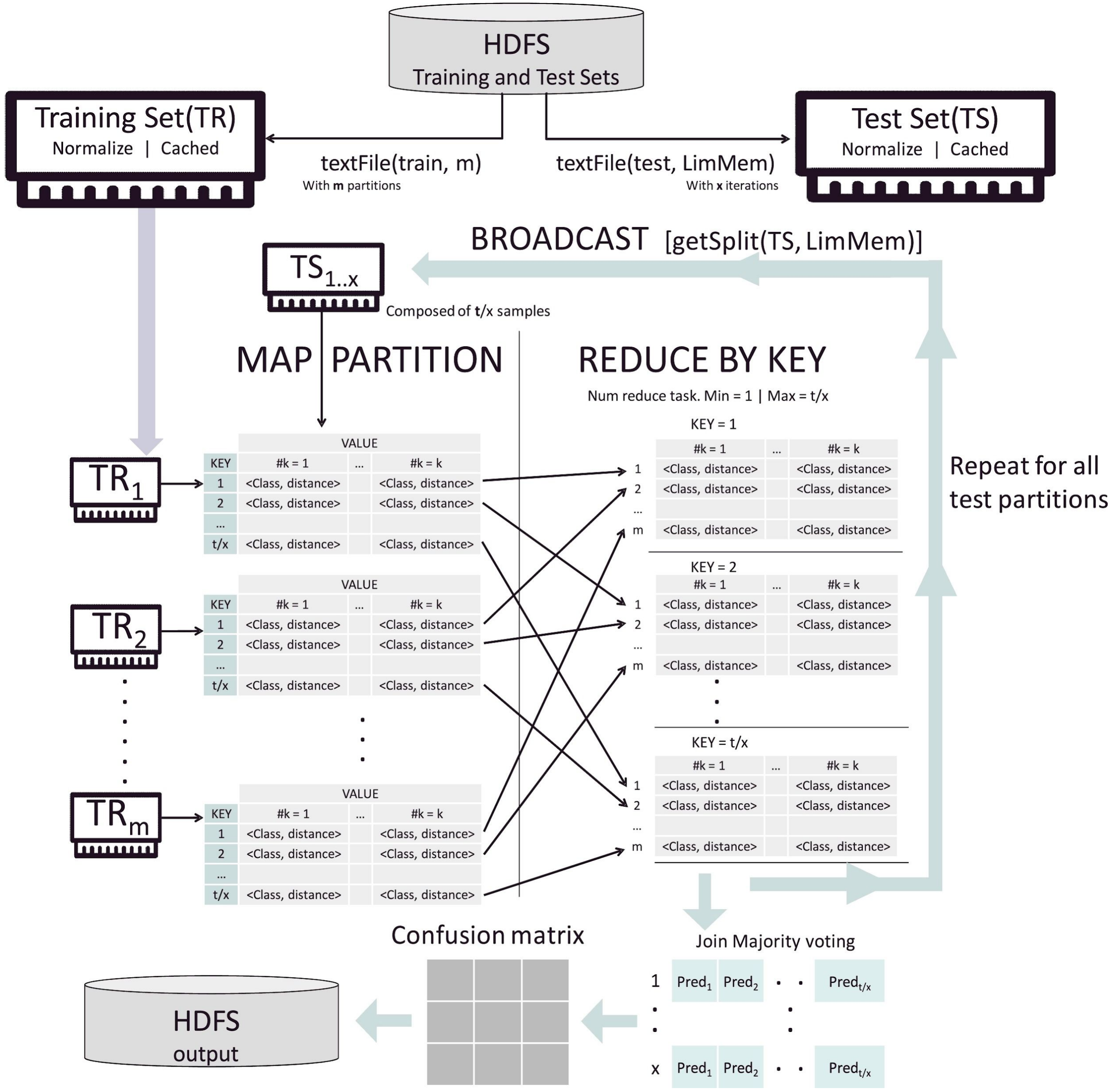
This refers to the SQLContext (or SparkSession) object in PySpark, which allows you to run SQL queries on structured data

that selects the buildingID column from the hvac table. hvac could be a DataFrame that was registered earlier as a table or view.

- `x = sqlCtx.sql('SELECT buildingID from hvac')`

```
from pyspark.sql.types import *
hvacText = sc.textFile("/pathto/file/hvac.csv")
hvac = hvacText.map(lambdas: s.split(",")) \
.filter(lambda s: s[0] != "Date") \
.map(lambda s:(str(s[0]), str(s[1]),
int(s[2]), int(s[3]),
str(s[4]) ))
sqlCtx = SQLContext(sc)
hvacSchema = StructType([StructField("date",
StringType(), False),
StructField("time", StringType(),
False),
StructField("targettemp",
IntegerType(), False),
StructField("actualtemp",
IntegerType(), False),
StructField("buildingID",
StringType(), False)])
hvacDF = sqlCtx.createDataFrame(hvac,
hvacSchema)
```

```
%%sql_show
SELECT buildingID ,
      (targettemp - actualtemp) AS temp_diff,
      date FROM hvac
WHERE date = "3/23/2016"

+-----------+---------+---------+
| buildingID|temp_diff|     date|
+-----------+---------+---------+
|headquarters|       13|3/23/2016|
|       lab1|      -10|3/23/2016|
|   coldroom|       34|3/23/2016|
+-----------+---------+---------+
```

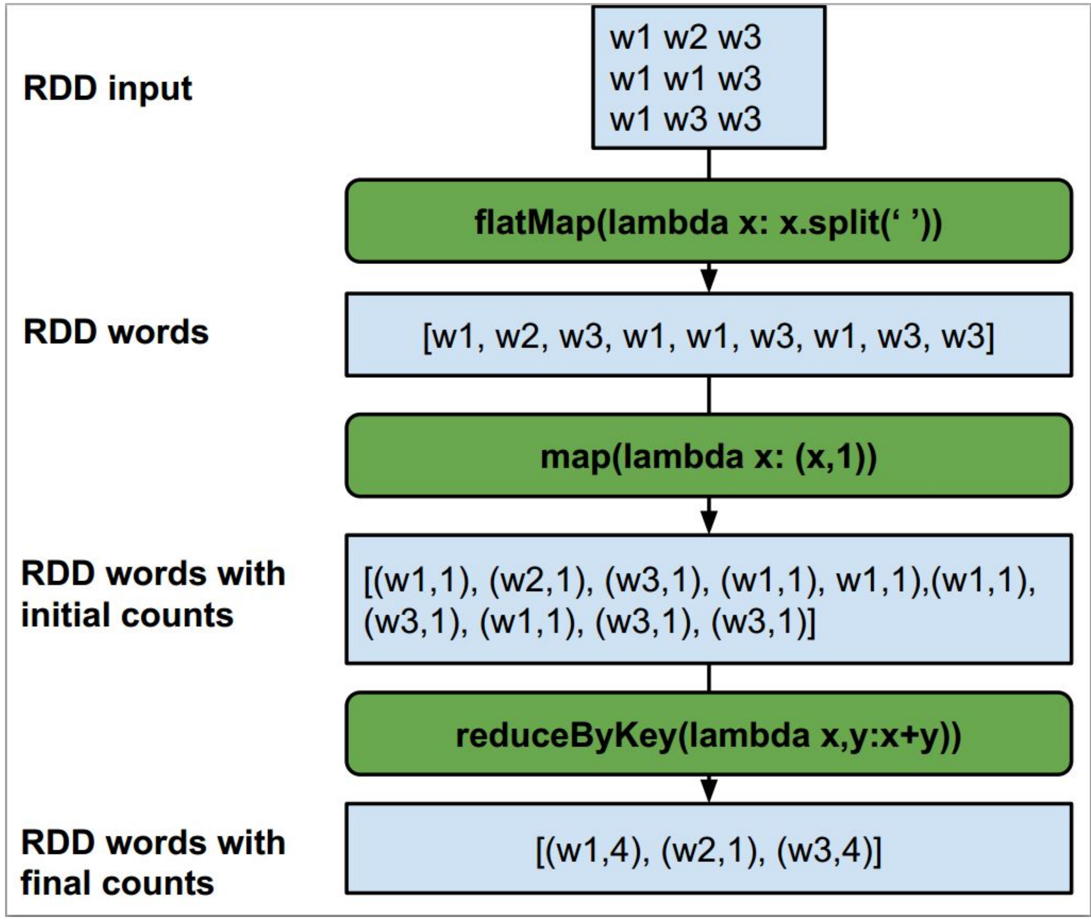


https://sparkbyexamples.com/

- First, the flatMap function takes the input file that is returned by the sc.textFile function that returns the lines of the file.
- This flatMap does two things it applies the **lambda function to each line**, creating a list of space separated words.
- Then the second thing flatMap does by default is flattening the list of lists, meaning that [[w1,w2],[w2,w3]] becomes [w1,w2,w2,w3]

- Second, a map function is applied to the resulting RDD that is produced by flatMap.
- The map operation applies the lambda function provided to **each element in the RDD**.
- Here each element is a word in the list of words RDD and the map produces a pair for each word composed of the word as the key and the initial count as of that word as 1.

- Finally, the aggregation is performed with the **reduceByKey** function on the resulting RDD.
- This is similar to the regular reduce operation that takes two elements and applies a function to those two elements, but in this case the words are first grouped by key, which in the case of (w1,1) is the word part of the pair.
- This gives the following [(w1,1),(w1,1),(w2,1)] ==> [(w1,2),(w2,1)]



Amazon Elastic MapReduce

- Amazon EMR is a **managed cluster platform** that simplifies running big data frameworks, such as ***Apache Hadoop and Apache Spark, on AWS*** to process and analyze **vast amounts of data**.
- By using these frameworks and related open-source projects, such as **Apache Hive and Apache Pig**, you can process data for analytics purposes and business intelligence workloads.
- Additionally, you can use Amazon EMR to transform and move large amounts of data into and out of other AWS data stores and databases, *such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB*.

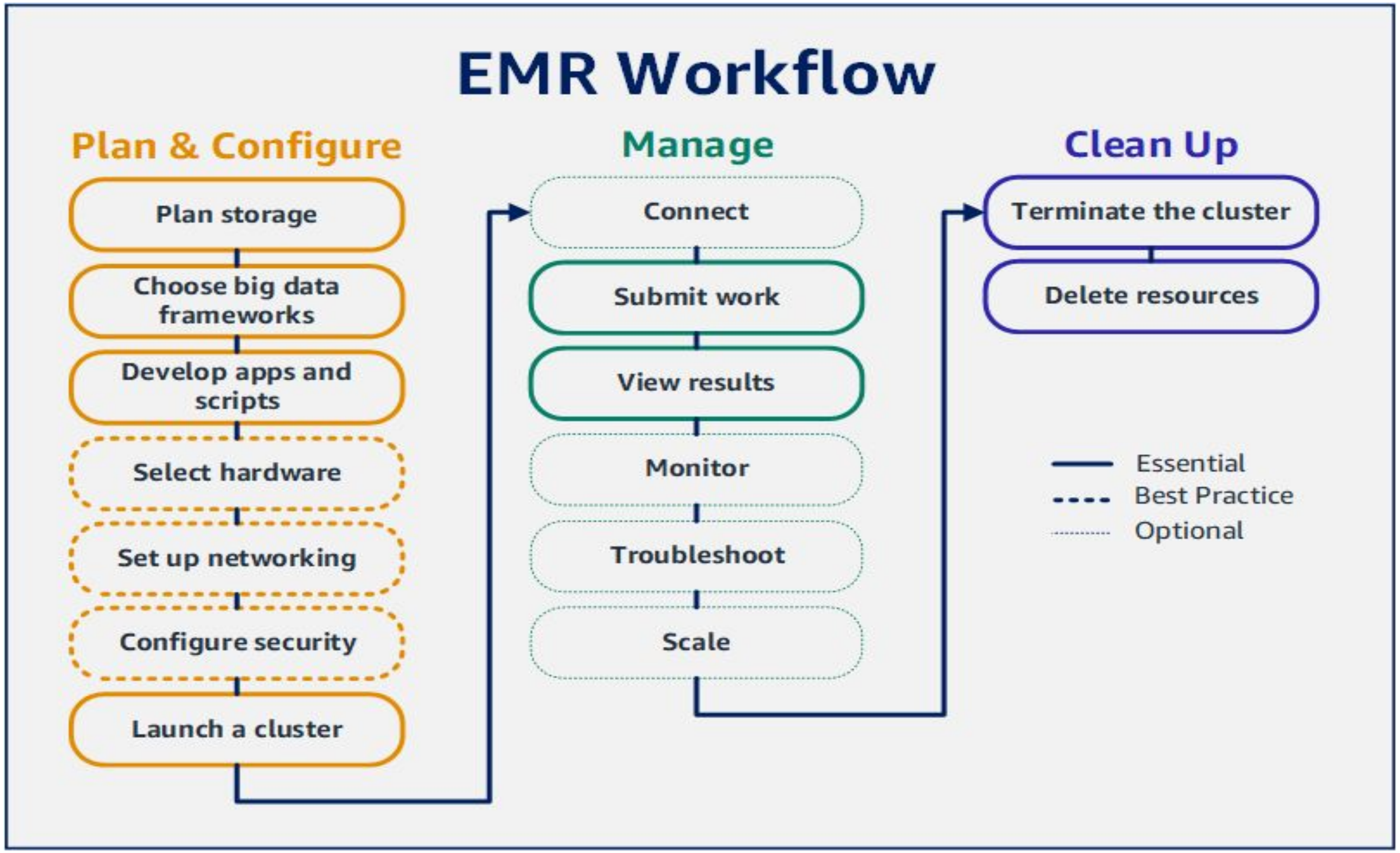
Overview of Amazon EMR

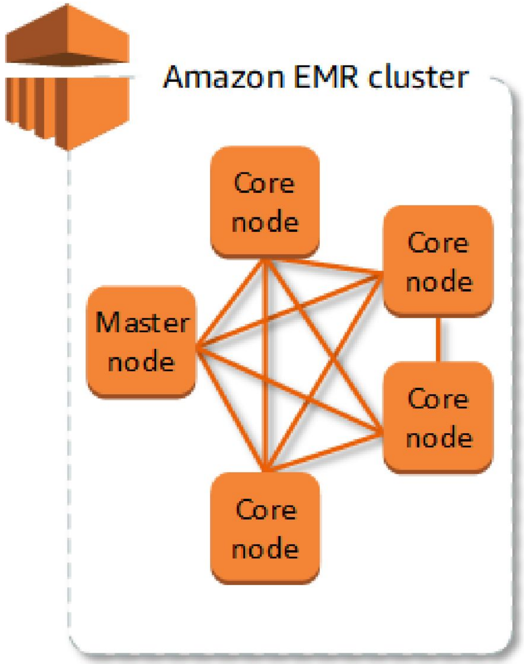
Formation of CFN cluster using AWS

- Understanding **Clusters and Nodes**
- The central component of Amazon EMR is the **CLUSTER**.
- A cluster is a collection of **Amazon Elastic Compute Cloud (Amazon EC2) instances**.
- Each instance in the **CLUSTER IS CALLED A NODE**.
- Each node has a role within the cluster, referred to as the node type.
- Amazon EMR also installs different software components on each node type, giving each node a role in a distributed application like **Apache Hadoop**.

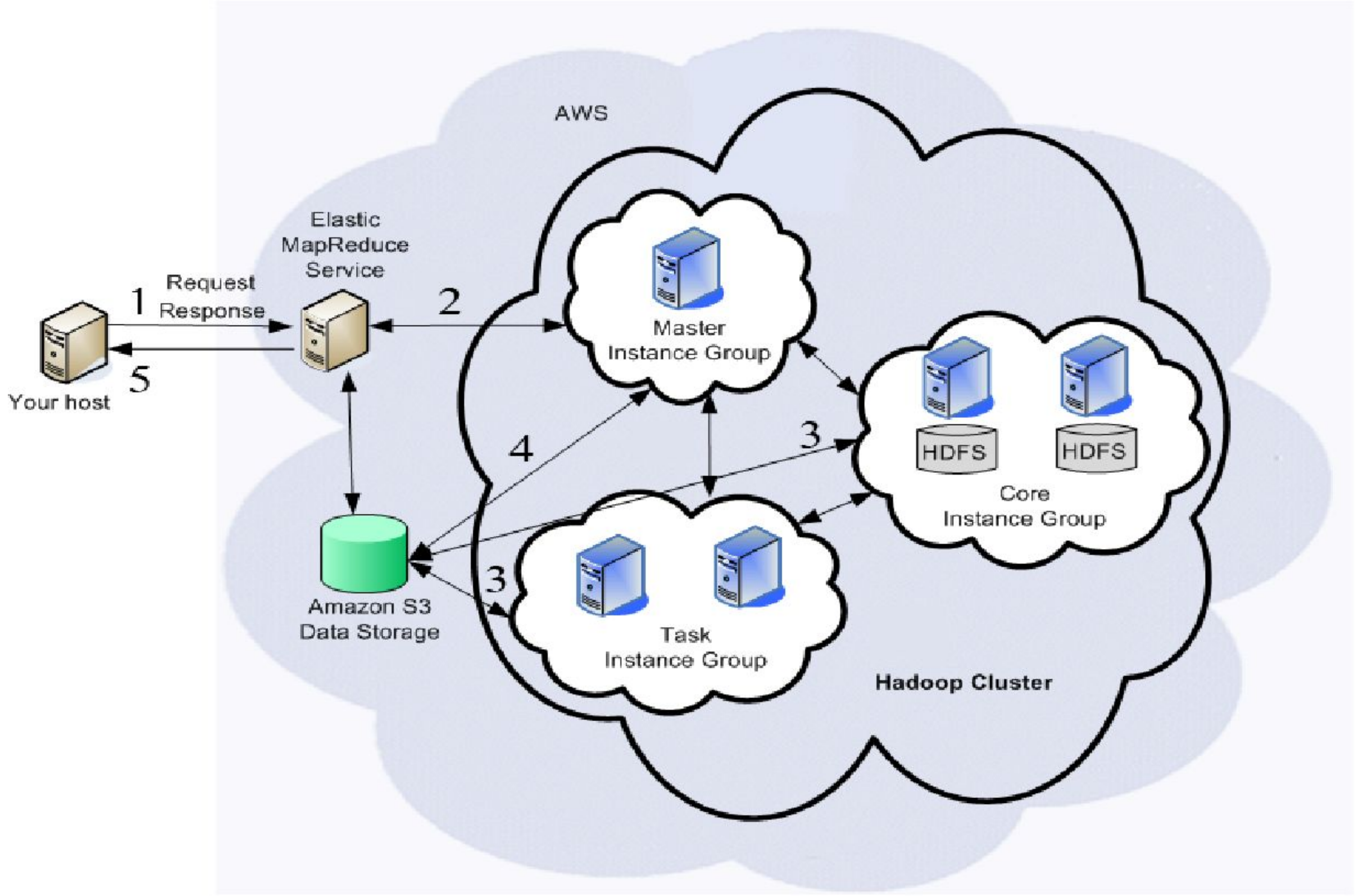
- The node types in Amazon EMR are as follows:
- **Master node:** A node that manages the cluster by running software components to **coordinate the distribution** of data and tasks among other nodes for processing.
- The master node tracks the status of tasks and monitors the health of the cluster.
- Every cluster has a master node, and it's possible to create a single-node cluster with only the master node.
- **Core node:** A node with software components that **run tasks and store data in the Hadoop Distributed File System (HDFS**) on your cluster.
- Multi-node clusters have at least one core node.

- **Task node:** A node with software components that only **runs tasks** and **does not store data in HDFS**.

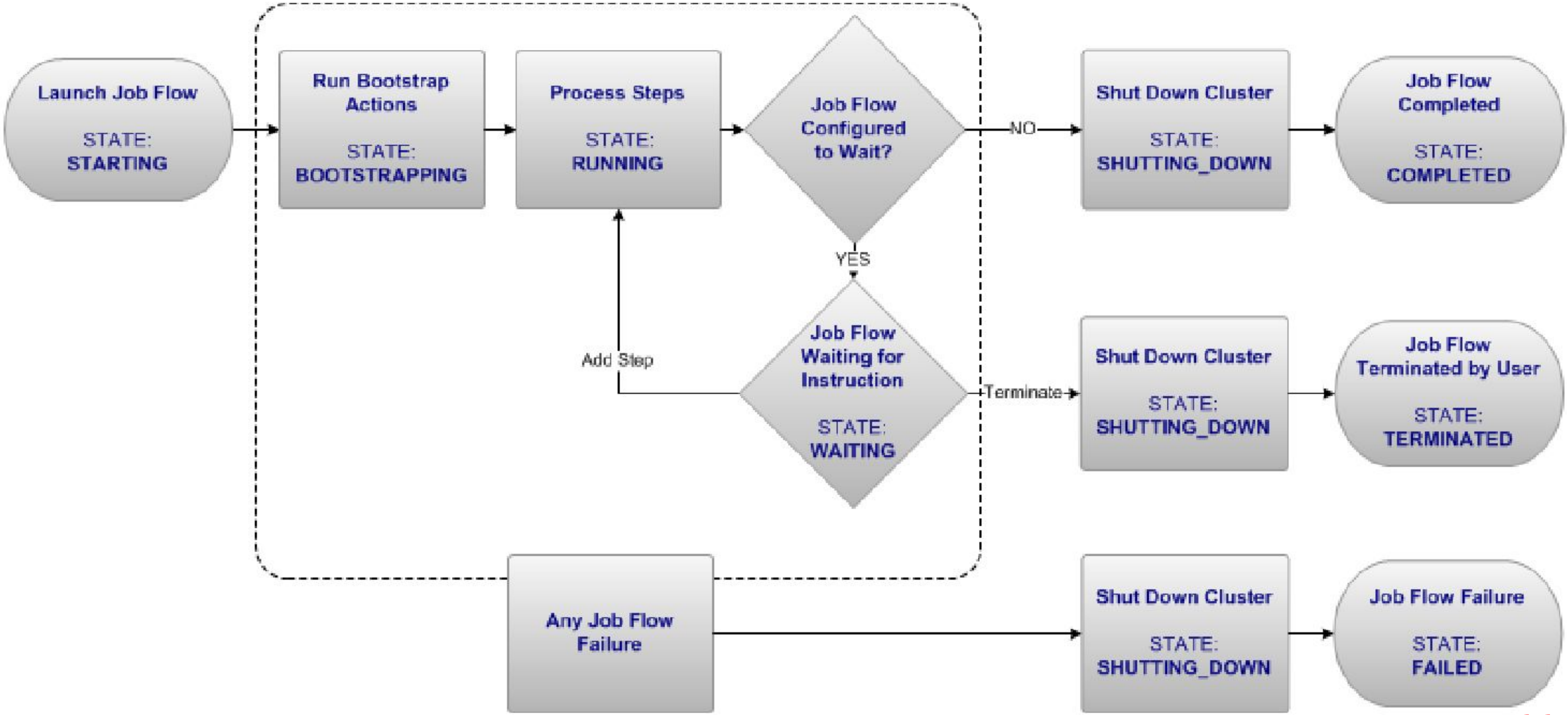




Architectural Overview of Amazon EMR



Operations in AEMR



- First load a small sample of wikipedia **access logs—from 2008 to 2010 —from S3**.
- Text-file rdd initially has just **one partition**.
- Repartition the rdd into **10 segments**.
- To count the NUMBER OF HITS for each person
- Define a **function mapname** that returns
- The name of the person in the page title
- Execute a map to replace each row with a new pair consisting of the name
- The value in the count field for that row

- RDD remapped with a version of the take() function

```
# Define list of famous names
namelist = ['Albert_Einstein', 'Lady_Gaga', 'Barack_Obama',
    'Richard_Nixon','Steve_Jobs', 'Bill_Clinton', 'Bill_Gates',
    'Michael_Jackson','Justin_Bieber', 'Vladimir_Putin',
    'Byron', 'Donald_Trump', 'Hillary_Clinton', 'Nicolas_Sarkozy',
    'Werner_Heisenberg', 'Arnold_Schwarzenegger', 'Elon_Musk',
    'Vladimir_Lenin', 'Karl_Marx', 'Groucho_Marx']

# Transform a line into an array by splitting on blank characters
def parseline(line):
    return np.array([x for x in line.split(' ')])

# Filter out lines not containing famous name in the page title
def filter_fun(row, titles):
    for title in titles:
        if row[1].find(title) > -1:
            return True
    else:
        return False

# Return name of person in page title
def mapname(row, names):
    for name in names:
        if row[1].find(name) > -1:
            return name
    else:
        return 'huh?'

# ------ Load and process data ------------------------------------
# Load Wikipedia data from S3
rawdata = sc.textFile( \
        "s3://support.elasticmapreduce/bigdatademo/sample/wiki")

# Repartition initial RDD into 10 segments, for parallelism
rawdata = rawdata.repartition(10)

# Split each line into an array
data = rawdata.map(parseline)

# Filter out lines without a famous name
filterd = data.filter(lambda p: filter_fun(p, namelist))

# Map: Replace each row with (name, count) pair.
# Reduce by name: Add counts
remapped =filterd.map(lambda row:(mapname(row,namelist),int(row[2])))
                .reduceByKey(lambda v1, v2: v1+v2)
```

Azure HDInsight and Data Lake

- MapReduce framework called **COSMOS** that is used as the main data analytics engine for many internal projects such as the Bing search engine.
- Cosmos is based on a **DIRECTED GRAPH EXECUTION** model and is programmed in an SQL-like language called SCOPE.
- While Cosmos was a candidate for release as a general **MapReduce tool for Azure users**.
- The company decided that the **Hadoop/YARN ecosystem** was of such great interest to their customers that they would keep Cosmos internal and support YARN as the public offering

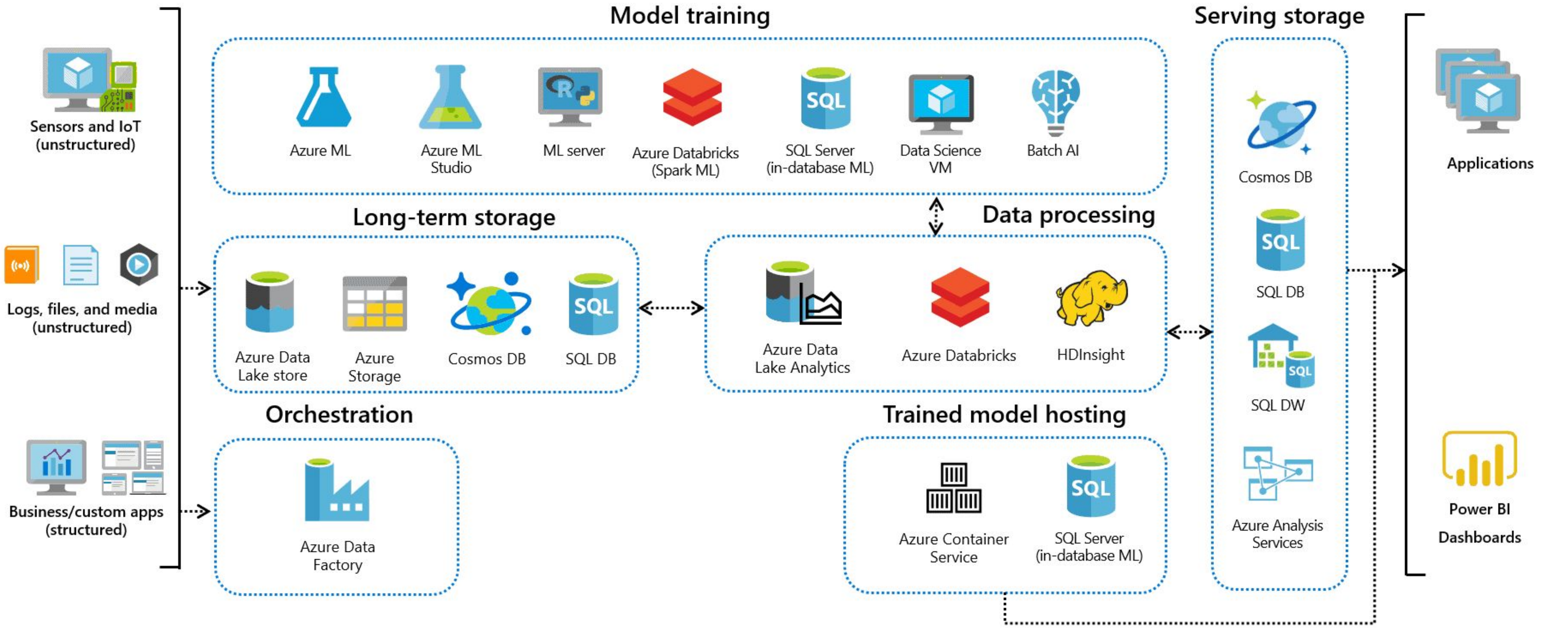
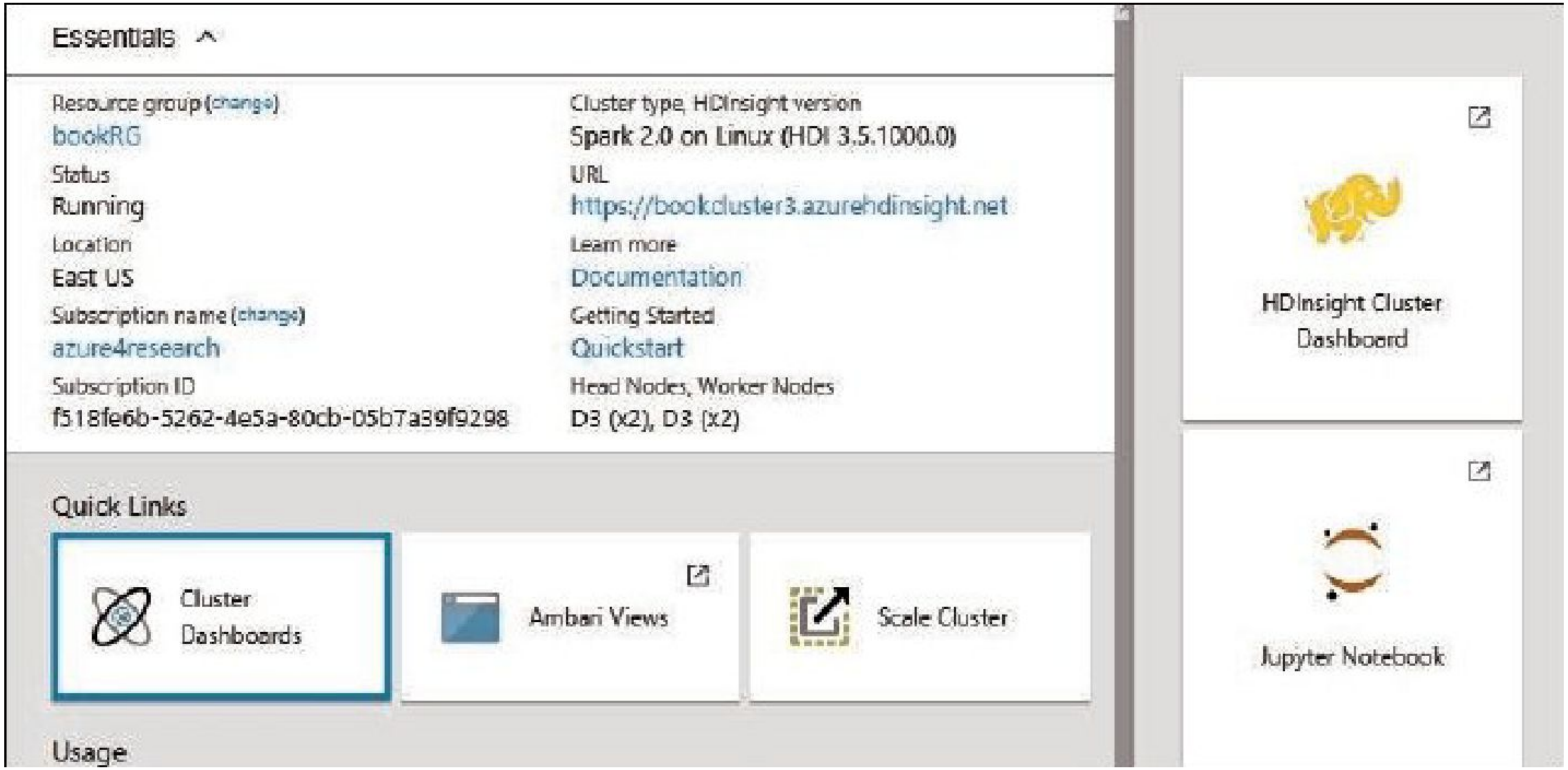


Image Source : © Microsoft Corporation

- Called HDInsight on Azure, this service supports **Spark, Hive, HBase, Storm, Kafka, and Hadoop MapReduce**, and is backed by a guarantee of 99.9% availability.
- HDInsight is based on the Hortonworks Data Platform distribution integrated with **Azure security**.
- All of the usual Hadoop and YARN components are there, **including HDFS as well as tools that integrate** other Microsoft business analytics tools such as Excel and SQL Server
- Creating an **HDInsight cluster with Spark and Jupyter** installed is easy.
- As with many Azure services, there is a **preconfigured template** that you can use.

- Clicking on the **DEPLOY TO AZURE** link in the documents takes you to your azure account and sets up the script.
- Need to fill in a few standard account details, such as a ***name for your cluster, the login id, and the password for the ssh and master nodes***.
- Clicking on the **HDINSIGHT CLUSTER** dashboard icon provides you with live status data about the cluster, such as **memory, network, and cpu use.**
- Clicking on the jupyter icon first authenticates you and then takes you to the jupyter home, where you **have a choice of pyspark or scala notebooks.**

Azure portal view of a HDInsight cluster

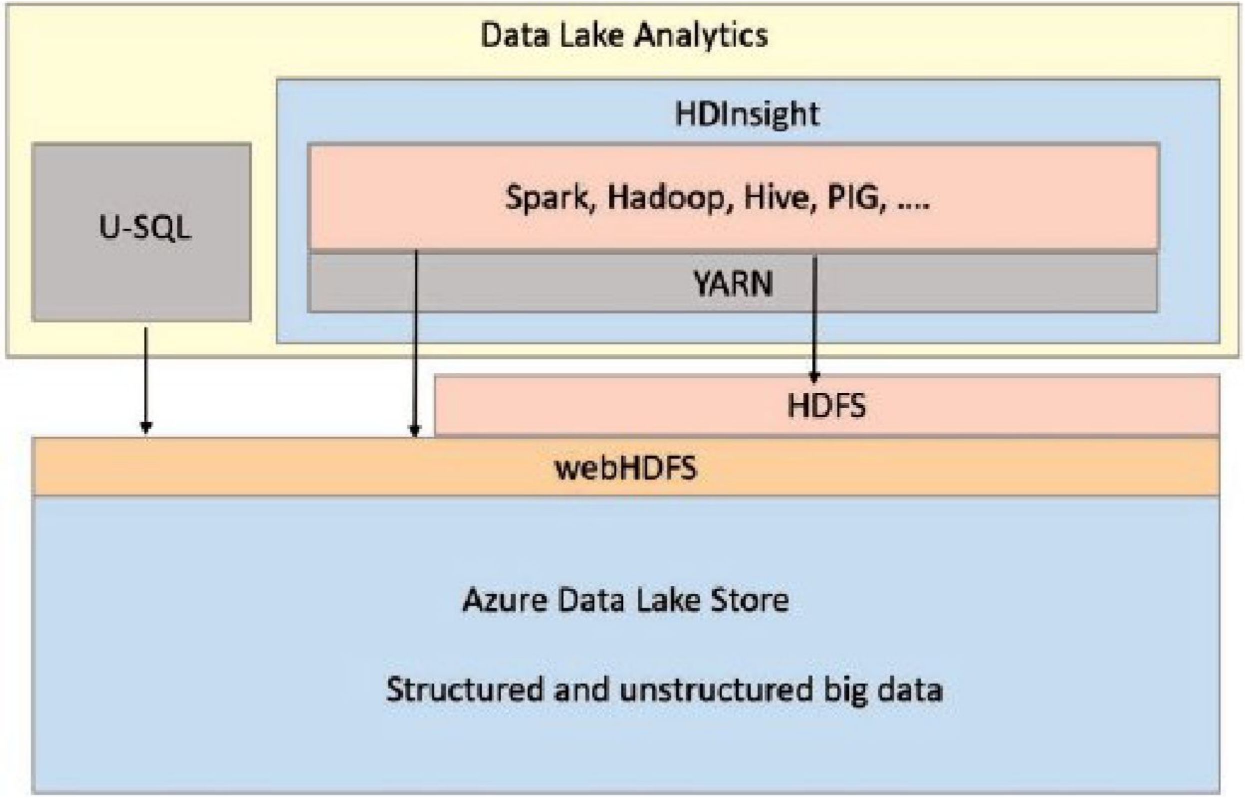


Azure Data Lake Store

- Data Lake also includes the Azure Data Lake Store, a **data warehouse for petascale data collections**.
- It does not have the 500 TB storage limits of Azure blobs, and is designed for massive throughput.
- It supports both **structured and unstructured data**.
- The access protocol for the Data Lake Store, **WebHDFS, is the REST API for HDFS.**
- Consequently you can access the Data Lake Store from anywhere, and with the same commands that you use to access HDFS.

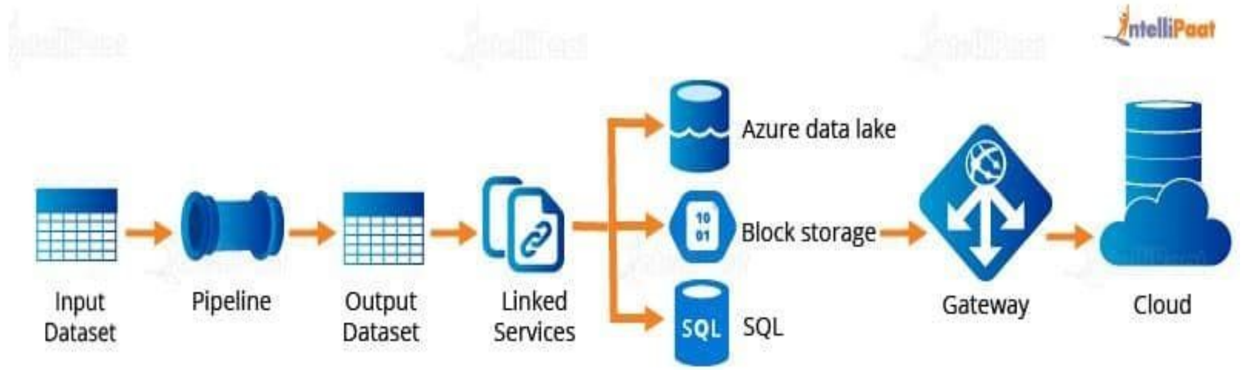
```
> python azure/datalake/store/cli.py
azure> help

Documented commands (type help <topic>):
========================================
cat    chmod  close  du     get    help  ls    mv    quit  rmdir
touch  chgrp  chown  df     exists head  info  mkdir put   rm
tail
azure>
```

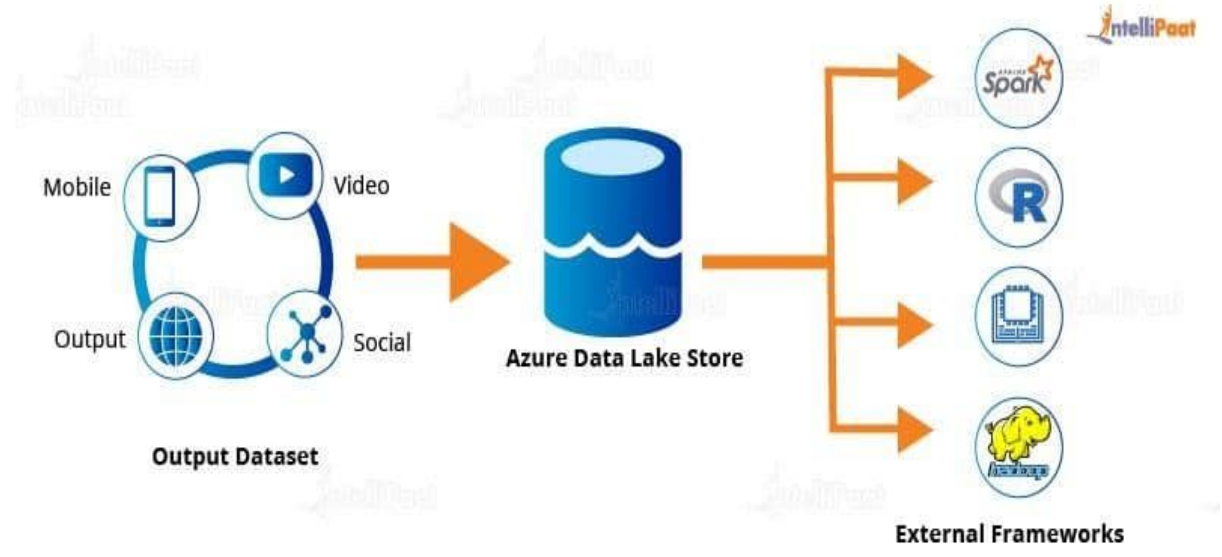
Conceptual view of the components that make up the Azure Data Lake

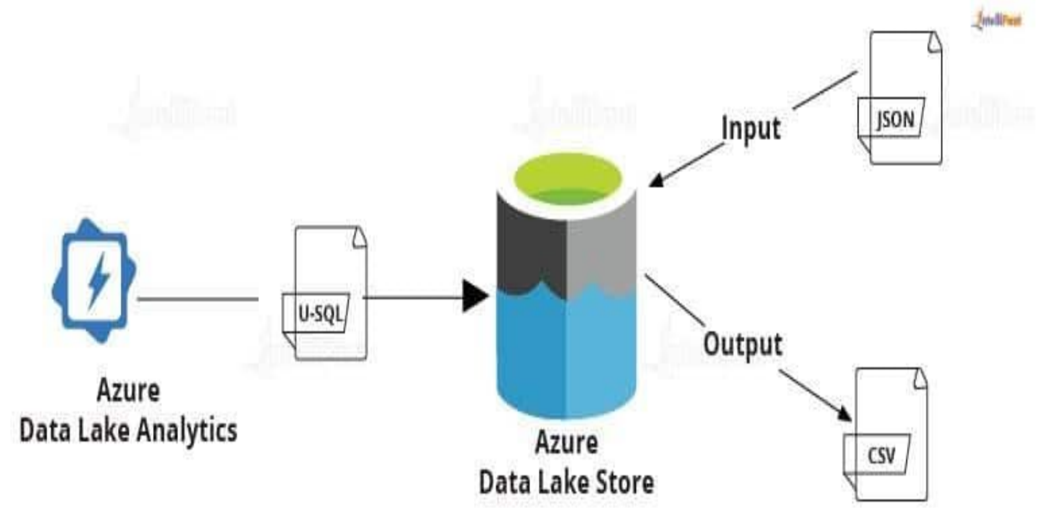


Data Lake Analytics (**Architecture for scenario based**)

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

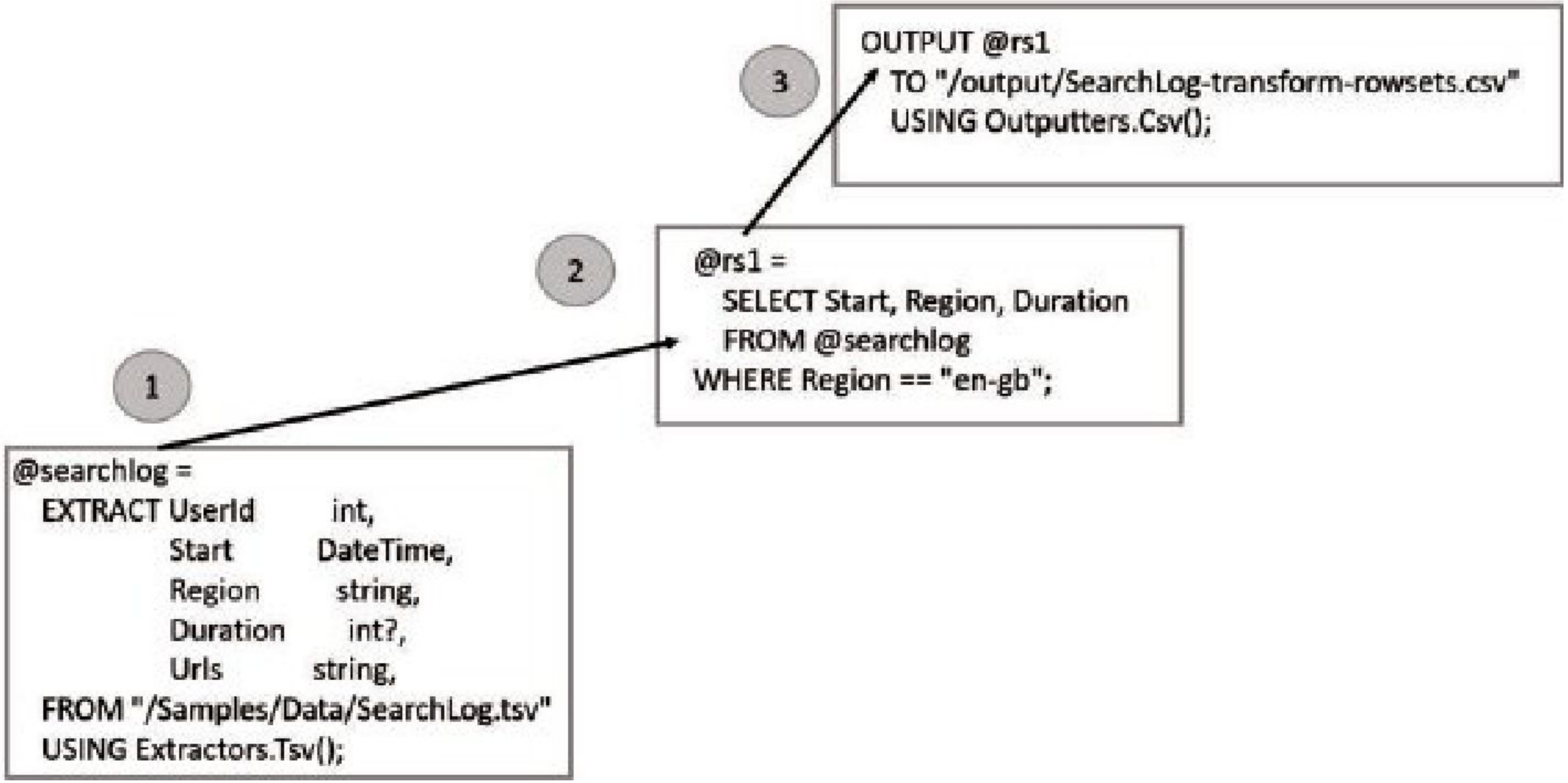






Data Lake Analytics

- Azure Data Lake Analytics consists of HDInsight and associated tools (Spark, Hadoop, etc.), plus a data analytics tool called **U-SQL for scripting large analytics tasks.**
- U-SQL combines **SQL queries and declarative program functions written in C#, Python, or R**.
- U-SQL is intended for massively parallel analysis of terabyte- to petabyte-size data collections.
- When you write a **U-SQL script, you are actually building a graph.**
- Through the scalable, distributed-query capability of **U-SQL, you can efficiently analyze data across relational stores such as Azure SQL Database**

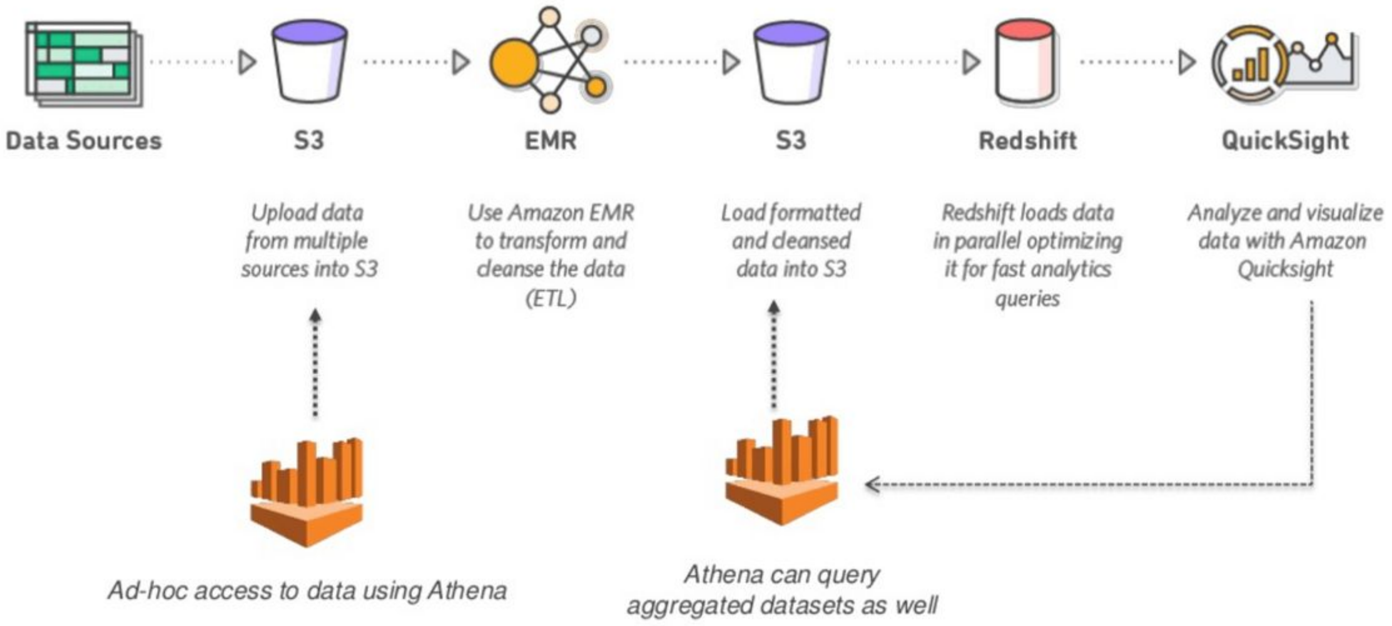


- A **Unified-SQL** program defines a **graph of queries**.
- In the first step an object **searchlog** is extracted from a file.
- In step two the object rs1 is created from **searchlog by** extracting items with **region "en-gb".**
- In the final step the object rs1 is converted to a **CSV file and saved.**

- **EXTRACT**: The EXTRACT statement **reads data from the CSV file in Azure Data Lake Storage** and defines the schema for the data (i.e., the data types for each column).
- **FROM:** Specifies the **path to the CSV file in Azure Data Lake Storage.**
- **USING Extractors.Csv():** Instructs **USQL to parse the file using the CSV extractor.**
- **SELECT:** Performs a query to compute
- **OUTPUT:** **Writes the result of the query** to a new CSV file in Azure Data Lake Storage.
- **USING Outputters.Csv():** Specifies that the **output format should be CSV.**

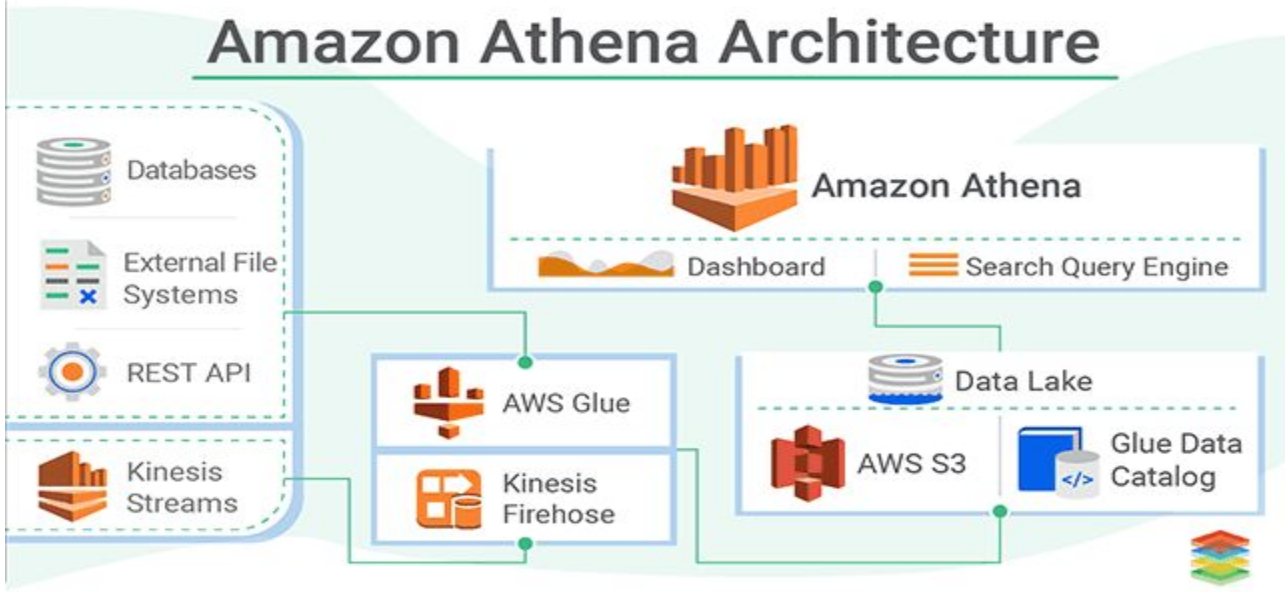
Amazon Athena Analytics

- Athena, a recent addition to the Amazon analytics toolbox, is designed to allow users to **query data** in Amazon **S3 without having to launch VM instances**.
- Like Data Lake Analytics, Athena is an example of the concept of **SERVERLESS COMPUTING**



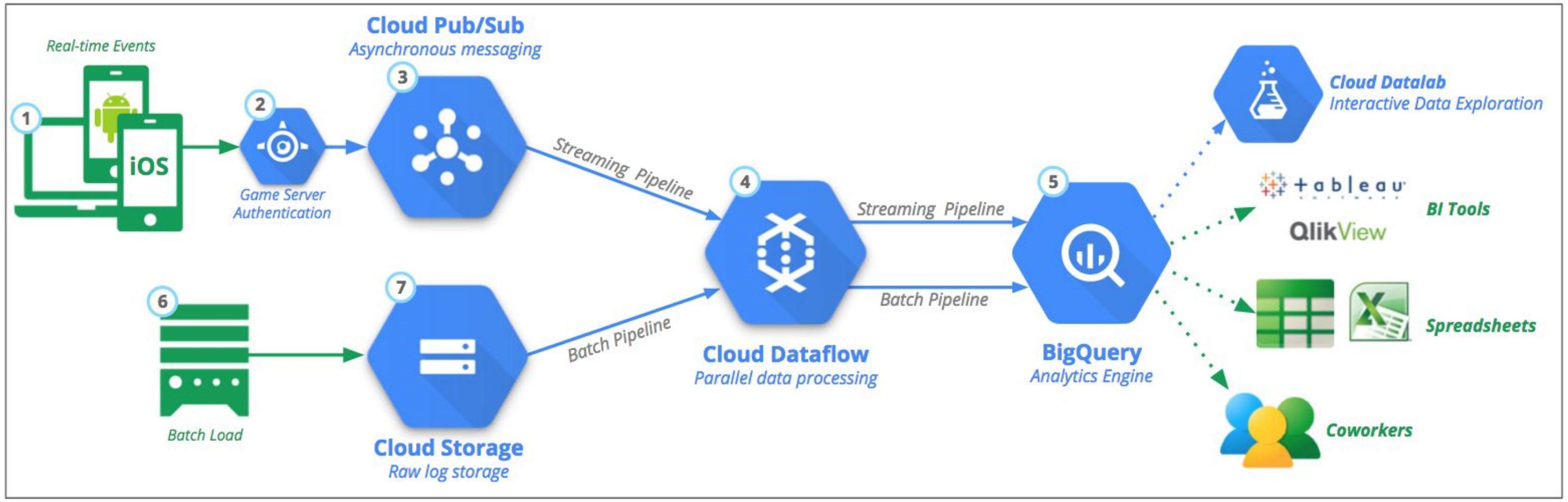
- Athena is accessed via the Amazon portal, which provides an interactive **query tool based on standard SQL**.
- Place your data in S3 in one of several standard forms, including text files, CSV files, Apache web logs, JSON files, and a column-oriented format such as Apache Parquet.
- ***Parquet files store the columns of a data table*** *in contiguous locations, suitable for efficient compression and decompression into distributed file systems like HDFS.*

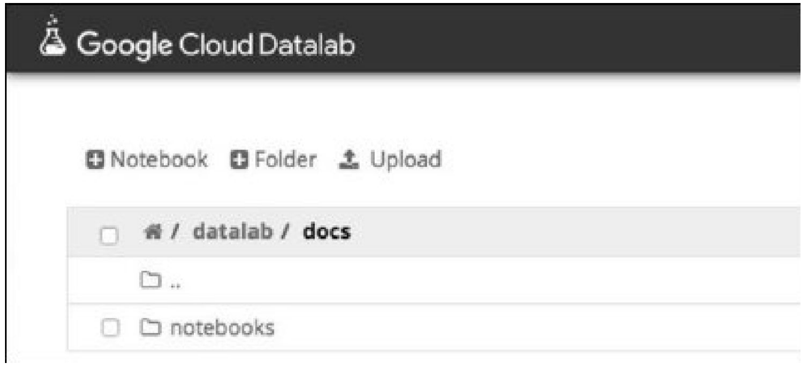
- Athena query editor allows you to define the schema for your data and the method to be used to extract the data from the S3 source.
- Athena treats data in s3 as **read-only**, and all transformations are made in the internal athena engine and storage.
- Visualize results with other tools such as **AMAZON QUICKSIGHT**



Google Cloud Datalab

- The Google Cloud provides a data analytics tool called **BigQuery.**
- Google's latest addition to **BigQuery is a Jupyter-based tool called Datalab.**
- Datalab home screen.
- Inside docs is a directory called **notebooks** that contains many great tutorials and samples





Rubella in Washington and Indiana	Looking for Weather Station Anomalies
Data Set: Google BigQuery archive contains a Centers for Disease Control and Prevention (CDC) dataset concerning diseases reported by state and city over a long period **Tools Used:**Python statement that captures its result as a Pandas **DataFrame** and pulls apart the time stamp fields and data values	**Dataset:** the Global Summary of the Day (GSOD) weather for 9,000 weather stations between 1929 and 2016. **Tools Used: Cloud Datalab mapping functions**

Machine Learning in the Cloud

- Machine learning makes **intelligent machines or software**, and on the other hand, **cloud computing provides storage and security** to access these applications

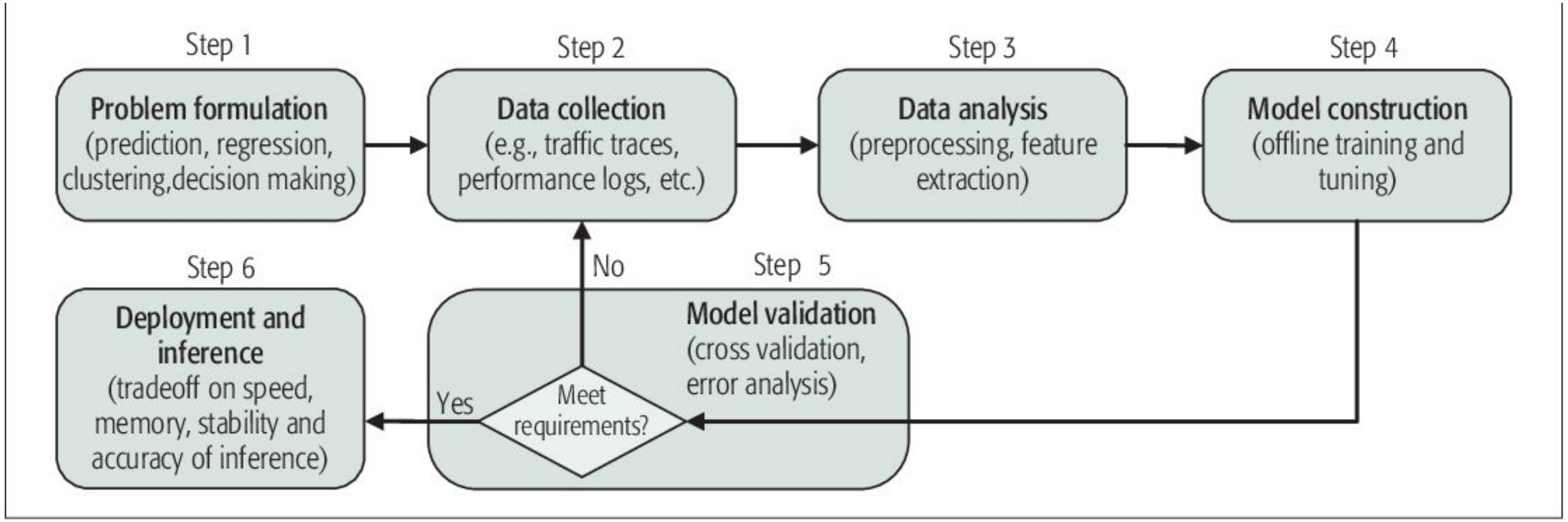
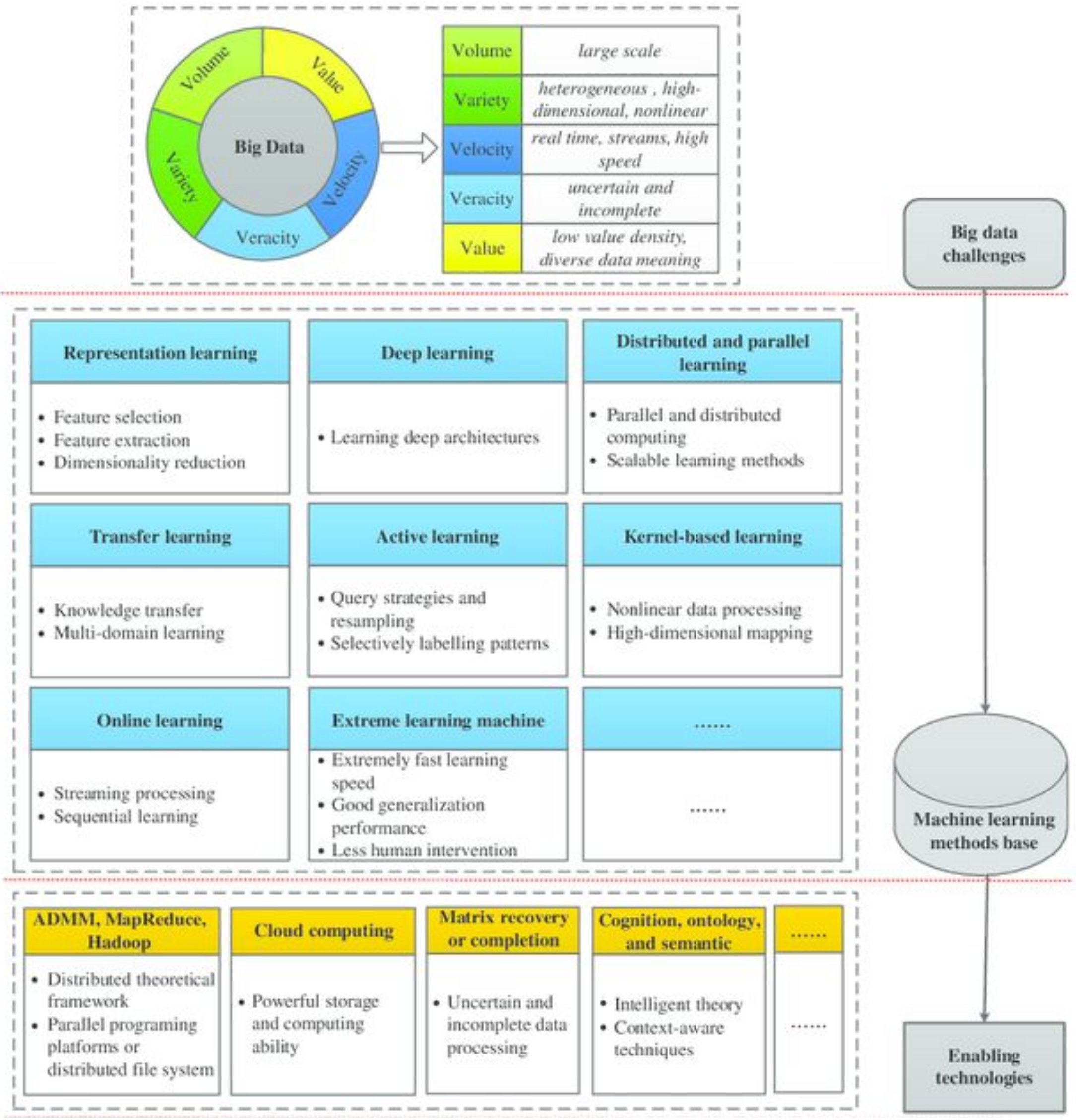
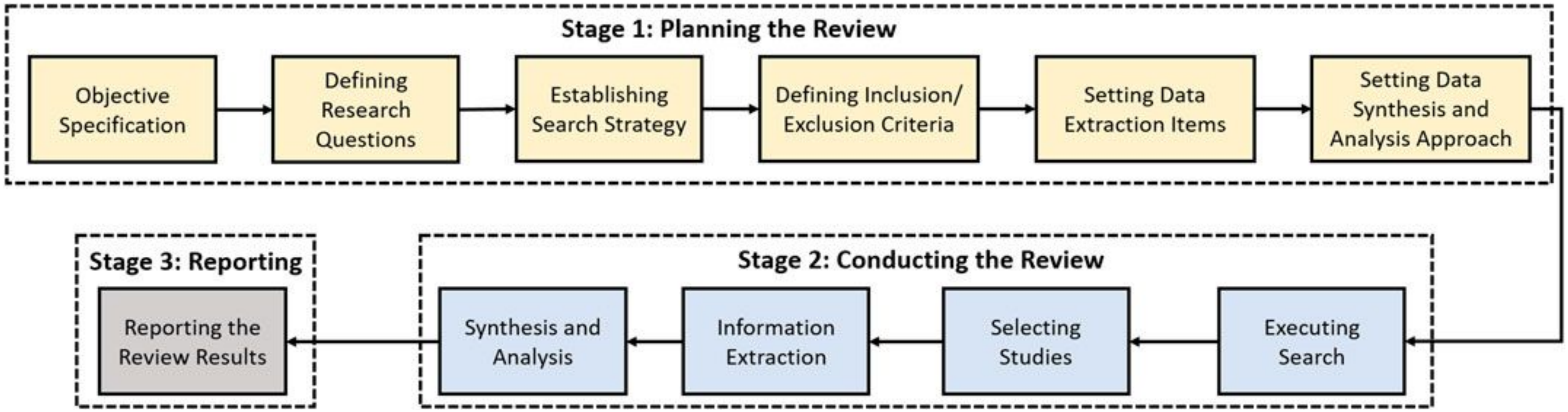


FIGURE 1. The typical workflow of machine learning for networking.



Big Data
Volume
Value
Velocity
Veracity
Variety
Volume
large scale
Variety
heterogeneous , high-dimensional, nonlinear
Velocity
real time, streams, high speed
Veracity
uncertain and incomplete
Value
low value density, diverse data meaning
Big data challenges
Representation learning
• Feature selection
• Feature extraction
• Dimensionality reduction
Deep learning
• Learning deep architectures
Distributed and parallel learning
• Parallel and distributed computing
• Scalable learning methods
Transfer learning
• Knowledge transfer
• Multi-domain learning
Active learning
• Query strategies and resampling
• Selectively labelling patterns
Kernel-based learning
• Nonlinear data processing
• High-dimensional mapping
Online learning
• Streaming processing
• Sequential learning
Extreme learning machine
• Extremely fast learning speed
• Good generalization performance
• Less human intervention
......
......
Machine learning methods base
ADMM, MapReduce, Hadoop
• Distributed theoretical framework
• Parallel programing platforms or distributed file system
Cloud computing
• Powerful storage and computing ability
Matrix recovery or completion
• Uncertain and incomplete data processing
Cognition, ontology, and semantic
• Intelligent theory
• Context-aware techniques
......
......
Enabling technologies

cloud-based ML or machine learning as a service (MLaaS) architecture

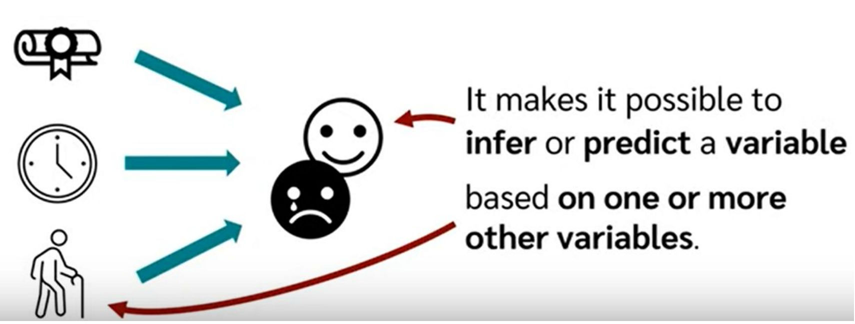


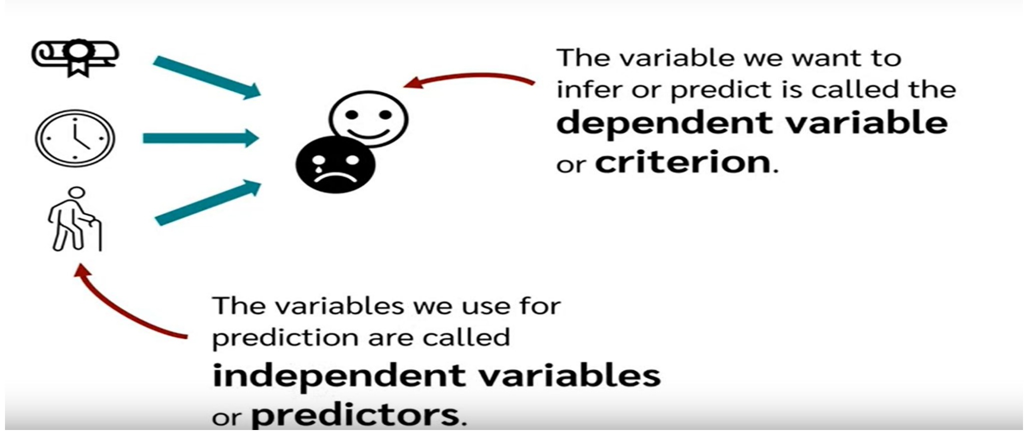
Spark Machine Learning Library (MLlib)

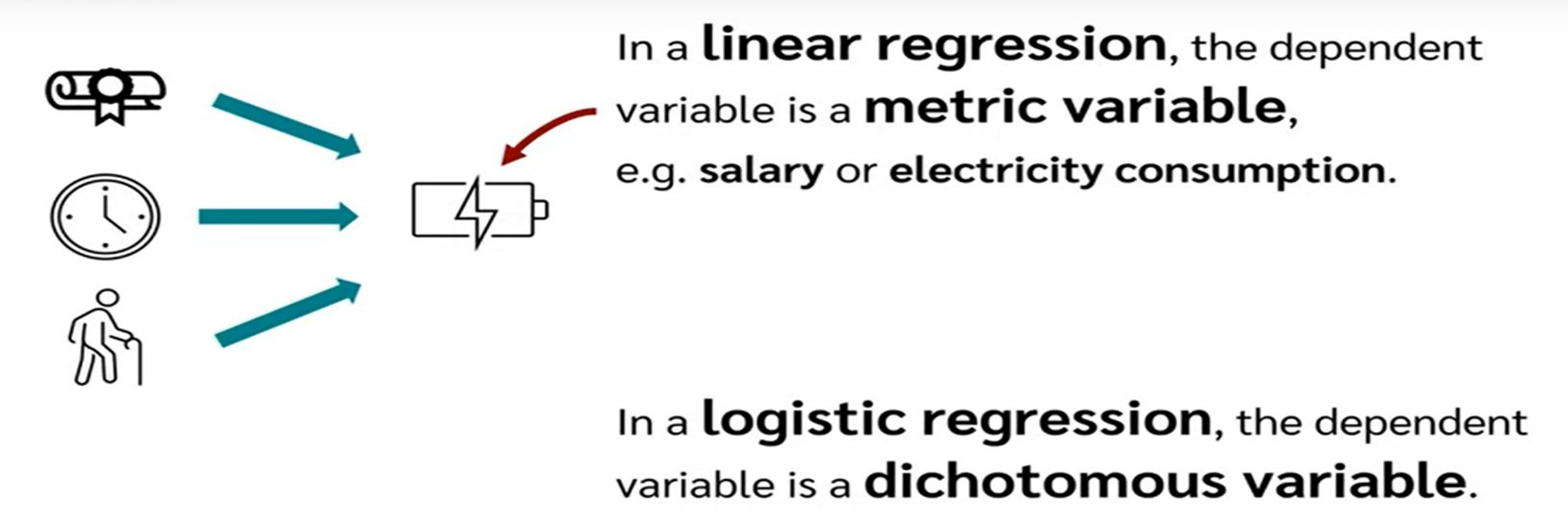
DataFrames	• **Dataframes** are containers **created from spark rdds** to hold vectors and other structured types in a manner that permits efficient execution . • Spark dataframes are similar to pandas dataframes and share some operations. • They are distributed objects that are part of the execution graph. • Can convert them to pandas dataframes to access them in python
Transformers	• Transformers are operators that **convert one DataFrame to another**. • Since they are nodes on the execution graph, they are not evaluated until the **entire graph is executed**.
Estimators	• Estimators **encapsulate ML and other algorithms**. • you can use the **fit(...) method to pass a DataFrame** and parameters **to a learning algorithm to create a model.** • The model is now represented as a Transformer.
Pipeline	• A Pipeline (usually linear, but can be a **directed acyclic graph**) links Transformers and Estimators to specify an ML workflow. • Pipelines inherit **the fit(...) method from the contained estimator**. • Once the estimator is trained, the pipeline is a model and has a **transform(...) method** that can be used to **push new cases through the pipeline to make predictions.**

Regression

- Modelling relationships between variables







Dichotomous variables are variables with only **two values.**

For example:

Whether a person **buys** or does **not buy** a particular product

or

whether a disease is **present** or **not**

Y=predicted output (dependent variable

B1,b2coefficients (weights) for the input features

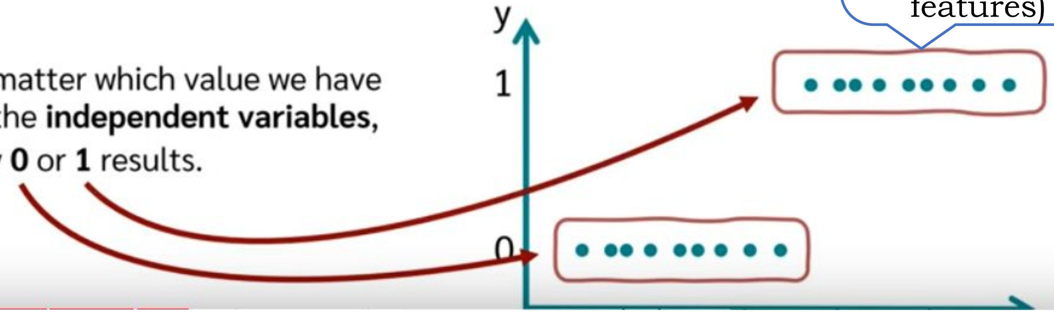
A=intercept (constant term)

$$\boxed{\hat{y}} = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a$$

X1,X2 independent variables (input features)

However, we now have a **dependent variable** that is either **0** or **1**.

No matter which value we have for the **independent variables**, only **0** or **1** results.



For **z**, the equation of the **linear regression** is now simply used.

$$\hat{y} = \boxed{b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a}$$

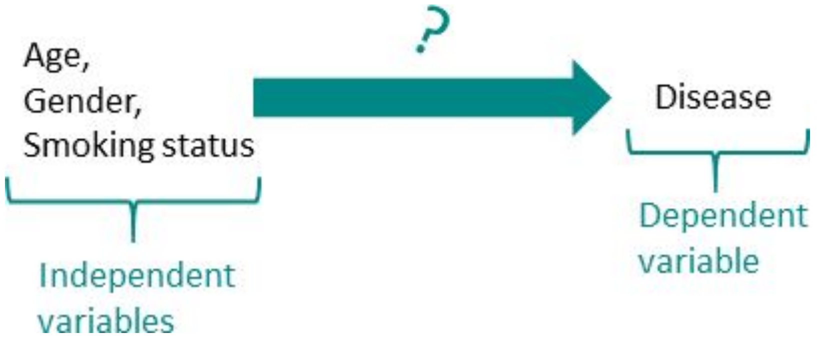
$$f(z) = \frac{1}{1 + e^{\boxed{-z}}} = \frac{1}{1 + e^{-\boxed{(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}}$$

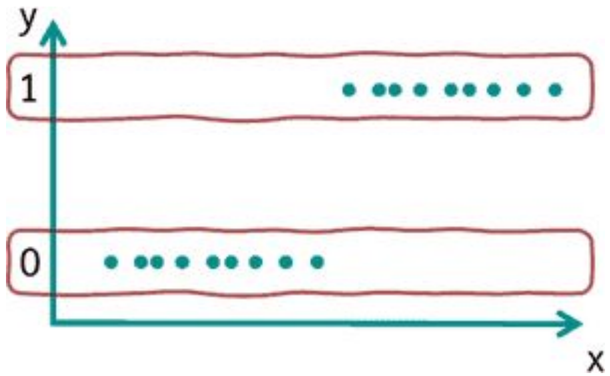
This gives us this **equation.**

binary classification problems. It models the probability that a given input belongs to a particular class (e.g., 0 or 1)

Logistic Regression

- In the basic form of logistic regression, dichotomous variables (0 or 1) can be predicted. For this purpose, the probability of the occurrence of value 1 (=characteristic present) is estimated.
- **In medicine,** for example, a frequent application is to find out which variables have an influence on a disease. In this case, 0 could stand for not diseased and 1 for diseased.
- Subsequently, the influence of age, gender and smoking status (smoker or not) on this particular disease could be examined.





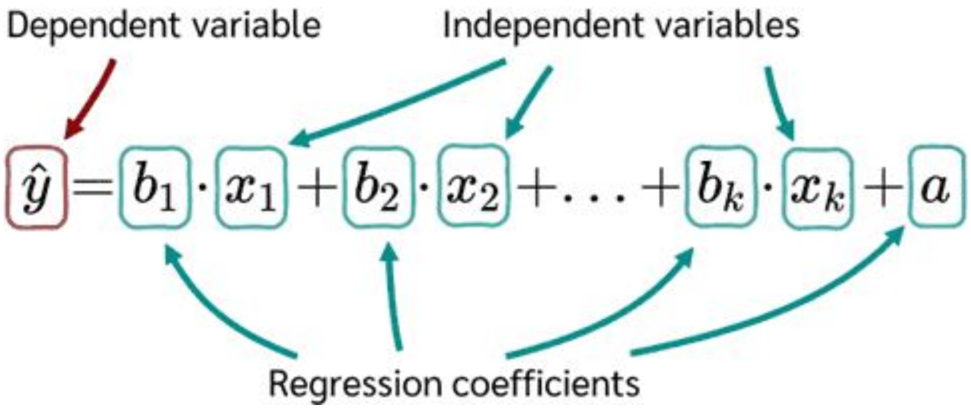
Logistic regression and probabilities

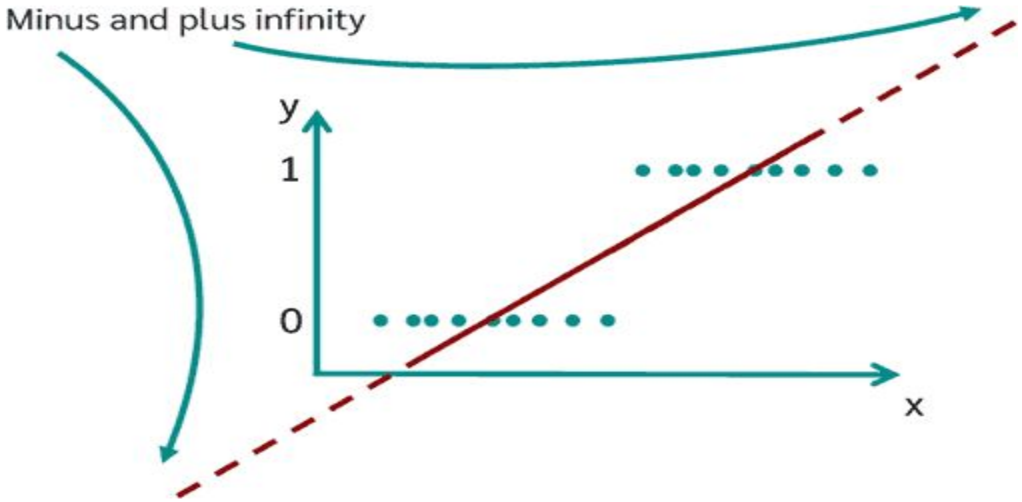
- In linear regression, the *independent variables (e.g., age and gender) are used to estimate the specific value of the dependent variable (e.g., body weight).*

- In logistic regression, on the other hand, the **dependent variable is dichotomous (0 or 1)** and **the probability that expression 1 occurs is estimated**.
- Returning to the example above, this means: How likely is it that the disease is present if the person under consideration has a certain age and smoking status.

Calculate logistic regression

- To build a logistic regression model, the linear regression equation is used as the starting point.





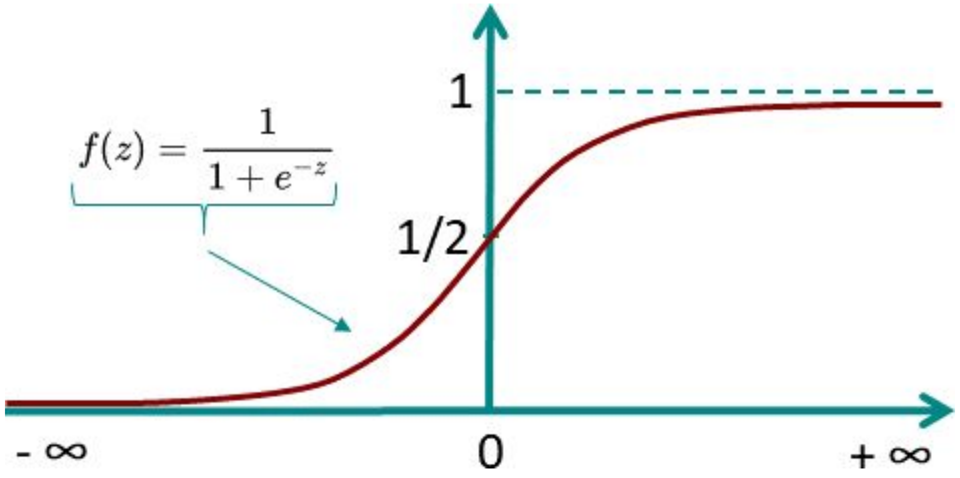
- As can be seen in the graph, however, values between plus and minus infinity can now occur.
- The goal of logistic regression, however, is **to estimate the probability of occurrence and not the value of the variable itself.**
- Therefore, the this equation must still be transformed
- To do this, it is necessary to restrict the value range for the prediction to the range **between 0 and 1.**
- To ensure that only values between 0 and 1 are possible, **the logistic function f** is used.

Logistic Regression

- Logistic regression is a Machine Learning classification algorithm that is used to **predict the probability of certain classes based on some dependent variables**.
- Logistic Regression is a classification algorithm for categorical variables like Yes/No, True/False, 0/1, etc.
- *Sage Maker is a fully managed machine learning (ML) service with built-in algorithms for linear regression and logistic regression, among several other statistical software packages.*
- *Every data scientist can use SageMaker to prepare, **build, train, and deploy logistic regression models quickly.***
- *SageMaker removes the heavy lifting from each step of the logistic regression process to make it easier to develop high-quality models.*

Logistic function

- The logistic model is based on the logical function. The special thing about the **logistic function is that for values between minus and plus infinity**, it always assumes only values between 0 and 1.



- So the logistic function is perfect to describe the **probability *P(y=1/x)***.
- If the logistic function is now applied to the upper regression equation the result I

$$\hat{y} = \boxed{b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_k \cdot x_k + a}$$

$$f(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-\boxed{(b_1 \cdot x_1 + \ldots + b_k \cdot x_k + a)}}}$$

- This now ensures that no matter in which range the *x* values are located, only values between 0 and 1 will come out.

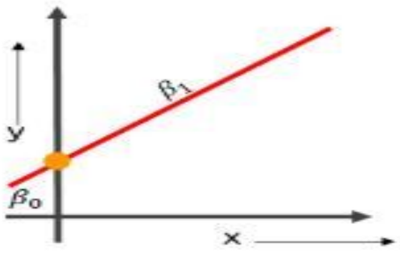
Scenario Based LR

- **Scenario:**An organization wants to determine an employee's salary increase based on their performance.
- For this purpose, a linear regression algorithm will help them decide.
- Plotting a regression line by considering the employee's **performance as the independent variable, and the salary increase as the dependent variable** will make their task easier.

- As such, we clip the line at zero and one, and **convert it into a sigmoid curve (S curve).**
- Based on the **threshold values**, the organization can decide whether an employee will get a salary increase or not.

- Based on the threshold values, the organization can decide whether an **employee will get a salary increase or not.**
  - **Odds (θ) = Probability of an event happening / Probability of an event not happening**
  - **θ = p / 1 - p**
- The values of odds range from **zero to ∞**
- **T**he values of probability lies **between zero and one**.

https://www.simplilearn.com/tutorials/machine-learning-tutorial/logistic-regression-in-python

Consider the equation of a straight line: ***y* = *β*0 + *β*1* *x*** Here, **$\beta 0$ is the y-intercept** **$\beta 1$ is the slope of the line** **x is the value of the x coordinate** **y is the value of the prediction**	
Now to predict the odds of success, we use the following formula	$\log\left(\frac{p(x)}{1-P(x)}\right) = \beta_0 + \beta_1 x$
Exponentiating both the sides, we have	$e^{ln\left(\frac{p(x)}{1-p(x)}\right)} = e^{\beta_0+\beta_1 x}$ $\left(\frac{p(x)}{1-p(x)}\right) = e^{\beta_0+\beta_1 x}$
Let Y = e $\beta 0+\beta 1$ * x Then p(x) / 1 - p(x) = Y p(x) = Y(1 - p(x)) p(x) = Y - Y(p(x)) p(x) + Y(p(x)) = Y p(x)(1+Y) = Y p(x) = Y / 1+Y	$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$ 

- In short, the logistic regression model computes a **sum of the input features** (in most cases, there is a bias term), and **calculates the logistic of the result.**
- Suppose we have a set of feature vectors **xi ∈ Rn for i in [0, m].**
- Associated with each **feature vector is a binary outcome yi.**
- We are interested in the conditional probability **P(y = 1|x), which we approximate by a function p(x).**
- Because p(x) is between 0 and 1, it is not expressible as a linear function of x, and thus we cannot use regular linear regression.
- Instead, we look at the **"odds" expression p(x)/(1 − p(x)) and guess that its log is linear.**

$$ln\left(\frac{p(x)}{1-p(x)}\right) = b_0 + b \cdot x.$$

- where the offset b0 and the vector b = [b1, b2, ...bn] define a hyperplane for linear regression.
- Solving this expression for p(x) we obtain:

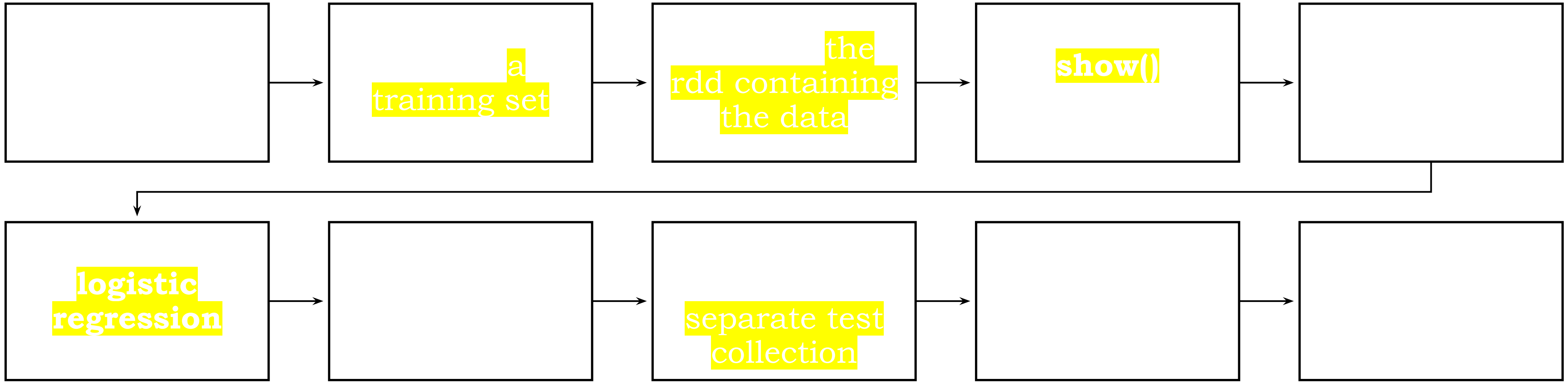
$$p(x) = \frac{1}{1+e^{-(b_0+b\cdot x)}}$$

- We then predict y = 1 if p(x) > 0 and zero otherwise.
- Unfortunately, finding the best b0 and b is not as easy as in the case of linear regression.
- However, **simple Newton-like iterations converge to good solutions** if we have a sample of the feature vectors and known outcomes

$$\sigma(t) = \frac{e^t}{e^t+1} = \frac{1}{1+e^{-t}}$$

Chicago Restaurant Example

- To illustrate the use of Spark MLlib, we apply it to an example from the **Azure HDInsight** tutorial namely predicting whether restaurants pass or fail health inspections based on the free text of an inspector’s comments



Read the data from blob storage csv parse that takes each line in the CSV file and parses it using python's csv.Reader() function	**csv.Reader()** 1
Create a **training set** from a set of inspection reports that contain outcomes, for use in fitting our **logistic regression model.** 3 **Convert the rdd containing the data**, inspections, to create a dataframe, df, with four fields: ***record id, restaurant name, inspection result, and any recorded violations***	inspections = **spark.sparkContext.textFile**(\ 2 'wasb:///HdiSamples/HdiSamples/FoodInspectionDataFood_Inspections1.csv').map(csvParse) / **Read a text file from HDFS, a local file system (available on all nodes), or any Hadoop-supported file system URI, and return it as an RDD of Strings.** schema = StructType([StructField("id",**IntegerType(), False), StructField("name",StringType(), False),StructField("results",StringType(), False),StructField("violations",StringType(), True)])** df = **spark.createDataFrame**(inspections.map(\ Creates a DataFrame from an RDD 4 lambda l: (int(l[0]), l[2], l[3], l[4])), schema) df.registerTempTable('**CountResults**') 5 **create a dataframe (df) and a temporary table (CountResults) with a few columns that are useful for the predictive analysis**

show() function to return values to the Python environment.

6

to get a small sample of the data

```
df.show(5)
+-------+--------------------+---------------+--------------------+
|     id|                name|        results|          violations|
+-------+--------------------+---------------+--------------------+
|1978294|KENTUCKY FRIED CH...|           Pass|32. FOOD AND NON-...|
|1978279|          SOLO FOODS|Out of Business|                    |
|1978275|SHARKS FISH & CHI...|           Pass|34. FLOORS: CONST...|
|1978268|CARNITAS Y SUPERM...|           Pass|33. FOOD AND NON-...|
|1978261|           WINGSTOP|           Pass|                    |
+-------+--------------------+---------------+--------------------+
only showing top 5 rows
```

DataFrame operations to count the **passing and failing grades**

7

```
print("Passing = %d"%df[df.results == 'Pass'].count())
print("Failing = %d"%df[df.results == 'Fail'].count())
```

Passing = 61204
Failing = 20225

To train a logistic regression model we need a **DataFrame with a binary label and feature vector for each record.**

8

We do not want to use records associated with “out of business” or other special cases, so we **map “Pass” and “Pass with conditions” to 1**, “Fail” to 0, and all others to -1, which we filter out

```
def labelForResults(s):
if s == 'Fail':
return 0.0
elseif s == 'Pass w/ Conditions' or s == 'Pass':
return 1.0
else:
return -1.0
```

9

THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

```
label = UserDefinedFunction(labelForResults,DoubleType())
labeledData =df.select(label(df.results).alias('label'), \df.violations).where('label >= 0')
```

to show one row of the labeled data

10

DataFrame with two columns, **label and violations** and to create and **run the Spark Mllib** pipeline that we will use **to train our logistic regression model**

11

\# 1) Define **pipeline components**

\# a) Tokenize 'violations' and place result in new column 'words' **"tokenize" each violations string to get the individual words in each string**

12

\# b) Hash 'words' to create new column of 'features' **HashingTF to convert each set of tokens into a feature vector that can then be passed to the logistic regression algorithm to construct a model.**

13

14

\# c) Create instance of logistic regression

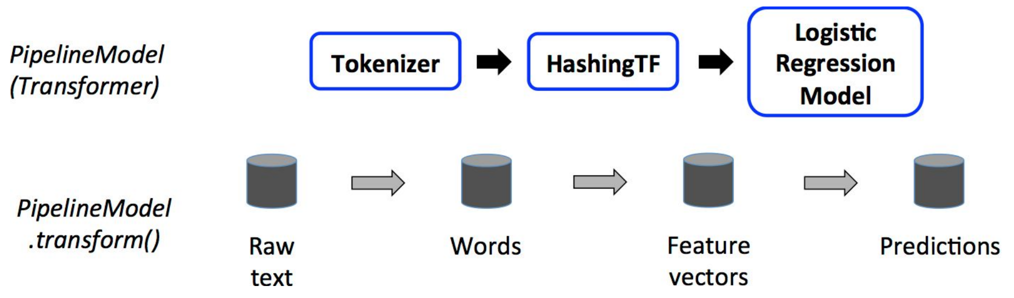
\# 2) Construct pipeline: tokenize, hash, logistic Regression

15

\# 3) Run pipeline to create model

```
tokenizer = Tokenizer(inputCol="violations", outputCol="words")
hashingTF = HashingTF(inputCol="words", outputCol ="features")
lr = LogisticRegression(maxIter=10, regParam=0.01)
pipeline = Pipeline(stages=[tokenizer, hashingTF,lr])
model = pipeline.fit(labeledData)
```

Want to point all these steps



test the model with a separate test collection 16 To create a new dataframe, **predictionsdf** that contains the prediction generated by the model. The snippet also creates a temporary table called predictions based on the dataframe. 17 **Model.transform()** method applies the same transformation to any new data with the same schema, and arrive at a prediction of how to classify the data 18	testData = **spark.sparkContext.textFile**('/data_path/Food_Inspections2.csv')\ .map(csvParse) \ .**map**(lambda l: (int(l[0]), l[2], l[3], l[4])) testDf = **spark.createDataFrame**(testData, schema). where("results = 'Fail' OR results = 'Pass' OR \ results = 'Pass w/ Conditions'") **predictionsDf = model.transform(testDf)**
logistic regression model has appended several new columns to the data frame, **including one called prediction** 19	numSuccesses = **predictionsDf.where**(\ """(prediction = 0 AND results = 'Fail') OR \ (prediction = 1 AND (results = 'Pass' OR \ results = 'Pass w/ Conditions'))""").count() numInspections = **predictionsDf.count()** print("There were %d inspections and there were %d predictions"\ %(numInspections,numSuccesses)) print("This is a %2.2f sucess rate"\%(float(numSuccesses) / float(numInspections) * 100))

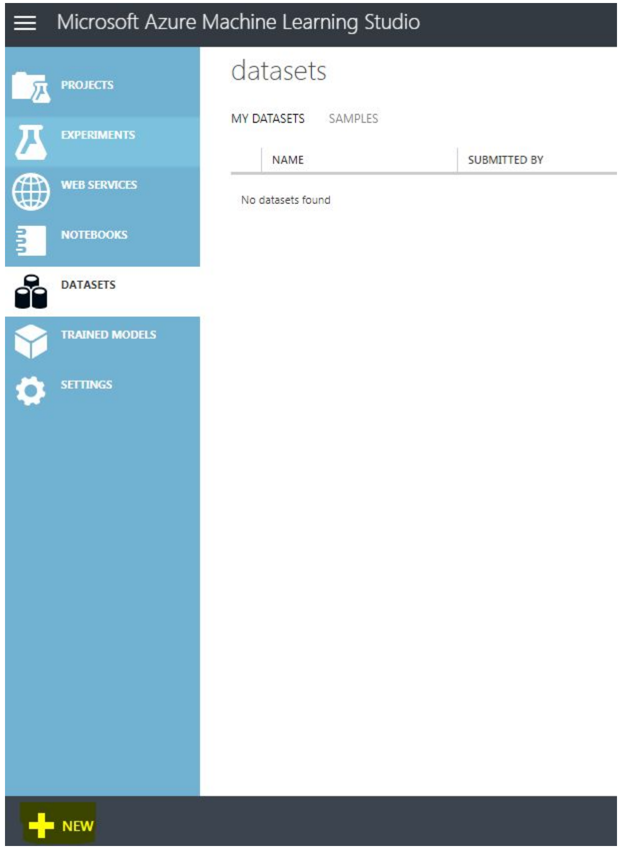
There were **30694 inspections and there were 27774** predictions This is a 90.49\% success rate 20	• examine other measures of success, such as precision and recall, that are widely used in ML research. • When applied to our ability to **predict failure**, recall is the probability that we predicted as failing a randomly selected inspection from those with failing grades. • **recall probability is only 67%.** • Our ability to predict failure is thus well **below our ability to predict passing.** • The reason may be that other factors involved with failure are not reflected in the report

Azure Machine Learning Workspace

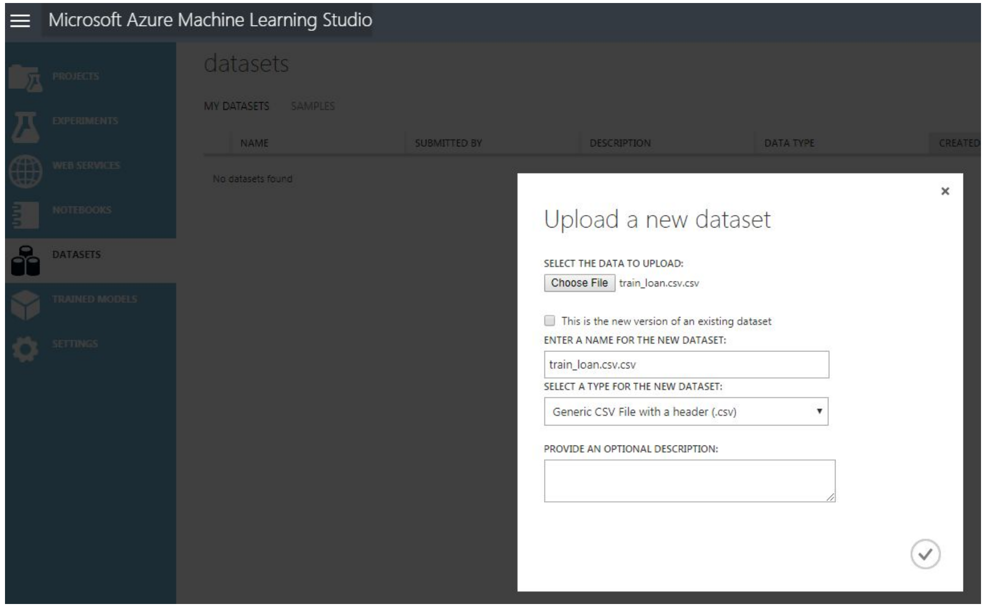
- Azure Machine Learning is a cloud portal for designing and **training machine learning cloud** services.
- It is based on a **drag-and-drop component** composition model
- In which you build a **solution to a machine learning** problem by **dragging parts of the solution from a pallet of tools and connecting them together into a workflow graph.**

Steps involved in Azure ML using (LR) for bank loan dataset

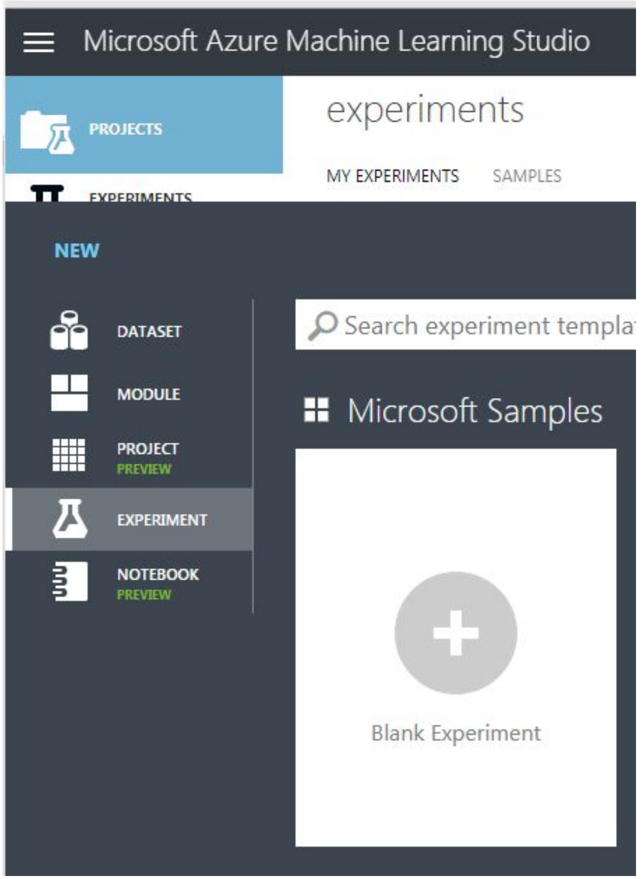
1. The **DATASET**
2. Creating a Simple **Prediction Model** for Loan Eligibility Prediction
3. Creating a **New Experiment**
4. Adding a **Dataset to the Experiment**
5. Visualizing the Data
6. Handling **Missing Values**
7. Selecting **Columns**
8. Splitting Data in **Training and Testing Set**
9. Training the **Logistic Regression in Azure ML Algorithm**
10. Evaluating the Logistic Regression in Azure ML Algorithm



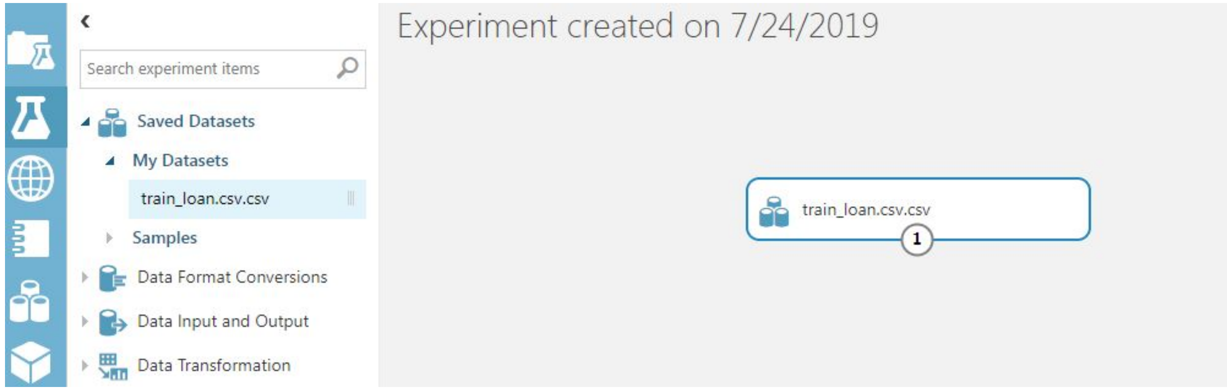
Microsoft Azure Machine Learning Studio
datasets
PROJECTS
EXPERIMENTS
WEB SERVICES
NOTEBOOKS
DATASETS
TRAINED MODELS
SETTINGS
MY DATASETS
SAMPLES
NAME
SUBMITTED BY
No datasets found
NEW

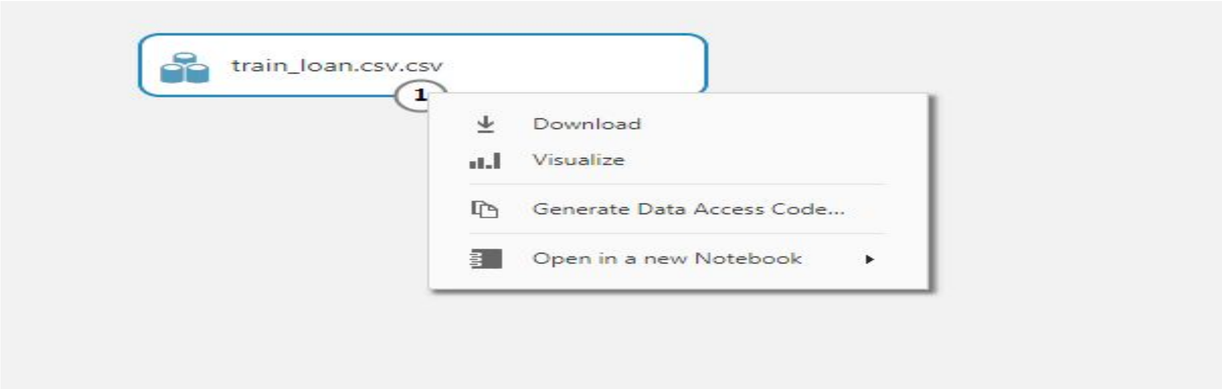


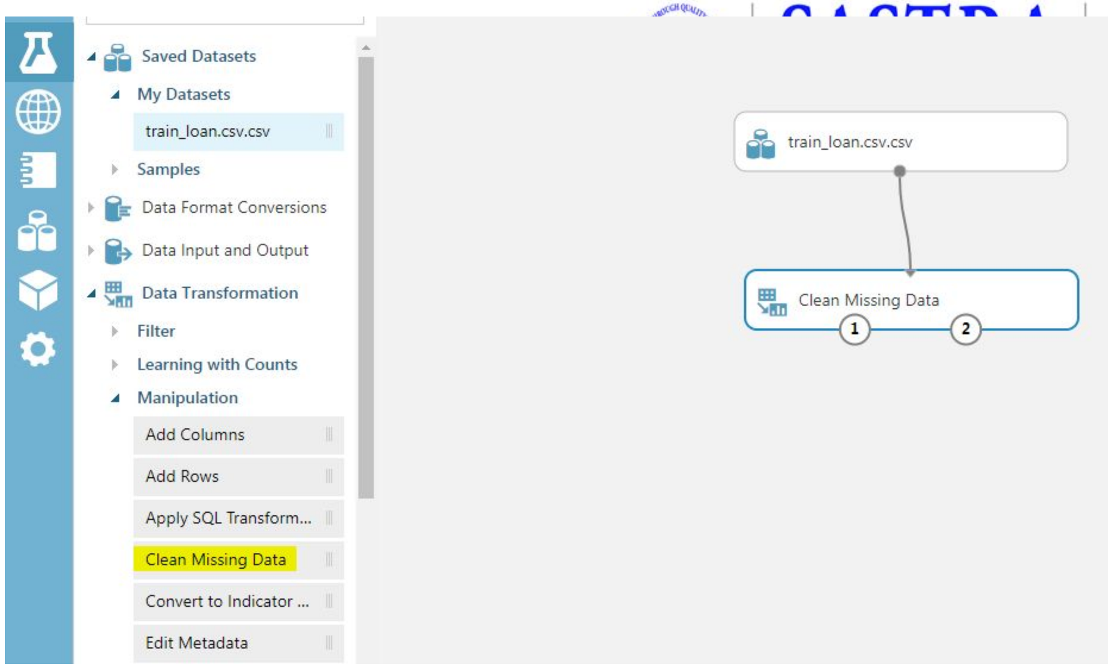
Microsoft Azure Machine Learning Studio
datasets
MY DATASETS
SAMPLES
NAME
SUBMITTED BY
DESCRIPTION
DATA TYPE
No datasets found
DATASETS
Upload a new dataset
SELECT THE DATA TO UPLOAD:
Choose File
train_loan.csv.csv
This is the new version of an existing dataset
ENTER A NAME FOR THE NEW DATASET:
train_loan.csv.csv
SELECT A TYPE FOR THE NEW DATASET:
Generic CSV File with a header (.csv)
PROVIDE AN OPTIONAL DESCRIPTION:

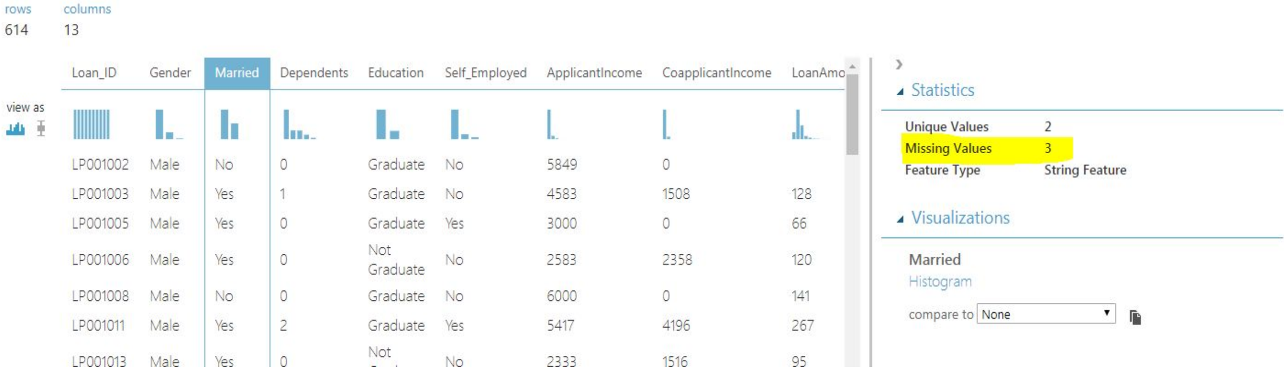


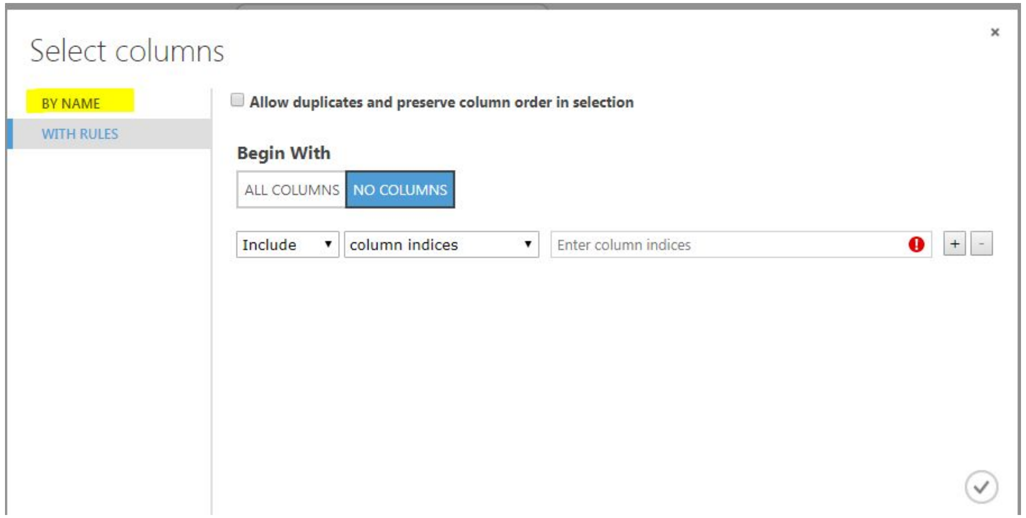
Microsoft Azure Machine Learning Studio
PROJECTS
experiments
MY EXPERIMENTS
SAMPLES
NEW
DATASET
MODULE
PROJECT
PREVIEW
EXPERIMENT
NOTEBOOK
PREVIEW
Search experiment templa
Microsoft Samples
Blank Experiment

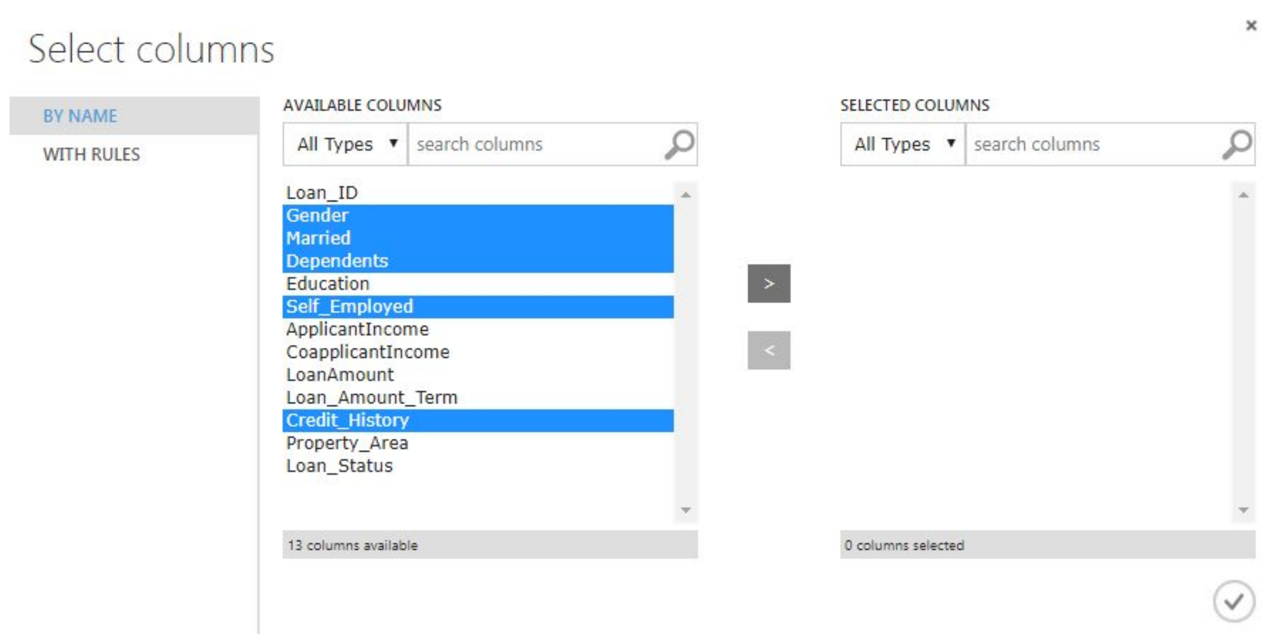

Experiment created on 7/24/2019
Search experiment items
Saved Datasets
My Datasets
train_loan.csv.csv
Samples
Data Format Conversions
Data Input and Output
Data Transformation
train_loan.csv.csv
1

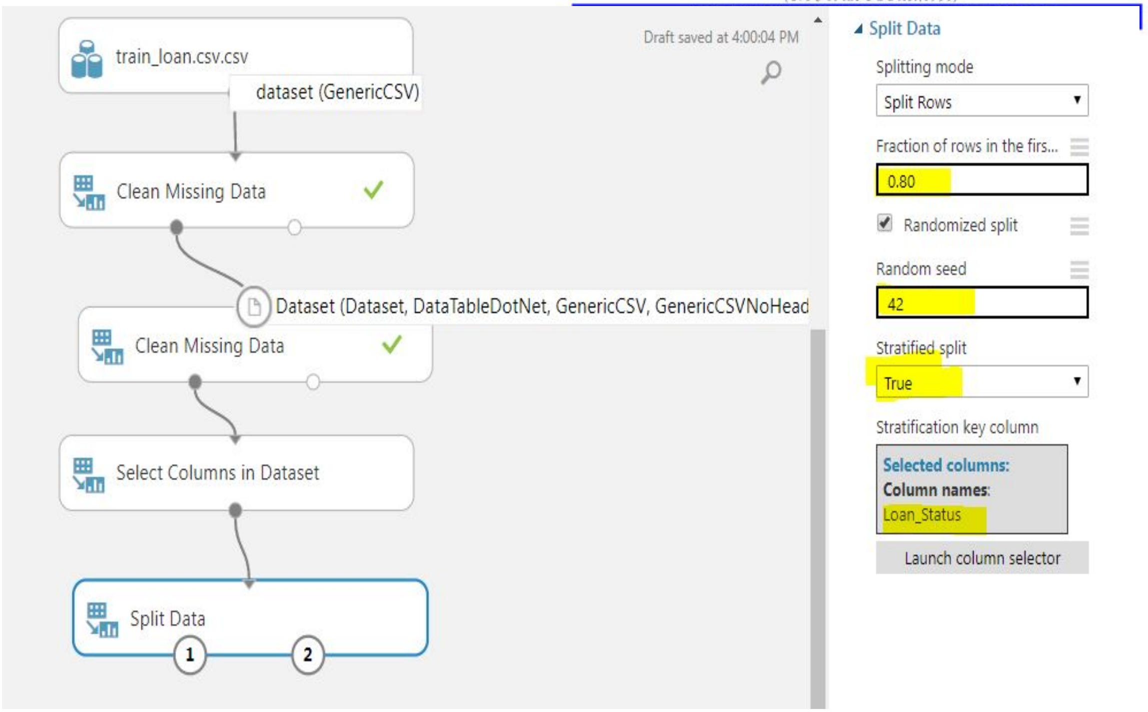

train_loan.csv.csv
1
Download
Visualize
Generate Data Access Code...
Open in a new Notebook

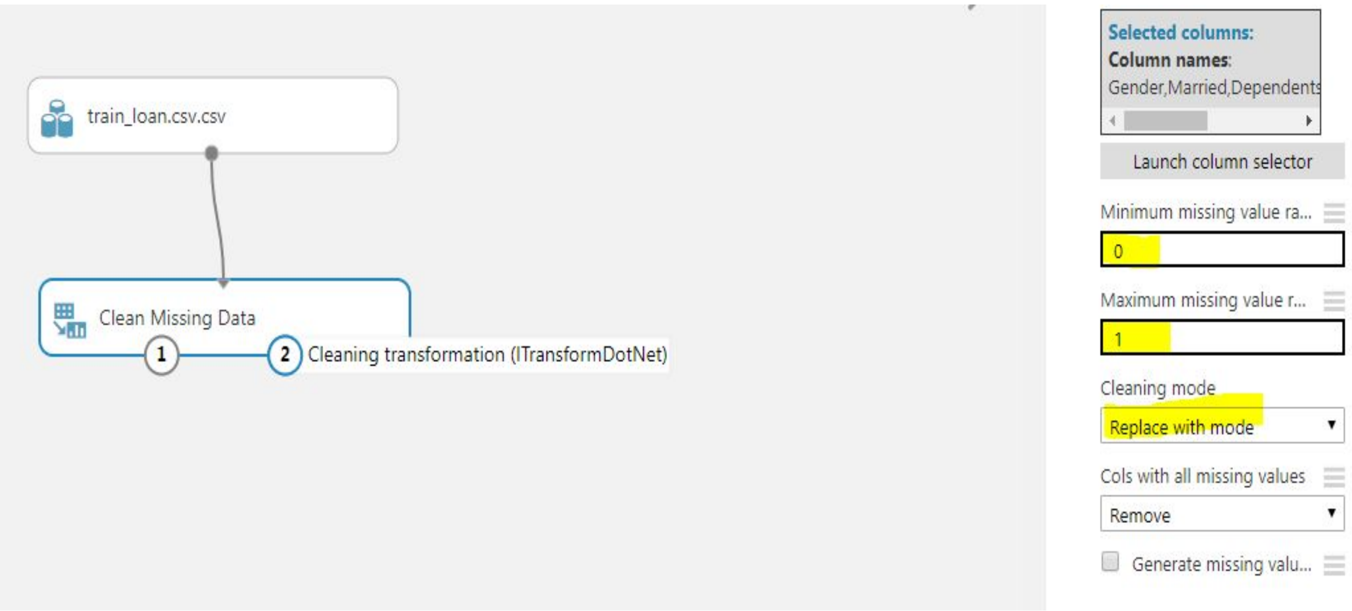

Saved Datasets
My Datasets
train_loan.csv.csv
Samples
Data Format Conversions
Data Input and Output
Data Transformation
Filter
Learning with Counts
Manipulation
Add Columns
Add Rows
Apply SQL Transform...
Clean Missing Data
Convert to Indicator ...
Edit Metadata
train_loan.csv.csv
Clean Missing Data
1
2

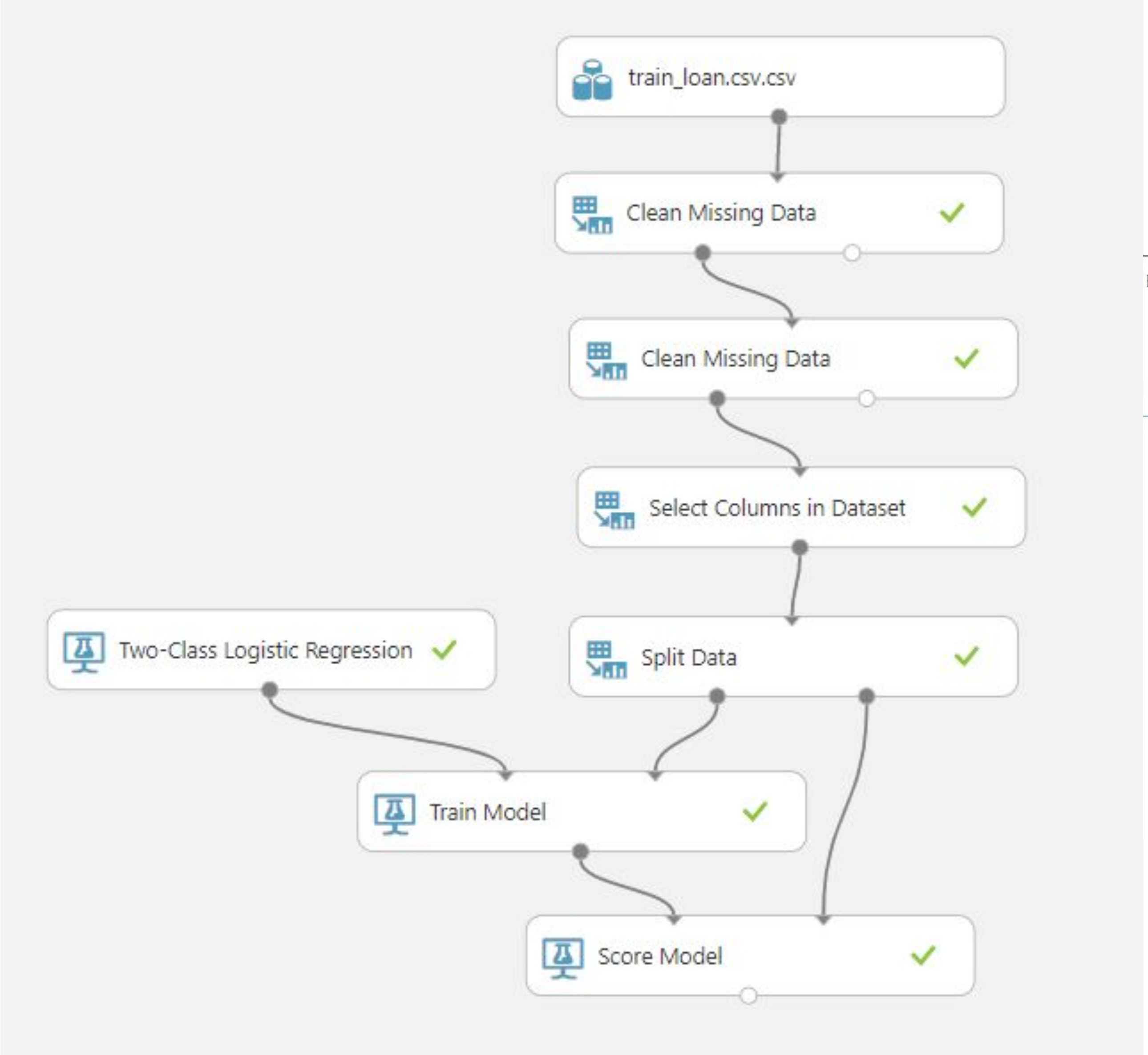

rows
614
columns
13
view as
Loan_ID
Gender
Married
Dependents
Education
Self_Employed
ApplicantIncome
CoapplicantIncome
LoanAmo
LP001002 Male No 0 Graduate No 5849 0
LP001003 Male Yes 1 Graduate No 4583 1508 128
LP001005 Male Yes 0 Graduate Yes 3000 0 66
LP001006 Male Yes 0 Not Graduate No 2583 2358 120
LP001008 Male No 0 Graduate No 6000 0 141
LP001011 Male Yes 2 Graduate Yes 5417 4196 267
LP001013 Male Yes 0 Not No 2333 1516 95
Statistics
Unique Values 2
Missing Values 3
Feature Type String Feature
Visualizations
Married
Histogram
compare to None


Select columns
BY NAME
WITH RULES
Allow duplicates and preserve column order in selection
Begin With
ALL COLUMNS
NO COLUMNS
Include
column indices
Enter column indices


Select columns
BY NAME
WITH RULES
AVAILABLE COLUMNS
All Types
search columns
Loan_ID
Gender
Married
Dependents
Education
Self_Employed
ApplicantIncome
CoapplicantIncome
LoanAmount
Loan_Amount_Term
Credit_History
Property_Area
Loan_Status
13 columns available
SELECTED COLUMNS
All Types
search columns
0 columns selected


train_loan.csv.csv
dataset (GenericCSV)
Clean Missing Data
Dataset (Dataset, DataTableDotNet, GenericCSV, GenericCSVNoHead
Clean Missing Data
Select Columns in Dataset
Split Data
Draft saved at 4:00:04 PM
Split Data
Splitting mode
Split Rows
Fraction of rows in the firs...
0.80
Randomized split
Random seed
42
Stratified split
True
Stratification key column
Selected columns:
Column names:
Loan_Status
Launch column selector

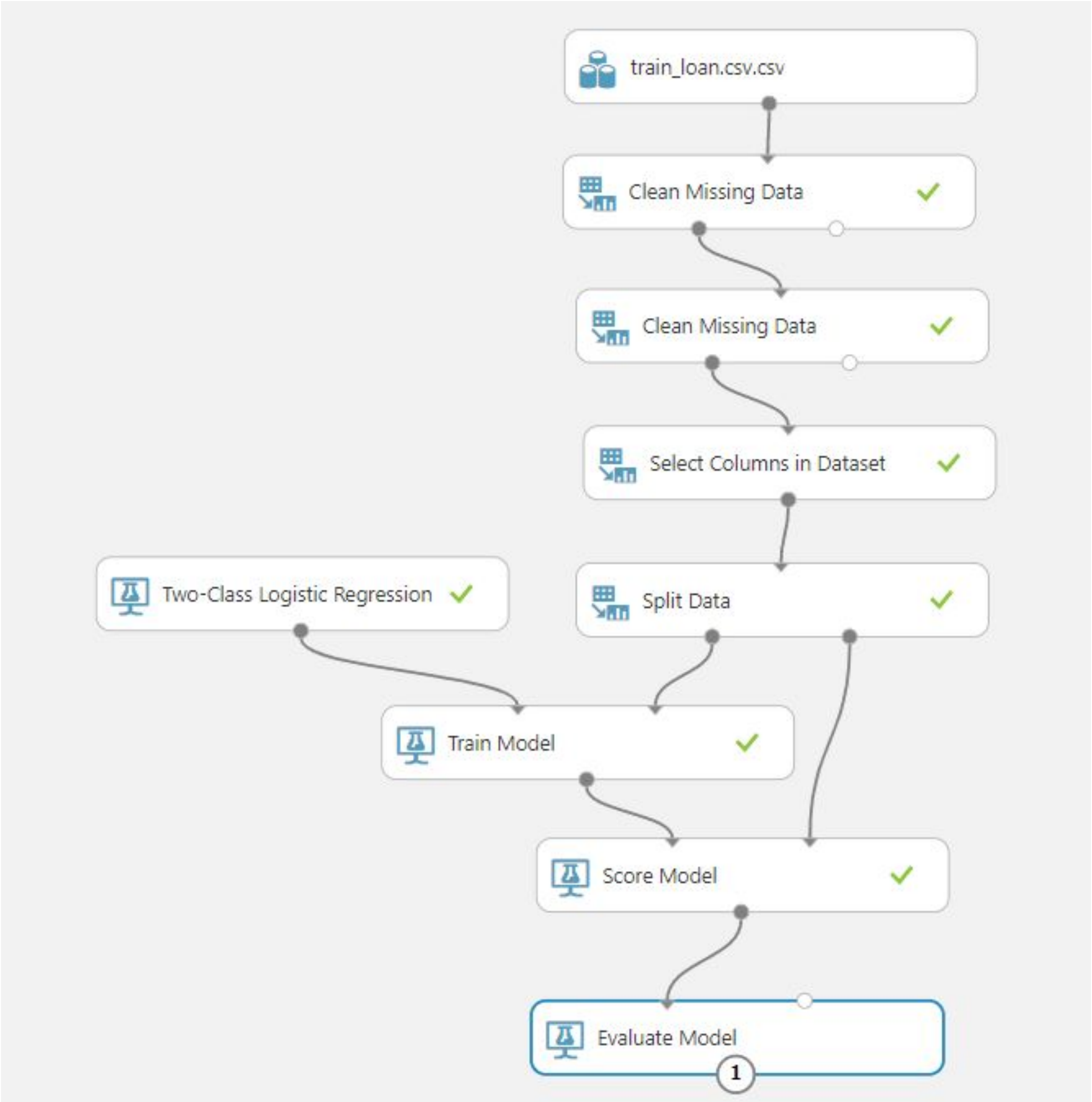

train_loan.csv.csv
Clean Missing Data
Cleaning transformation (ITransformDotNet)
Selected columns:
Column names:
Gender,Married,Dependents
Launch column selector
Minimum missing value ra...
0
Maximum missing value r...
1
Cleaning mode
Replace with mode
Cols with all missing values
Remove
Generate missing valu...



Experiment created on 7/24/2019 › Score Model › Scored dataset

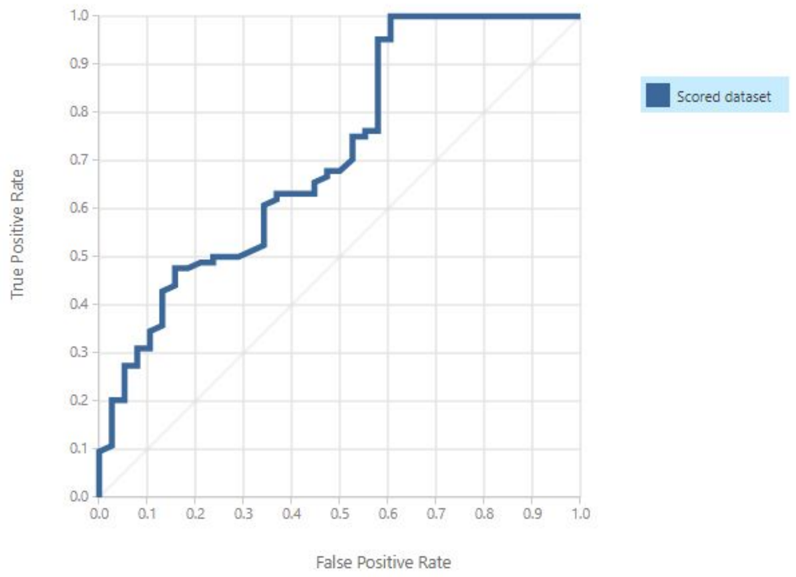
rows 122 columns 14

Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History	Property_Area	Loan_Status	Scored Labels	Scored Probabilities
No	8000	0	200	360	1	Semiurban	Y	Y	0.899406
No	3180	0	71	360	0	Urban	N	N	0.103311
No	2192	1742	45	360	1	Semiurban	Y	Y	0.855262
No	9538	0	187	360	1	Urban	Y	Y	0.764085
Yes	17263	0	225	360	1	Semiurban	Y	Y	0.720853
No	645	3683	113	480	1	Rural	Y	Y	0.668009
No	1500	1800	103	360	0	Semiurban	N	N	0.160041
No	2309	1255	125	360	0	Rural	N	N	0.110494
Yes	4053	2426	158	360	0	Urban	N	N	0.043796
Yes	5677	1424	100	360	1	Rural	Y	Y	0.802852
No	2301	985.799988	78	180	1	Urban	Y	Y	0.845482



Experiment created on 7/24/2019 › Evaluate Model › Evaluation results

ROC PRECISION/RECALL LIFT



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
84	0	0.811	0.785	0.5	0.714
False Positive	**True Negative**	**Recall**	**F1 Score**		
23	15	1.000	0.880		

Example

- Our goal is to **train a system to classify** scientific papers, based on their abstracts, into one of five categories: physics, math, computer science, biology, or finance. **(EXAMPLE DISCUSSED IN MICROSERVICES CONCEPT)**
- As training data we take a relatively small sample of abstracts from the arXiv online **library arxiv.org**.
- Existing Approach :**simple Markov chain algorithms Issues:**
- Not been able to show that they **converge to the correct probability distribution**
- Complicated direct samplers which require **extended-precision arithmetic** to evaluate numerically unstable polynomials

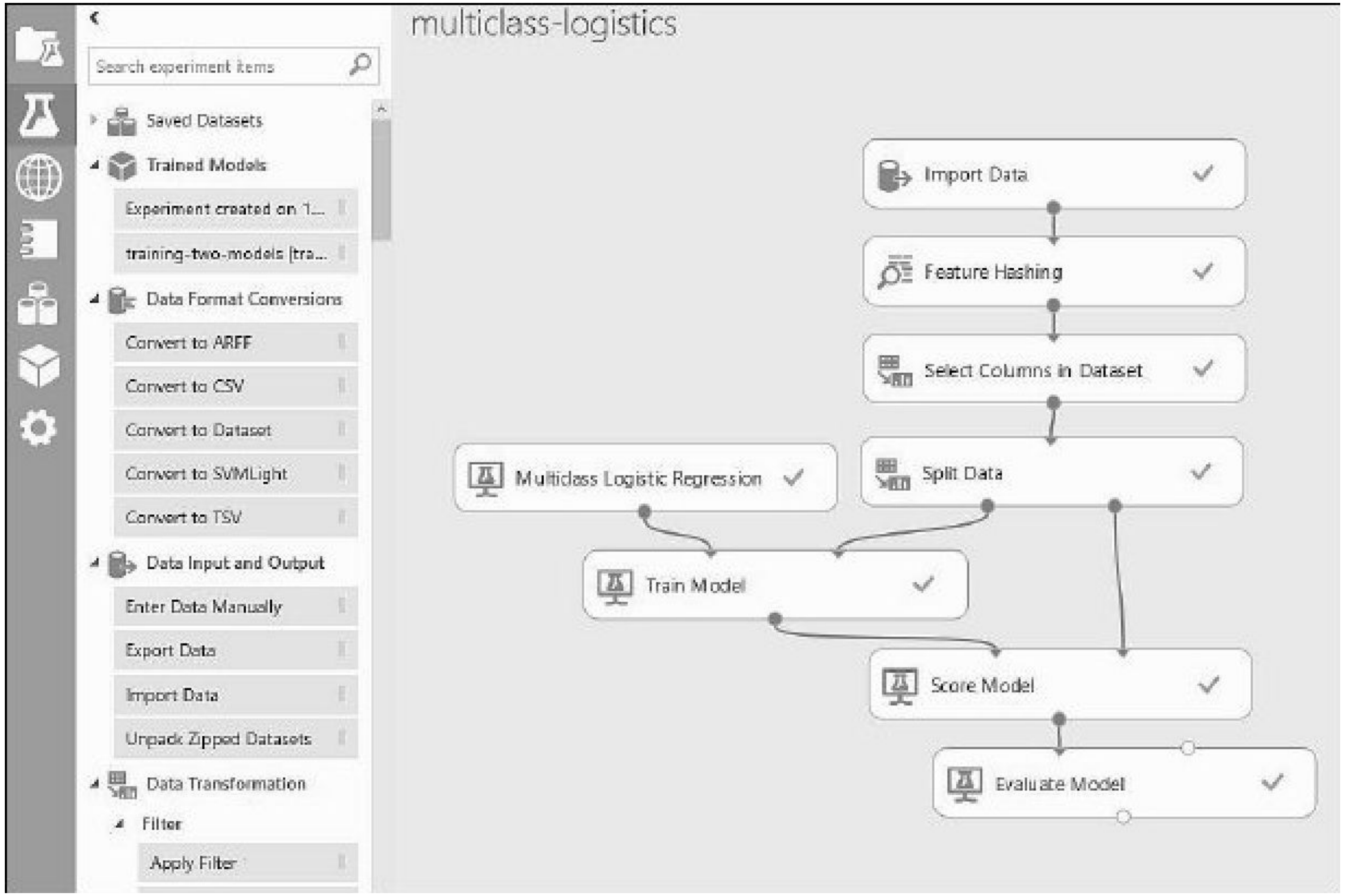
- **Solution:** Azure ML based on a multiclass version of the **logistic regression algorithm**
- Azure blob storage, where we have placed **a large subset of our arXiv samples in a CSV file.**
- Feature Hashing, builds a vectorizer based on the vocabulary in the document collection.
- **Convert each document into a numerical vector** corresponding to the key words and phrases in the document collection.
- This numeric representation is essential for the actual ml phase.
- Use the **Feature Hashing component** to transform a stream of English text into a set of integer features

- Tuple now has a **large number of columns:** class, title, abstract, and vector[0], ..., vector[n-1], **where n is the number of features.**
- To configure the algorithm, select two parameters, a **hashing bin size and anagram length.**
- The example to ml training,
- **REMOVE THE ENGLISH TEXT** of the abstract and the title, leaving only the class and the vector for each document.
- Accomplish this with a **select columns in dataset**.
- Next we split the data into two subsets:
- **a training subset and a test subset**

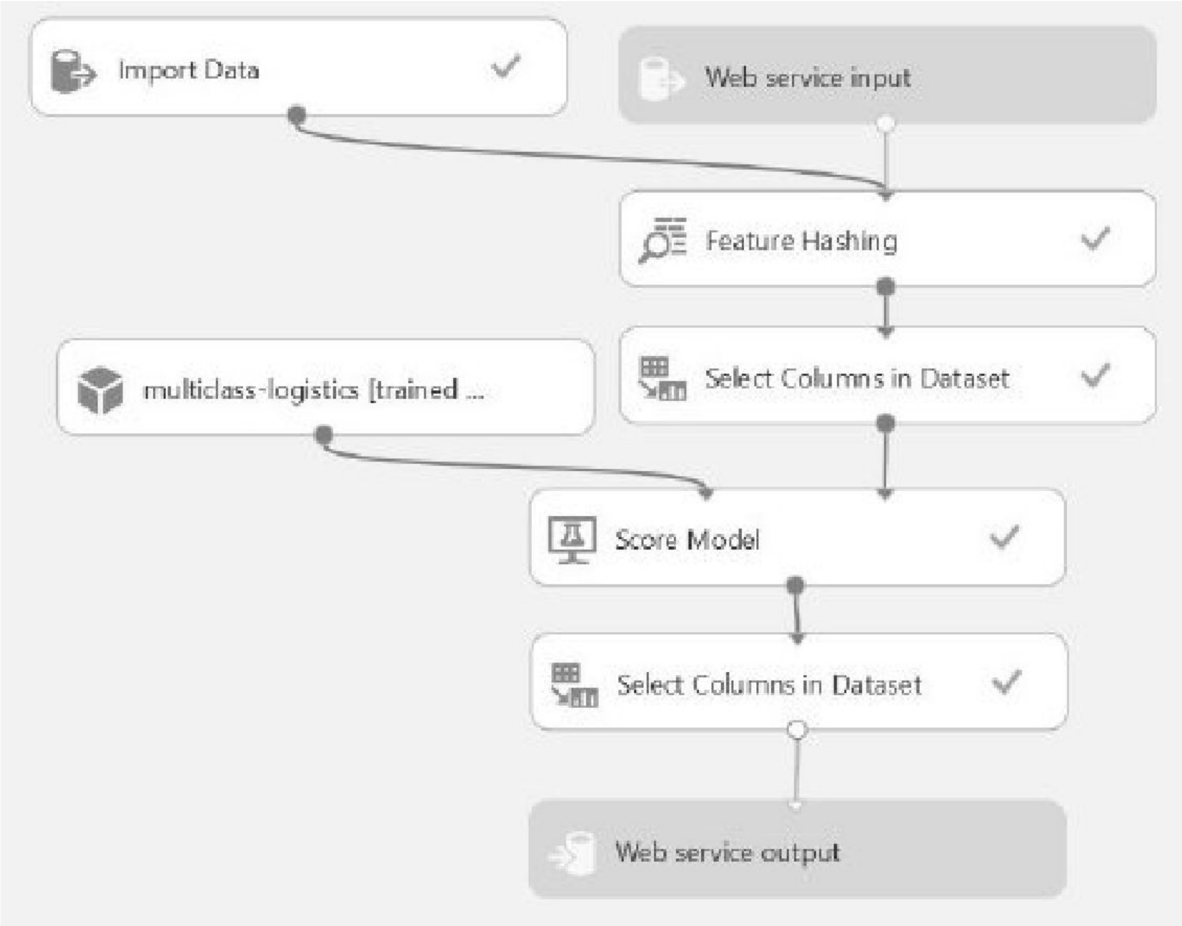
- Train Model component accepts as one input a binding to an ML method (recall this is not a dataflow graph)
- The **other input is the projected training data**.
- The output of the **Train Model task is not data for final** set but a trained model that may also be saved for later use.
- Use this **trained model to classify our test data**.
- To this end, we use the **SCORE MODEL COMPONENT**, which appends another new column to our table
- Scored Label, providing the classification predicted by the trained model for each row.
- Evaluate Model component, which **computes a confusion matrix.**

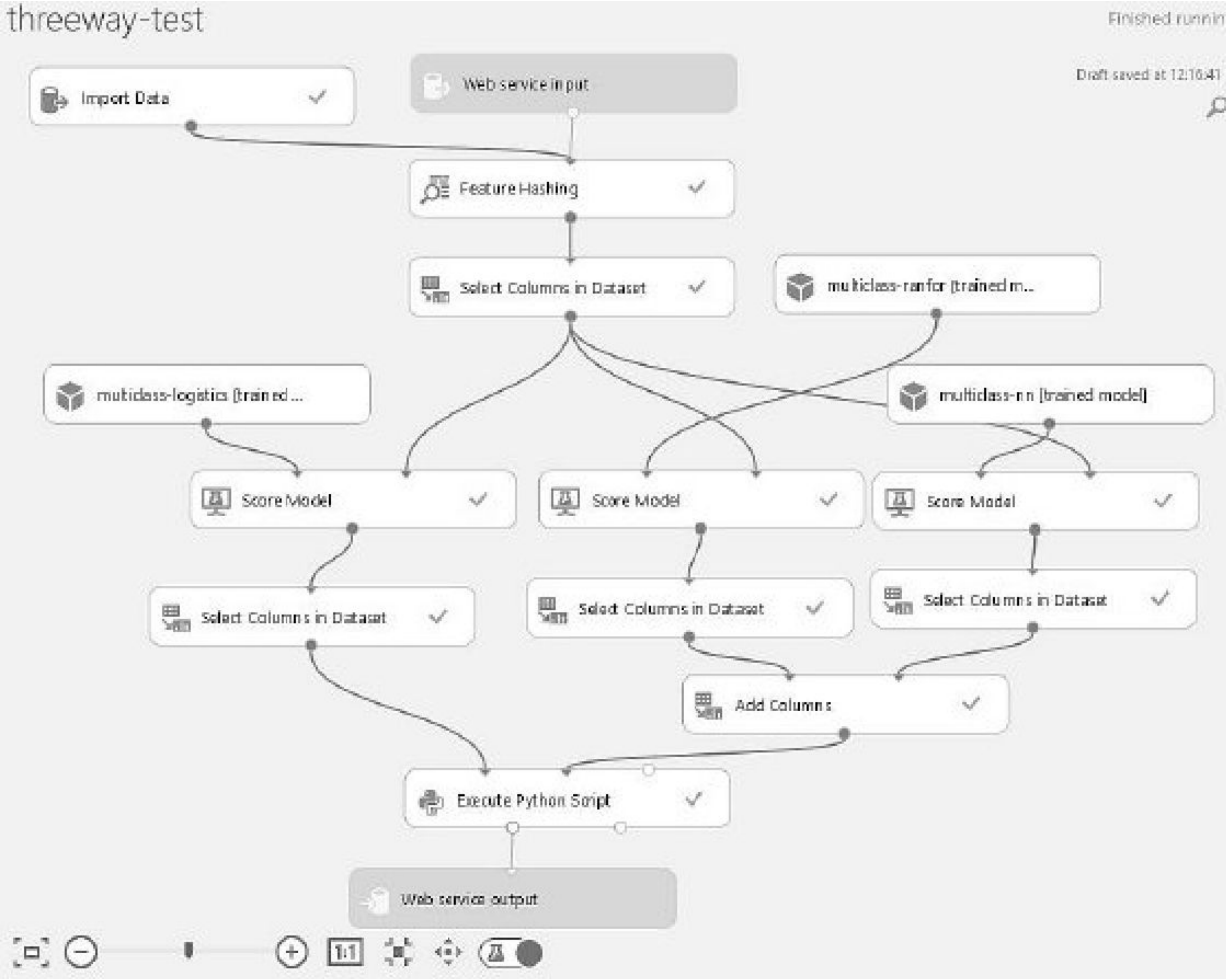
- Selecting visualize in that menu brings up useful information.
- **Set Up Web Service**.
- Azure ML portal rearranges the graph by eliminating the split-train-test parts and leaves just the feature hashing, column selection, and the scoring based on the trained model.
- Select Columns node so that we can remove the vectorized document columns from the output of the web service

Azure ML graph used to train a **multiclass logistic regression model**.



Web service graph generated by Azure ML, with an additional node to remove the vectorized document





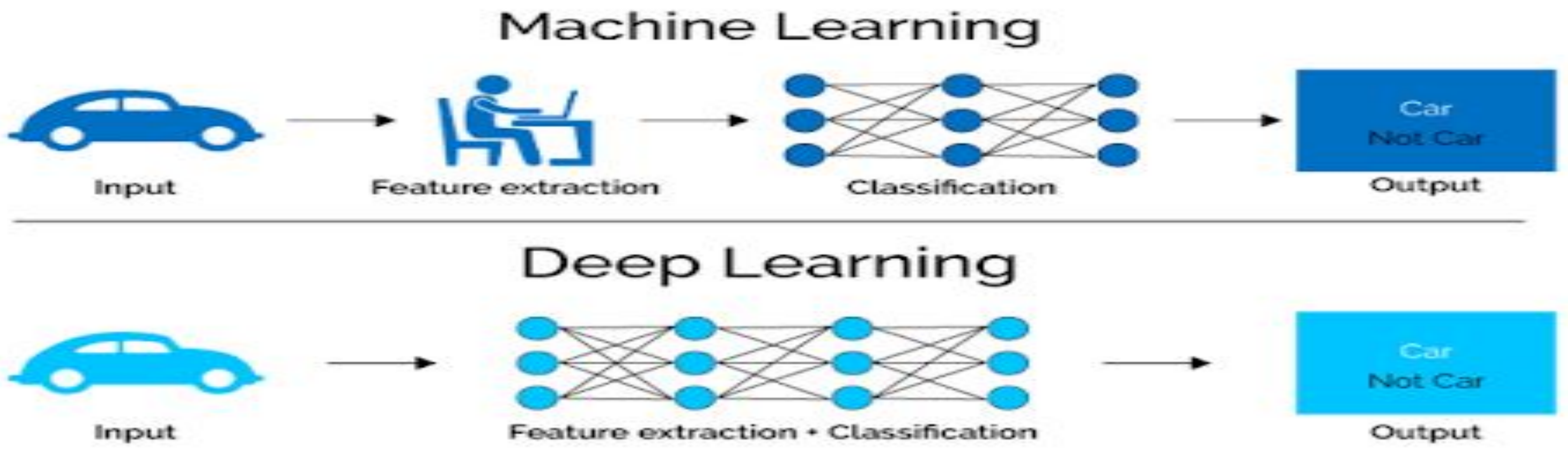
SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Amazon Machine Learning Platform

Amazon Lex	• Allows users to **incorporate voice input into applications**. • This service is an extension of amazon's echo product, a small networked device with a speaker and a microphone to which you can pose questions about the weather and make requests to schedule events, play music, and report on the latest news
Amazon Polly	• Amazon polly is the opposite of lex: it **turns text into speech**. • It can speak in 27 languages with a variety of voices. • Using the speech synthesis markup language, you can carefully control pronunciation and other aspects of intonation.
Amazon Rekognition	• Is at the **cutting edge of deep learning applications**. • It takes an image as **input and returns a textual description** of the items that it sees in that image

Deep Learning: A Shallow Introduction

- Deep learning can be considered as a subset of machine learning.
- It is a field that is based on learning **and improving on its own by examining computer algorithms**.
- While machine learning uses simpler concepts, deep learning works with **artificial neural networks**, which are designed to imitate how humans think and learn.

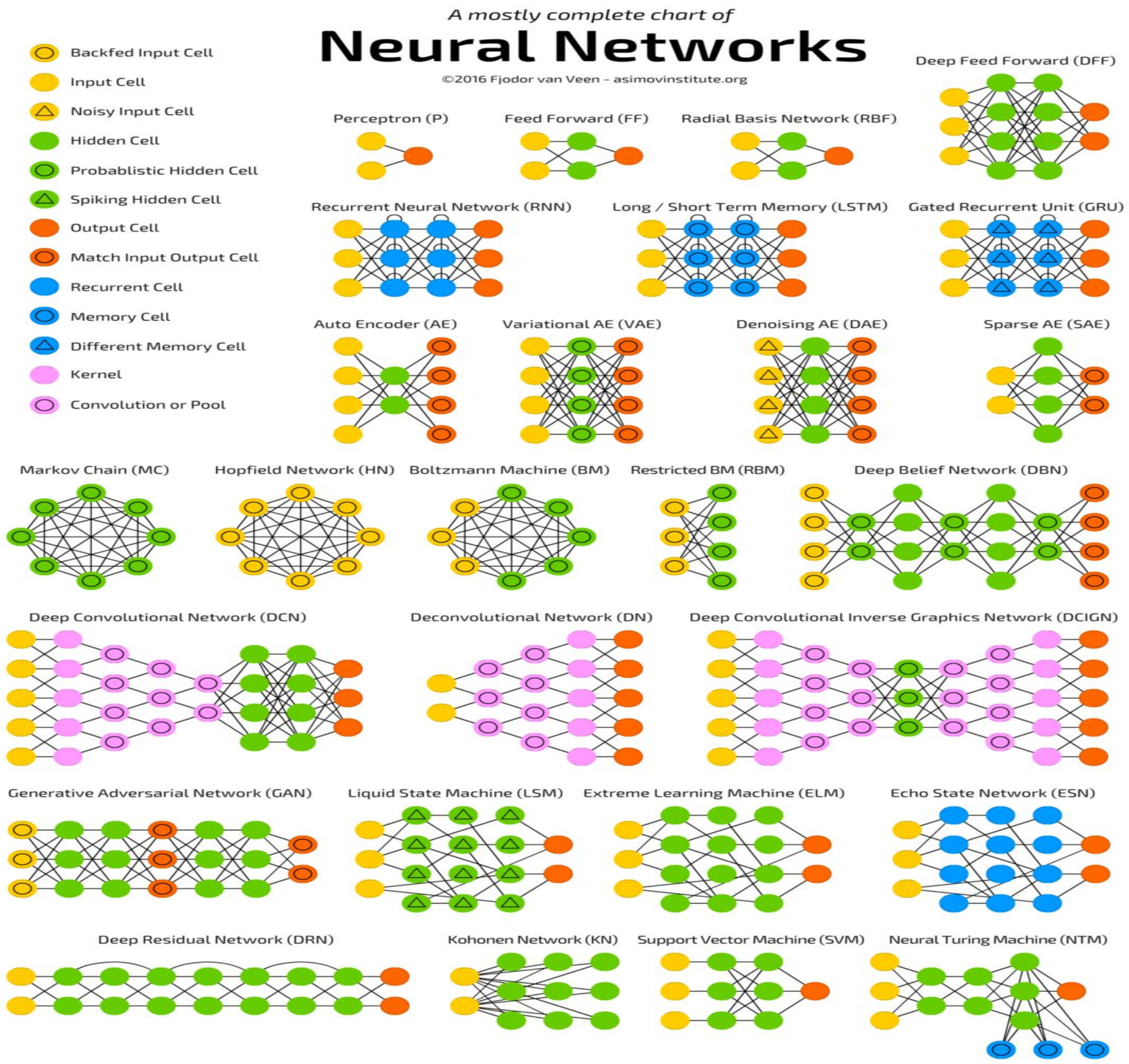


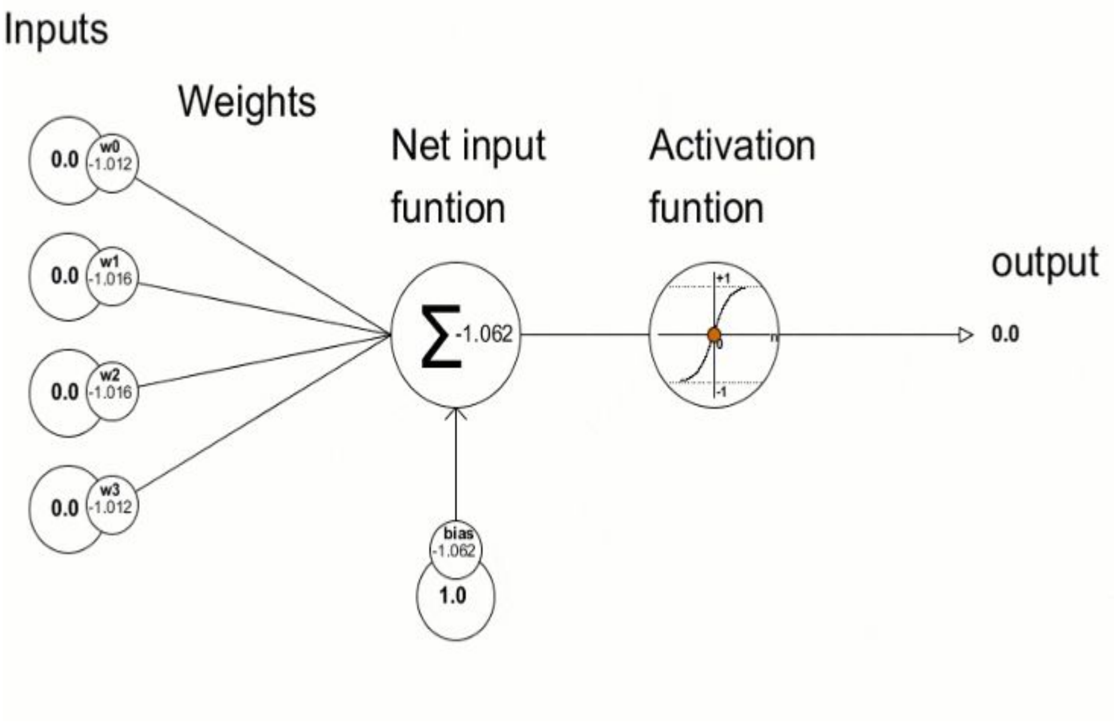
Basic concepts

- Neural network with **three inputs, one hidden layer, and two outputs.**
- **Input Layer**: As the name suggests, it accepts inputs in several different formats provided by the programmer.
- **Hidden Layer:** The hidden layer presents in-between input and output layers. **It performs all the calculations to find hidden features and patterns**.
- **Output Layer:** The input goes through a series of transformations using the hidden layer, which finally results in **output that is conveyed using this layer**
- **Bias** allows you to shift the **activation function by adding a constant (i.e. the given bias) to the input**
- **Activation Function** is a **mathematical formula that helps the neuron to switch ON/OFF**

Basic concepts

- **Feedforward neural network**: Consists of an input layer, one or a few hidden layers, and an output layer (a typical shallow neural network)
- **Convolutional neural network (CNN):** Deep neural network architecture widely applied **to image processing and characterized by convolutional layers** that shift windows across the input with nodes that share weights, abstracting the (typically image) input to feature maps
- **Recurrent neural network (RNN):** Neural network architecture with **FEEDBACK LOOPS** that model sequential dependencies in the input, as in time series, sensor, and text data; the most popular type of RNN is a long short-term memory network (LSTM)


A mostly complete chart of
Neural Networks
©2016 Fjodor van Veen - asimovinstitute.org
Backfed Input Cell
Input Cell
Noisy Input Cell
Hidden Cell
Probablistic Hidden Cell
Spiking Hidden Cell
Output Cell
Match Input Output Cell
Recurrent Cell
Memory Cell
Different Memory Cell
Kernel
Convolution or Pool
Perceptron (P)
Feed Forward (FF)
Radial Basis Network (RBF)
Deep Feed Forward (DFF)
Recurrent Neural Network (RNN)
Long / Short Term Memory (LSTM)
Gated Recurrent Unit (GRU)
Auto Encoder (AE)
Variational AE (VAE)
Denoising AE (DAE)
Sparse AE (SAE)
Markov Chain (MC)
Hopfield Network (HN)
Boltzmann Machine (BM)
Restricted BM (RBM)
Deep Belief Network (DBN)
Deep Convolutional Network (DCN)
Deconvolutional Network (DN)
Deep Convolutional Inverse Graphics Network (DCIGN)
Generative Adversarial Network (GAN)
Liquid State Machine (LSM)
Extreme Learning Machine (ELM)
Echo State Network (ESN)
Deep Residual Network (DRN)
Kohonen Network (KN)
Support Vector Machine (SVM)
Neural Turing Machine (NTM)



1. An artificial **neuron takes the inputs and their respective weights**.
2. It then applies dot products between **input values & its weights** and sums them up.
3. Finally, it applies activation function on above summation and fires the output

This can be written in a simple way as below –

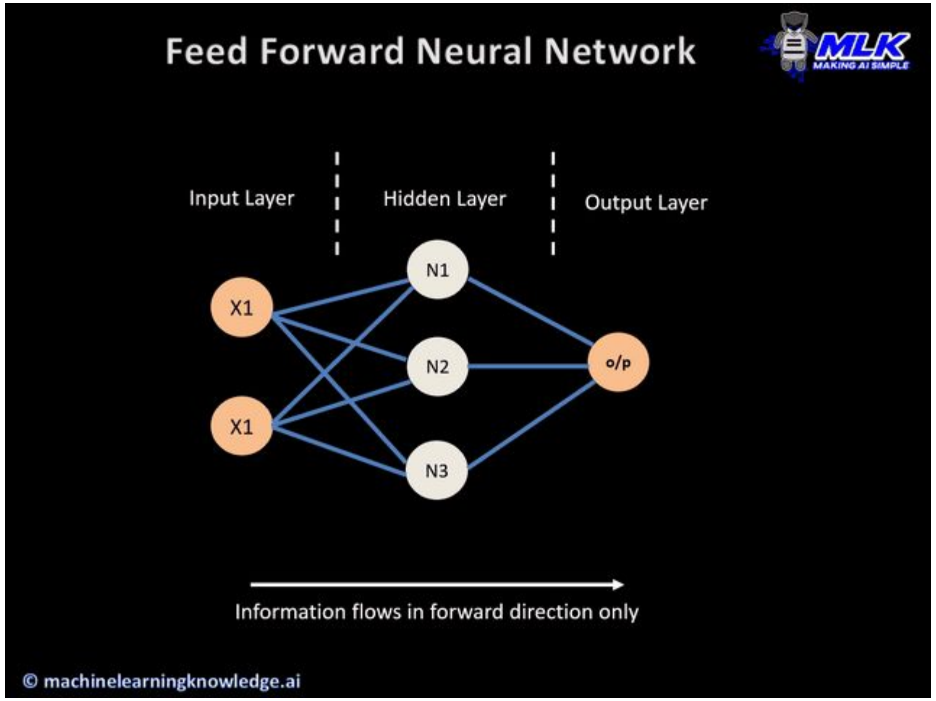
OUTPUT =

Activationfunction(Summation(Inputs*Weights **+** bias))

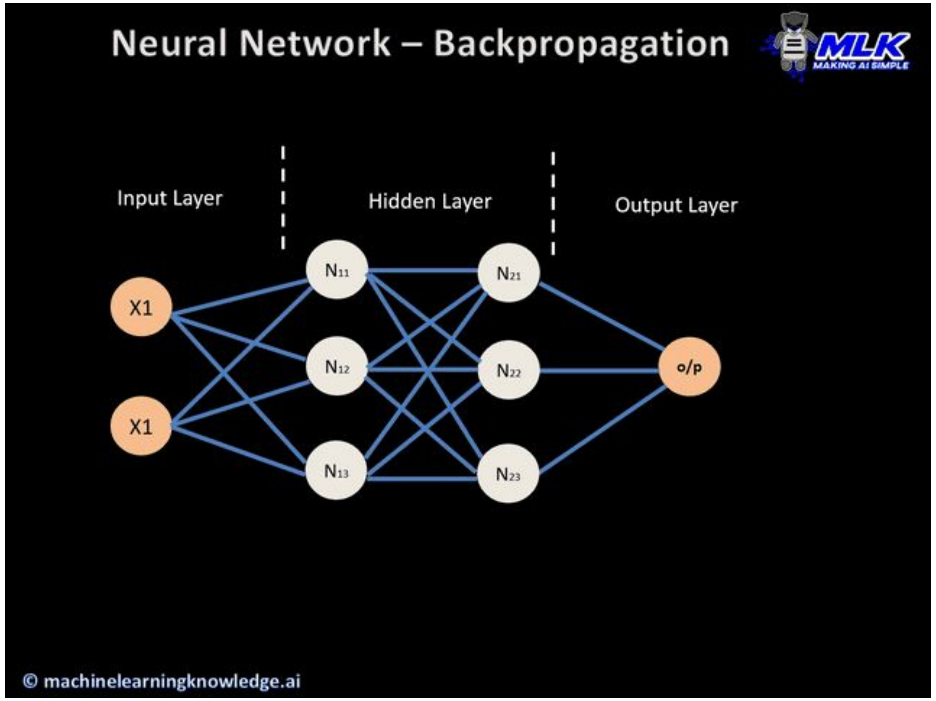
https://machinelearningknowledge.ai/animated-explanation-of-feed-forward-neural-network-architecture/

Activation Function adds Non linearity to Neural Network

Types of NN



Feed Forward Neural Network
MLK
MAKING AI SIMPLE
Input Layer
Hidden Layer
Output Layer
N1
X1
N2
o/p
X1
N3
Information flows in forward direction only
© machinelearningknowledge.ai



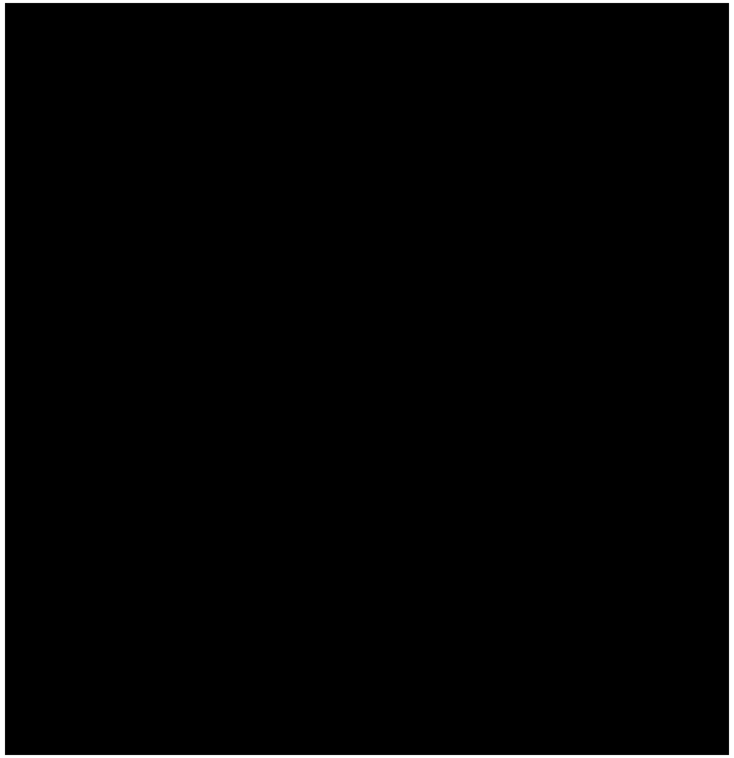
Neural Network – Backpropagation
MLK
MAKING AI SIMPLE
Input Layer
Hidden Layer
Output Layer
N11
N21
X1
N12
N22
o/p
X1
N13
N23
© machinelearningknowledge.ai

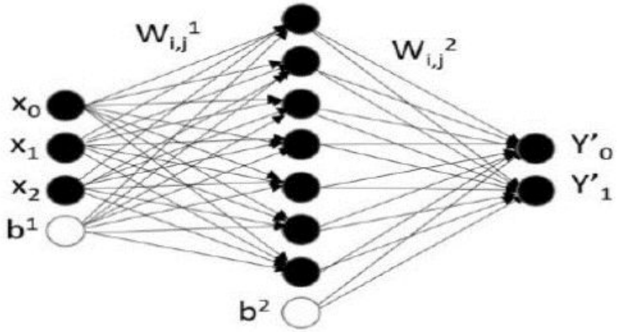
Basic Popular Activation Functions

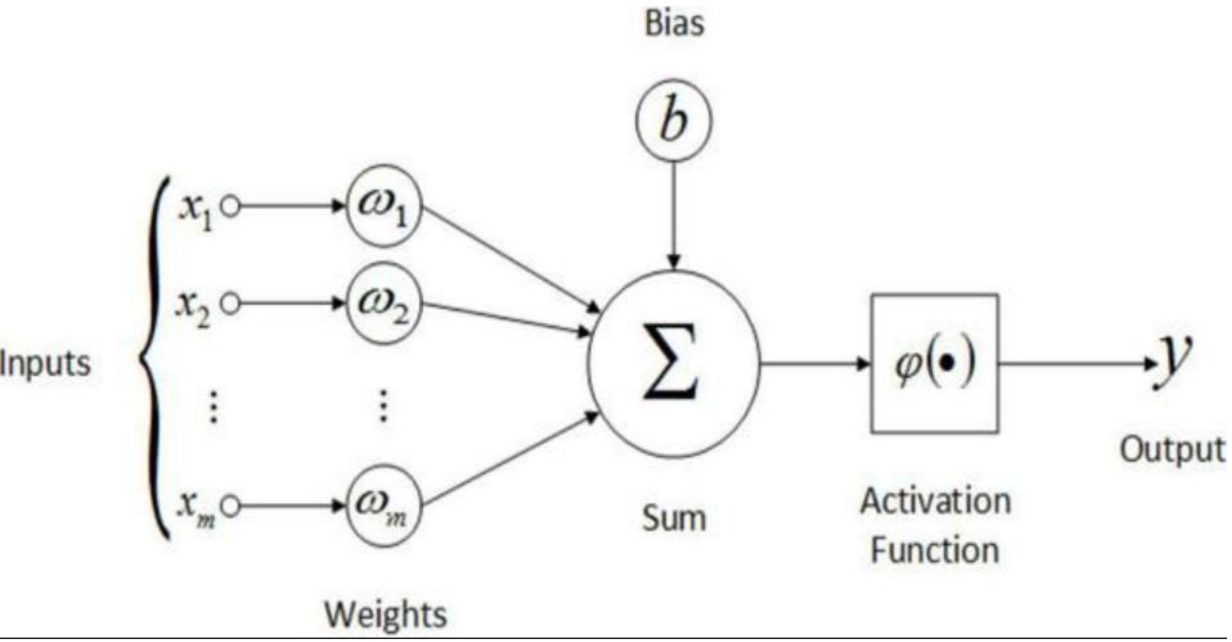
Sigmoid / Logistic	• Sigmoid function gives an **'S' shaped** curve. • In order to map predicted values to probabilities, we use the sigmoid function. • The function maps any real value into another value **between 0 and 1.** • The function is **differentiable so** we can find the **slope of the sigmoid curve at any two points.**
Tanh	The tanh (or hyperbolic tangent) Activation Function goes between **-1 and +1,** and is in fact a **shifted version of the sigmoid function**. For intermediate layers, the tanh function generally performs pretty well because, with values **between -1 and +1, performs better than sigmoid function**

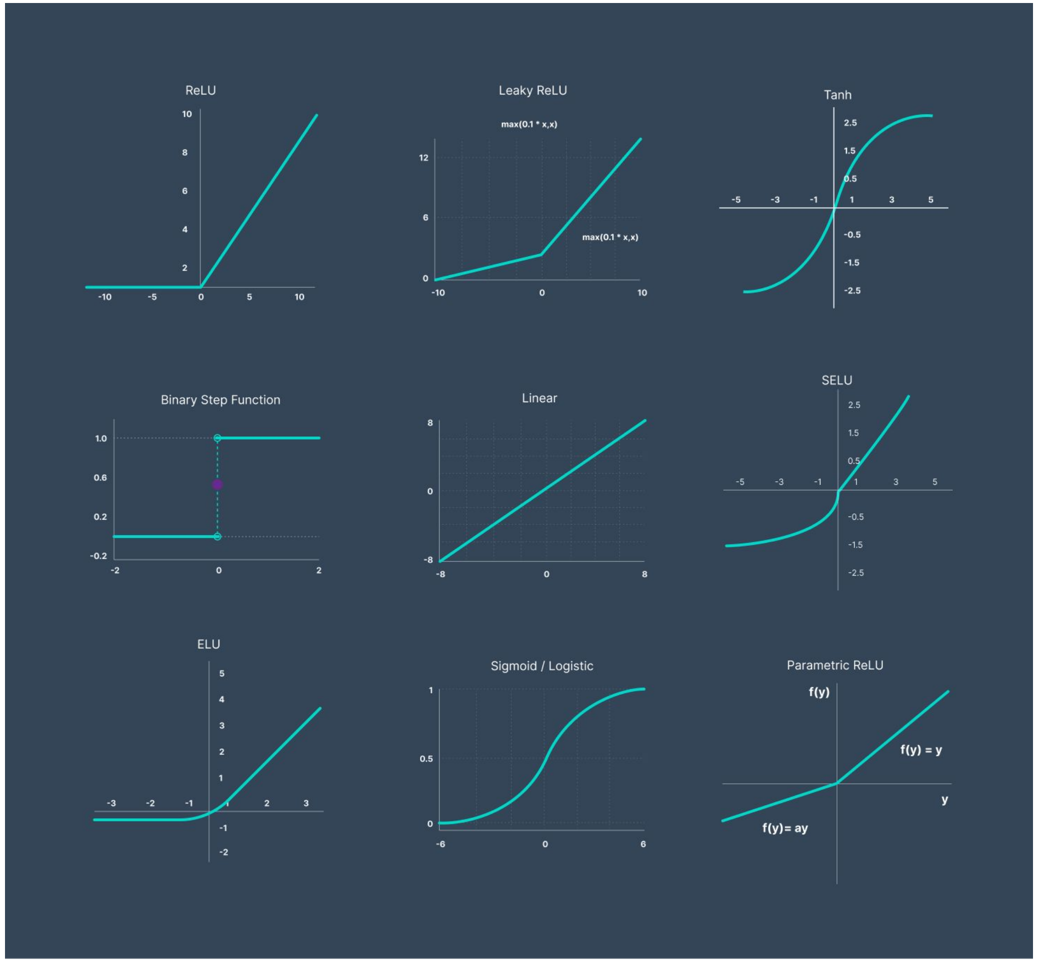
ReLU (Rectified Linear Unit)	• The **rectified linear activation function or ReLU** for short is a piecewise linear function that will output the input directly **if it is positive**, otherwise, it will output zero. • This means that for any input x, **the function outputs x if x is positive, and 0 if x is negative** • It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance.
Softmax	• Softmax function calculates the probabilities distribution of the event over a set of different events. • **Softmax function** is used to **compute the probability distribution over multiple classes or events, ensuring that the output values are between 0 and 1 and sum up to 1.** • Typically **Softmax is used only for the output layer**, for neural networks that need to classify inputs into multiple categories.

Activation function

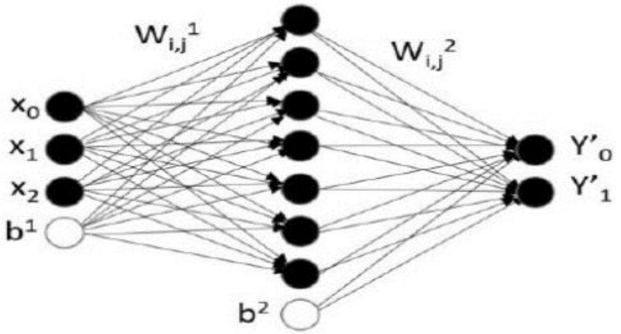



$W_{i,j}^1$
$W_{i,j}^2$
x_0
x_1
x_2
b^1
b^2
Y'_0
Y'_1


Bias
b
Inputs
x_1
x_2
x_m
ω_1
ω_2
ω_m
Weights
Σ
Sum
$\varphi(\bullet)$
Activation Function
y
Output


ReLU
Leaky ReLU
max(0.1 * x,x)
Tanh
Binary Step Function
Linear
SELU
ELU
Sigmoid / Logistic
Parametric ReLU
f(y)
f(y) = y
f(y)= ay
y

- **Weights are numeric values** that are multiplied by inputs
- The lines represent numerical **weights connecting the inputs** to the n interior neurons, and the terms b are offsets.



$$a_j = f\left(\sum_{i=0}^{2} x_i W_{i,j}^{1} + b_j\right) \qquad for \quad j = 1, n$$

$$y'_j = f'\left(\sum_{i=0}^{n} a_i W_{i,j}^{2} + b_j^{2}\right) \qquad for \quad j = 0, 1$$

- The purpose of the **activation function** is to introduce non-linearity into the output of a neuron.
- The functions **f and f' are called the activation functions** for the neurons.
- Two commonly used activation functions are the logistic function σ(t)

- **Rectified linear function**
- **Relu(x) = max(0, x).**
- Another common case is the **hyperbolic tangent function**.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- An advantage of σ(x) and tanh(x) is that they map values in the **range (−∞, ∞) to the range (0, 1),** which corresponds to the idea that a neuron is either on or off (not-fired or fired).

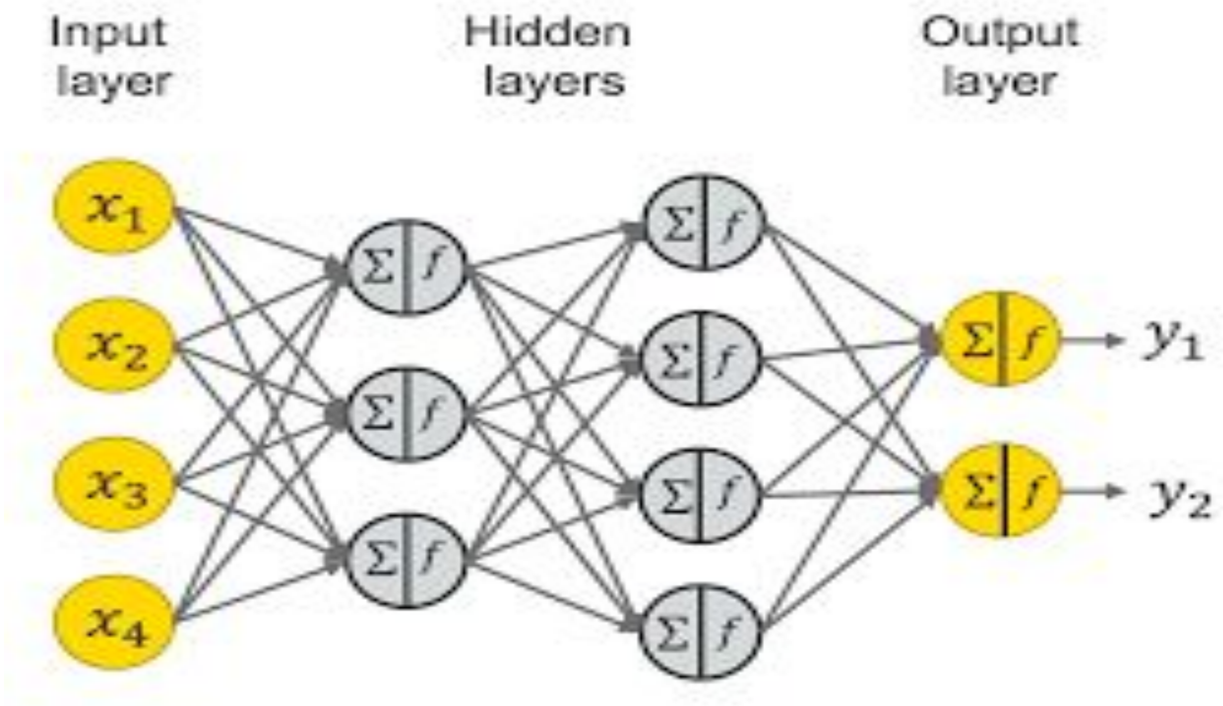
$$\mathrm{softmax}(x)_j = \frac{1}{1 + \sum_{k \neq j} e^{x_k - x_j}}$$

- Function is picking the **right values for the weights** by using **minimize the cost function**

$$C(x^i, y^i) = \sum \|y^i - y'(x^i)\|$$

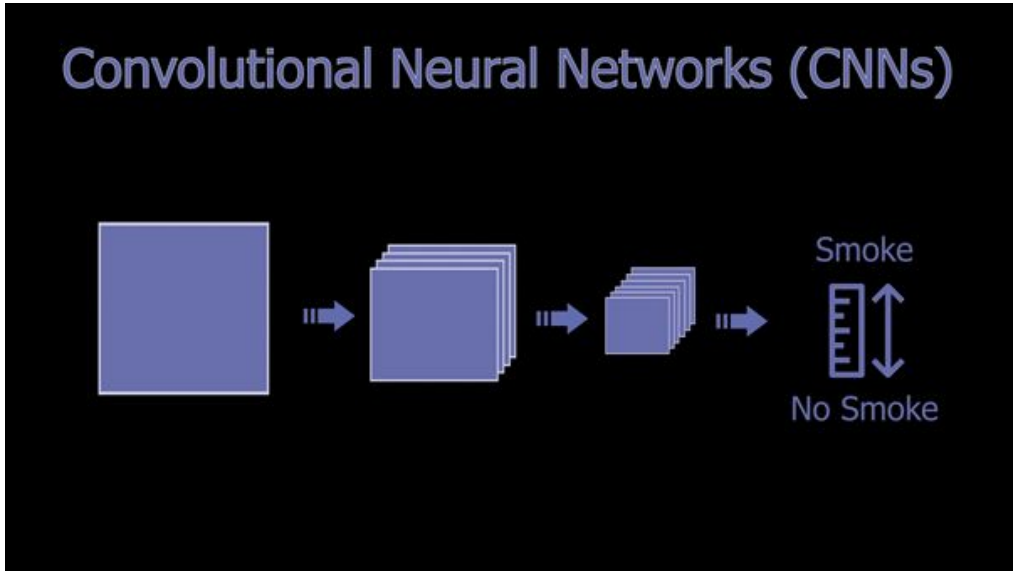
```
data = mx.symbol.Variable('x')
layr1 = mx.symbol.FullyConnected(data=data,name='W1', num_hidden=7)
act1 = mx.symbol.Activation(data=layr1,name='relu1', act_type="relu")
layr2 = mx.symbol.FullyConnected(data=act1,name='W2', num_hidden=4)
act2 = mx.symbol.Activation(data=layr2,name='relu2', act_type="relu")
layr3 = mx.symbol.FullyConnected(data=act2,name='W3', num_hidden=2)
Y = mx.symbol.SoftmaxOutput(data = layr3,name='softmax')
```

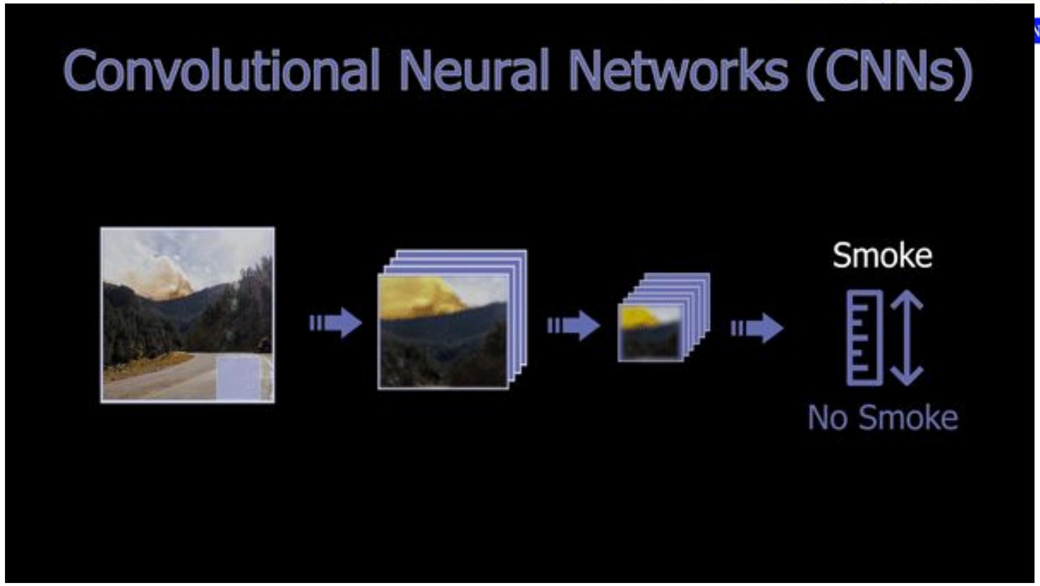
Deep Networks

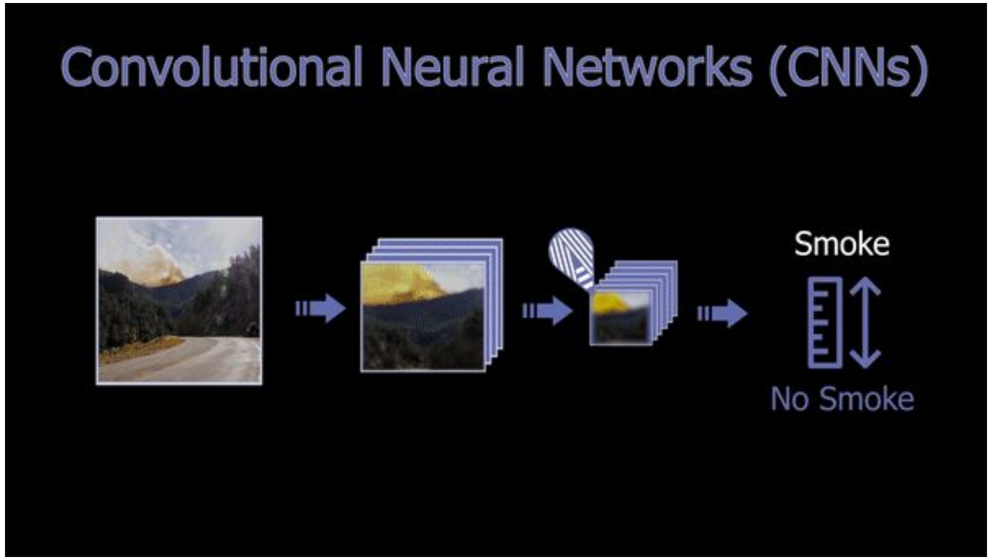


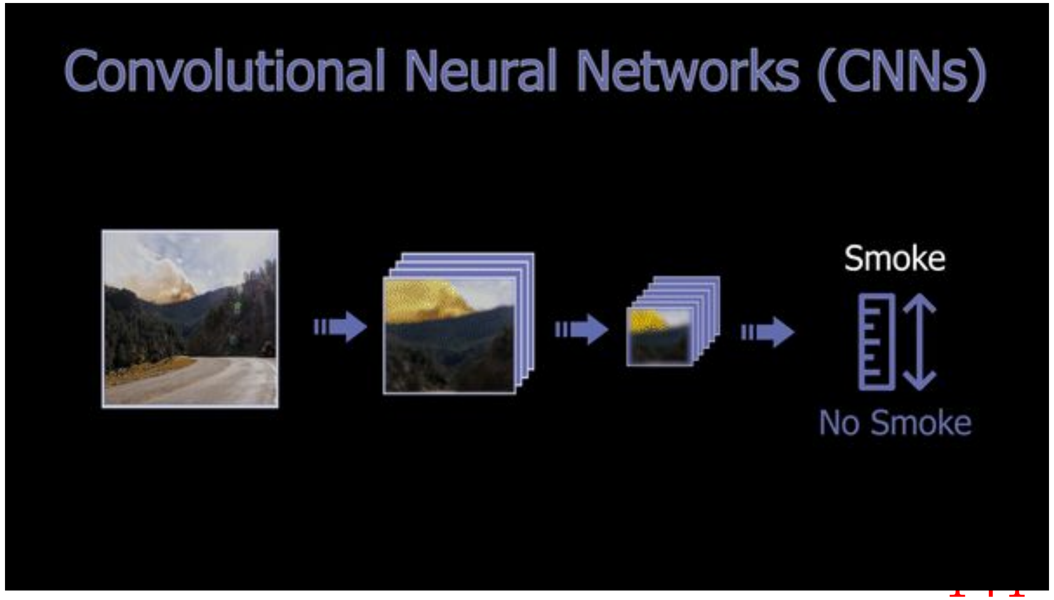
Convolutional Neural Networks

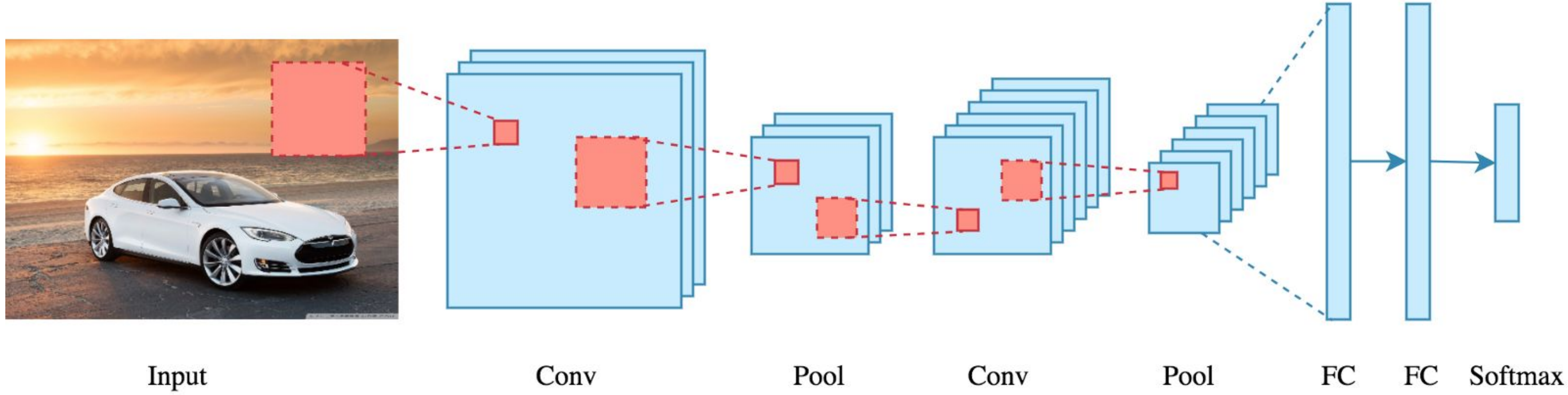
- Data with a regular spatial geometry such as images or one dimensional streams are often analyzed with a special class of network called a **convolutional neural network or CNN**.
- CNNs eliminate the need for **MANUAL FEATURE EXTRACTION**, so you do not need to identify features used to classify images.
- The CNN works by extracting features directly **from images**.
- This **automated feature extraction makes deep learning** models highly accurate for computer vision tasks such as object classification
- A special technique called **Convolution**. Now in mathematics convolution is a mathematical operation **on two functions that produces a third function** that expresses how the shape of one is modified by the other.


Convolutional Neural Networks (CNNs)
Smoke
No Smoke

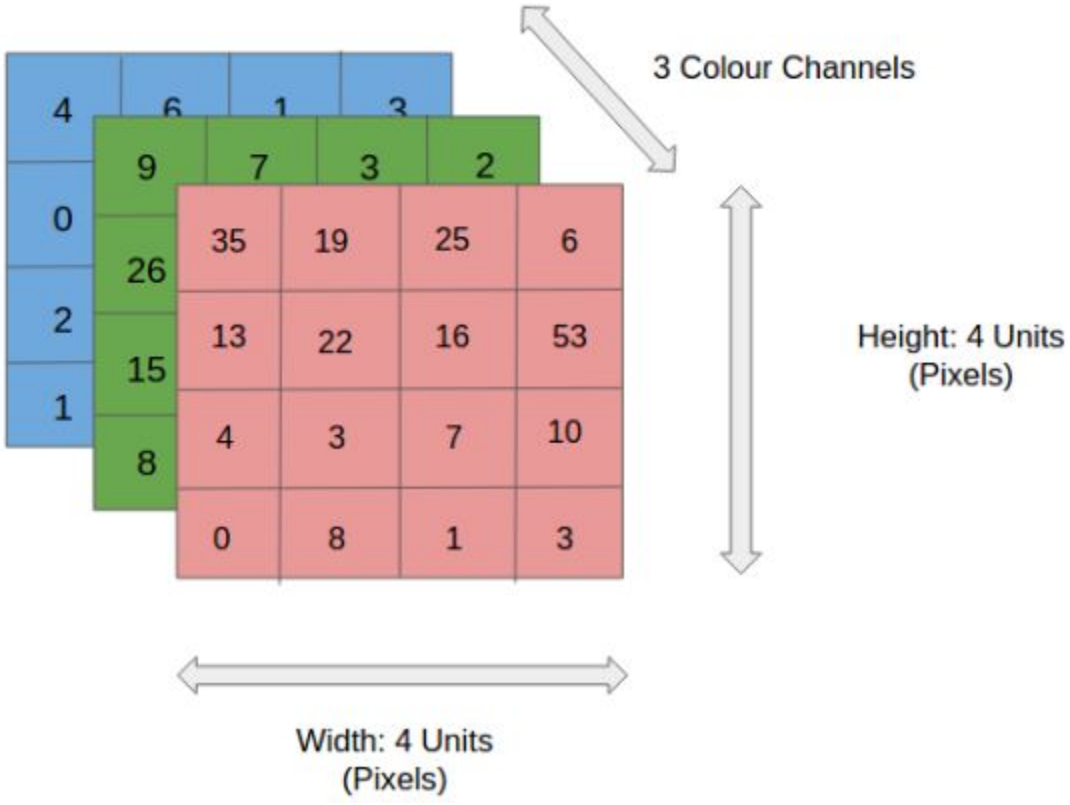

Convolutional Neural Networks (CNNs)
Smoke
No Smoke

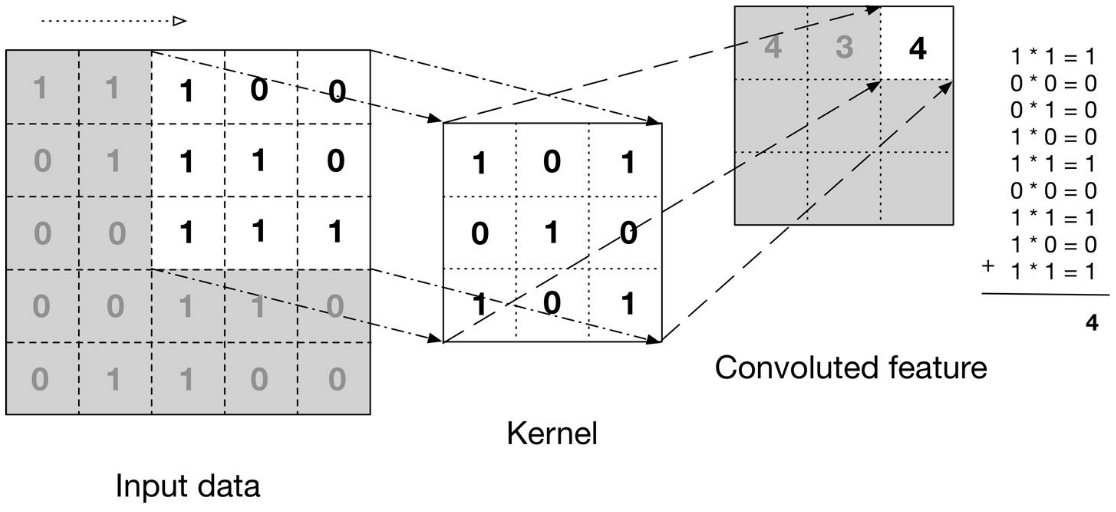

Convolutional Neural Networks (CNNs)
Smoke
No Smoke


Convolutional Neural Networks (CNNs)
Smoke
No Smoke


Input
Conv
Pool
Conv
Pool
FC
FC
Softmax

- An RGB image is nothing but a **matrix of pixel values** having three planes whereas a grayscale image is the same but it has a single plane.





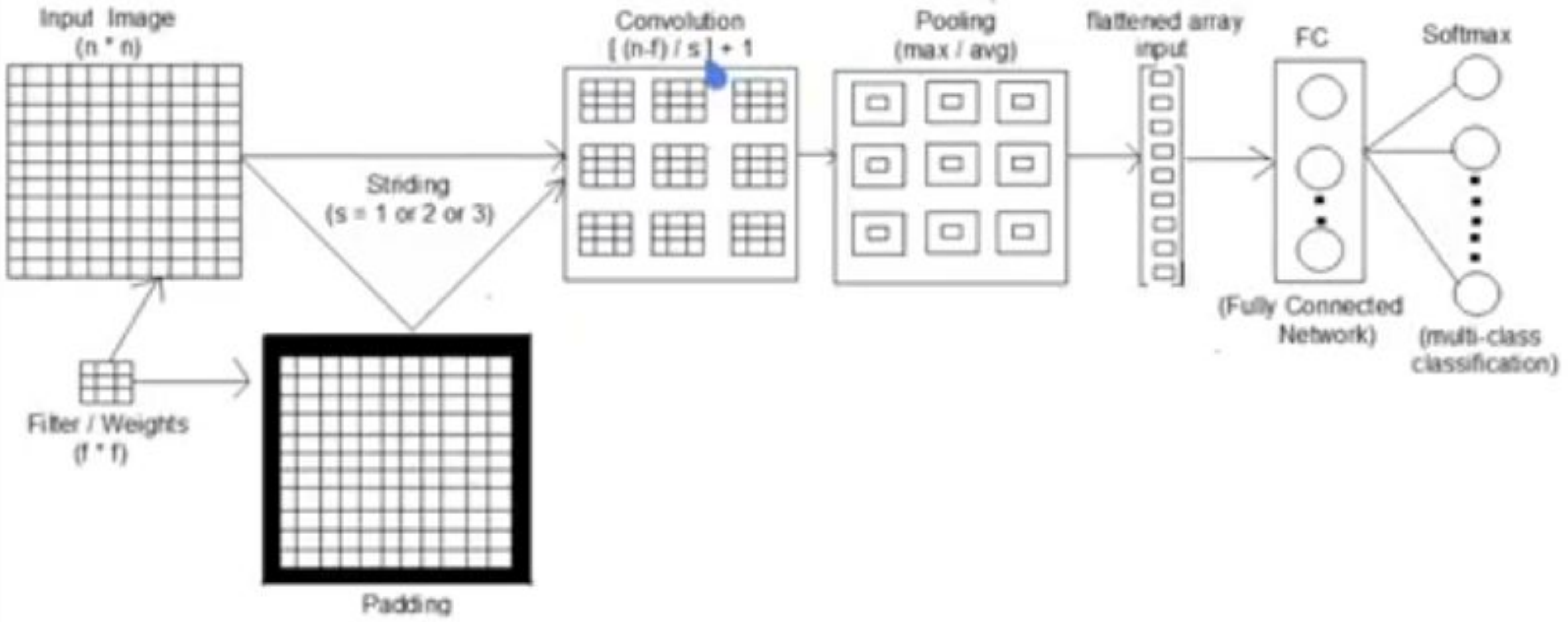
For next slide image matrix apply the below formula

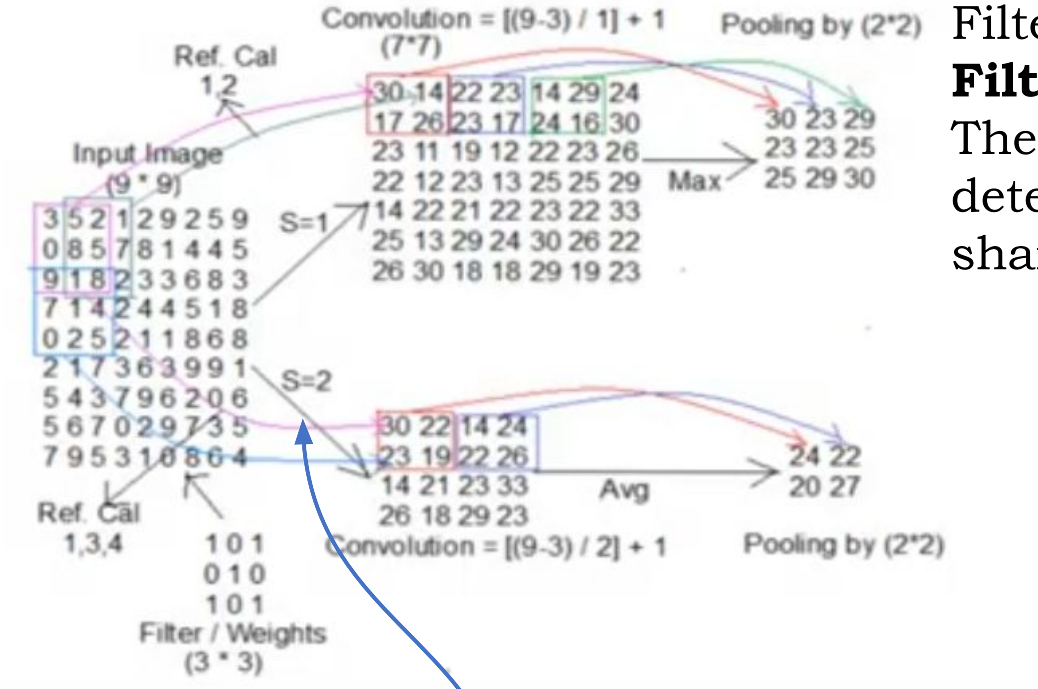
[(n-f)/s]+1 if we took s=1 [(9- 3)/ 1] +1 **if we took s=2 [(9- 3)/2] +1**

N-input img [6]+1 =7 [(6)/2] +1= 4

F-filter img

S-Striding img





Filter : we can define or we can use **Sobel Filter, Prewitt Filter**
The above mentioned are some of edge detection filter likewise we have sharpening, smoothing types of filter

Input * Weights and summation

$$\text{Cal-1}\ 30 = \begin{matrix}3&5&2\\0&8&5\\9&1&8\end{matrix} * \begin{matrix}1&0&1\\0&1&0\\1&0&1\end{matrix} = \begin{matrix}3*1&5*0&2*1\\0*0&8*1&5*0\\9*1&1*0&8*1\end{matrix} = \begin{matrix}3&0&2\\0&8&0\\9&0&8\end{matrix} = 3+2+8+9+8 = 30$$

$$\text{Cal-2}\ 14 = \begin{matrix}5&2&1\\8&5&7\\1&8&2\end{matrix} * \begin{matrix}1&0&1\\0&1&0\\1&0&1\end{matrix} = \begin{matrix}5*1&2*0&1*1\\8*0&5*1&7*0\\1*1&8*0&2*1\end{matrix} = \begin{matrix}5&0&1\\0&5&0\\1&0&2\end{matrix} = 5+1+5+1+2 = 14$$

$$\text{Cal-3}\ 22 = \begin{matrix}2&1&2\\5&7&8\\8&2&3\end{matrix} * \begin{matrix}1&0&1\\0&1&0\\1&0&1\end{matrix} = \begin{matrix}2*1&1*0&2*1\\5*0&7*1&8*0\\8*1&2*0&3*1\end{matrix} = \begin{matrix}2&0&2\\0&7&0\\8&0&3\end{matrix} = 2+2+7+8+3 = 22$$

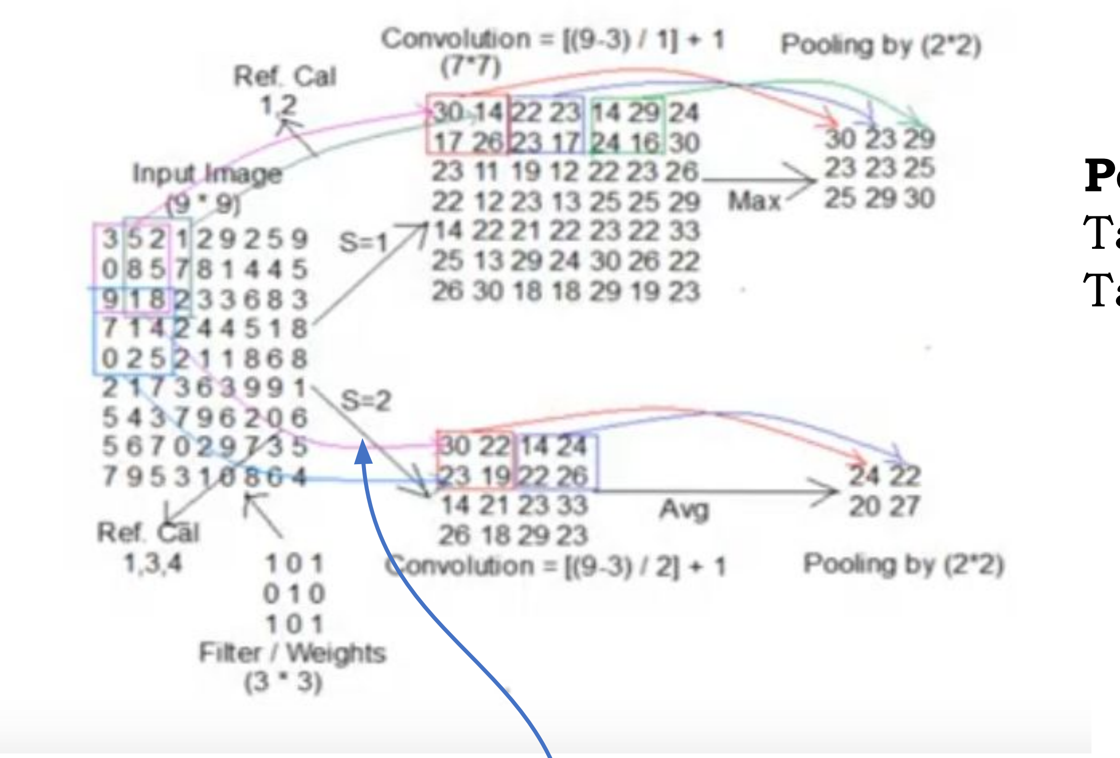
$$\text{Cal-4}\ 23 = \begin{matrix}9&1&8\\7&1&4\\0&2&5\end{matrix} * \begin{matrix}1&0&1\\0&1&0\\1&0&1\end{matrix} = \begin{matrix}9*1&1*0&8*1\\7*0&1*1&4*0\\0*1&2*0&5*1\end{matrix} = \begin{matrix}9&0&8\\0&1&0\\0&0&5\end{matrix} = 9+8+1+0+5 = 23$$

Stride is a *parameter* of the neural network's filter that modifies the **amount of movement over the image or video. (whether we can have 1 or 2 col in input matrix)**

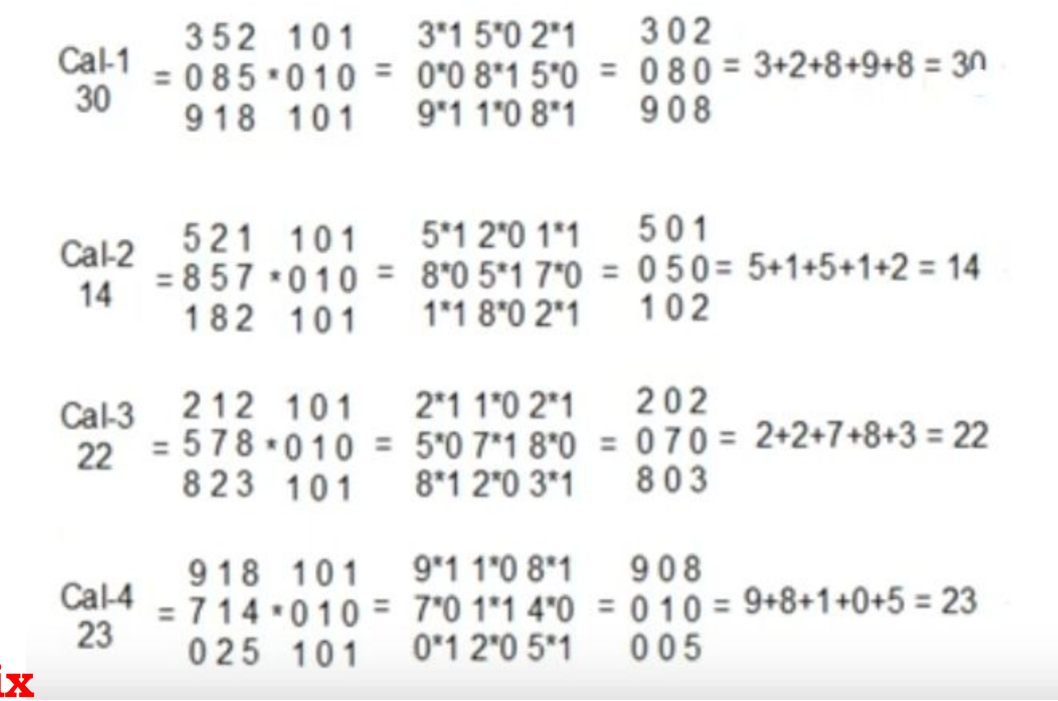
Padding is a process of adding zeros to the input matrix symmetrically

Pooling also we are applying the filter here in this ex (2*2)

Pooling also we are applying the filter here in this ex (2*2)
Taking MAX value (**30**,14,17,26)
Taking avg (30,22,23,19)= (30+22+23+19=23.5=**24**)



Input * Weights and summation



Stride is a *parameter* of the neural network's filter that modifies the **amount of movement over the image or video. (whether we can have 1 or 2 col in input matrix)**

Padding is a process of **adding zeros to the input matrix** like **border [0 0 0 0]** followed with input matrix mostly used to check the features in edges and 0 can be added in all sides of matrix

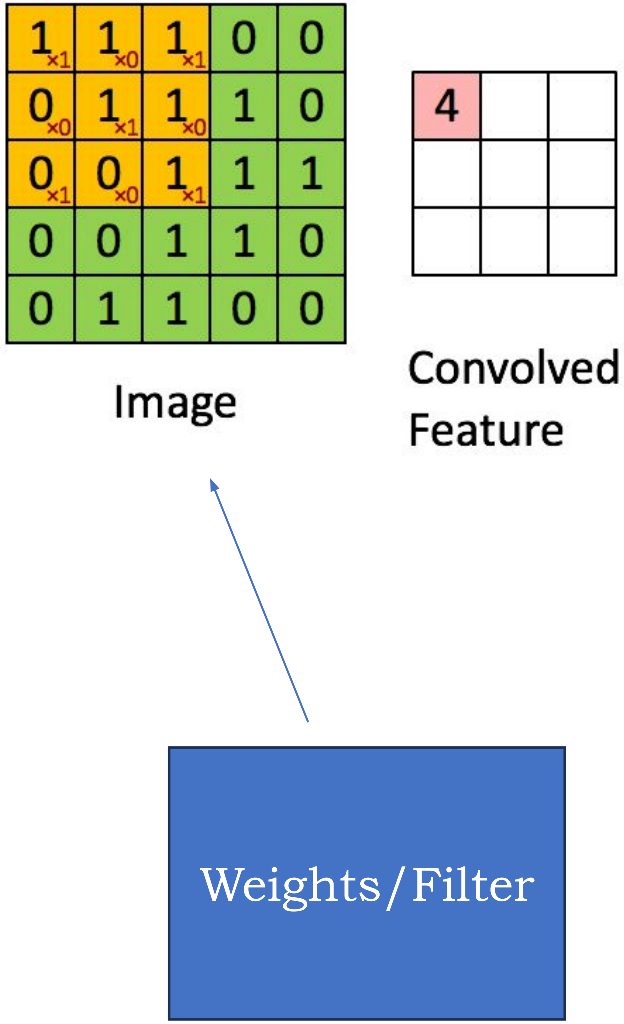
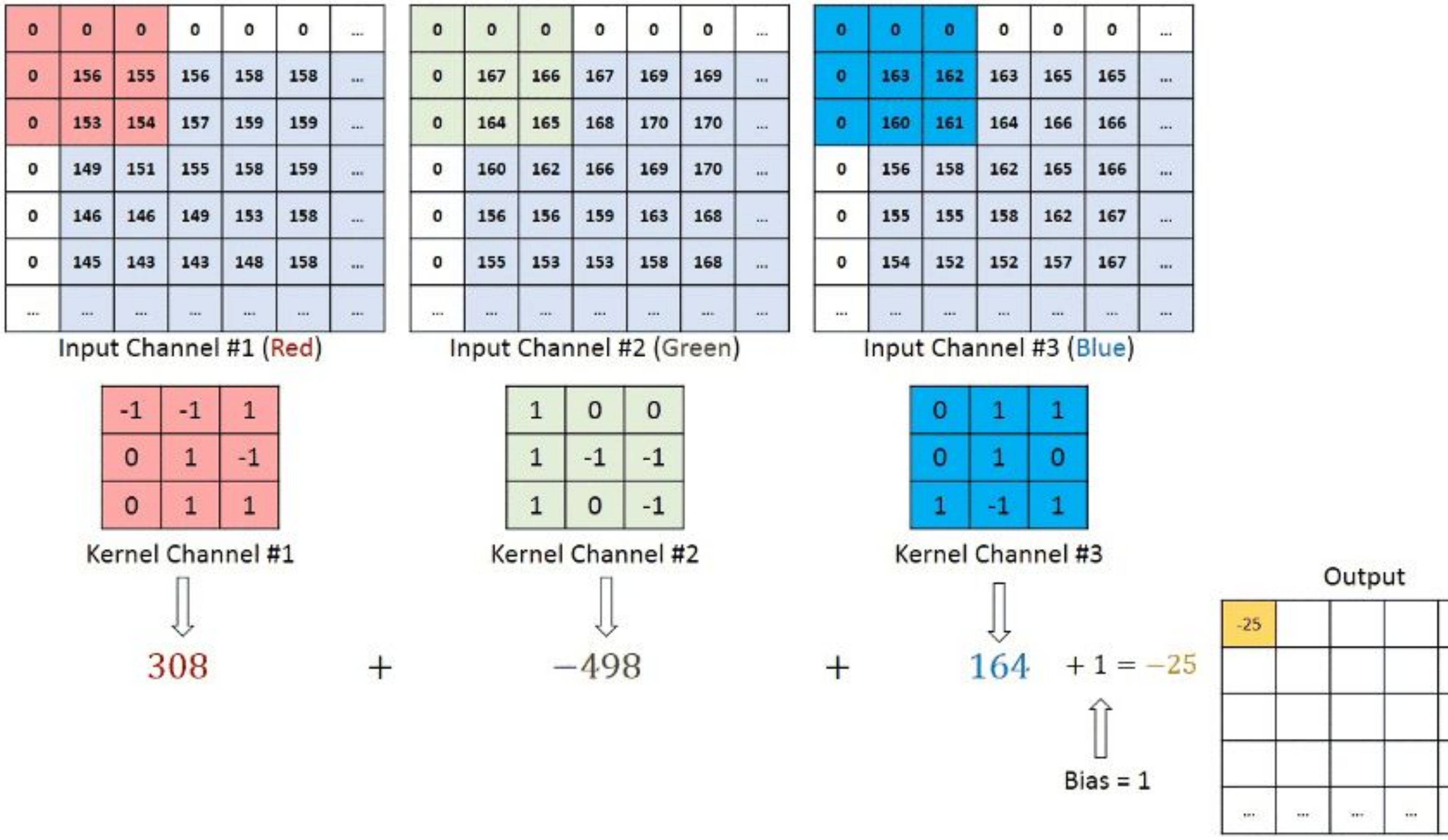


Image
Convolved Feature
Weights / Filter

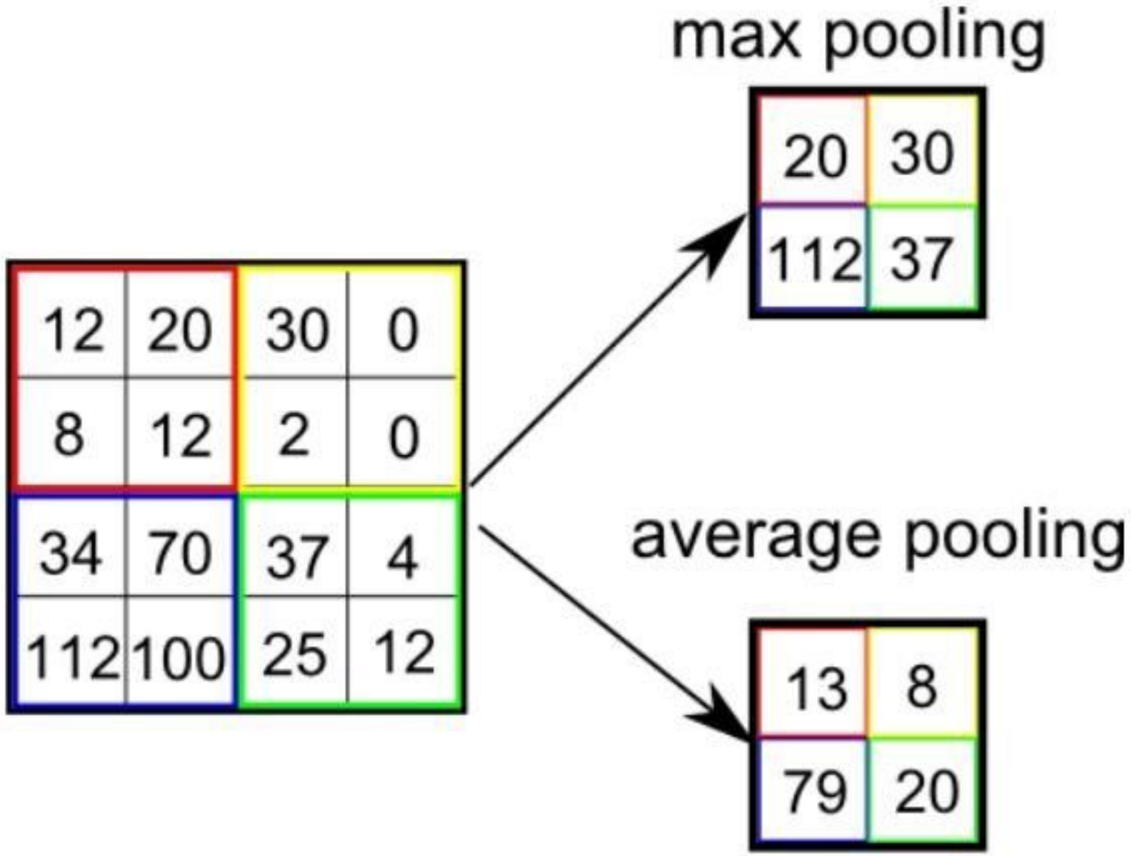

Input Channel #1 (Red)
Input Channel #2 (Green)
Input Channel #3 (Blue)
Kernel Channel #1
Kernel Channel #2
Kernel Channel #3
308
+
−498
+
164
+ 1 = −25
Bias = 1
Output
-25

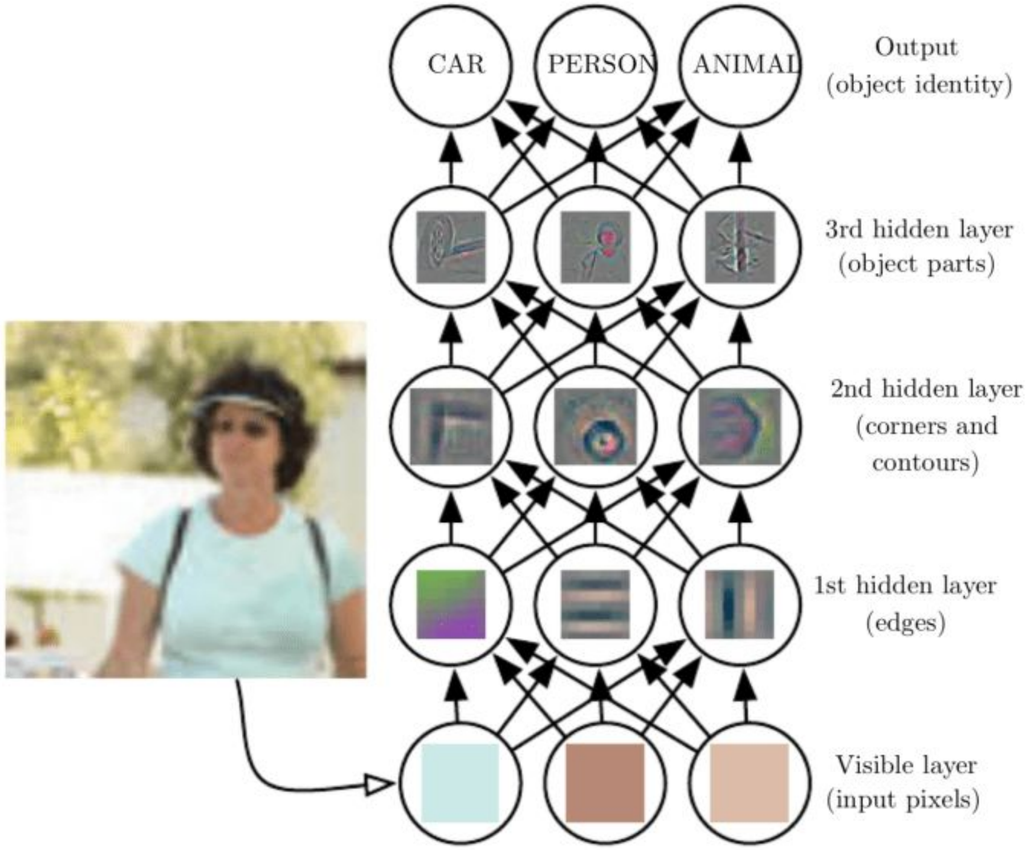
- CNNs learn to detect different features of an image using **tens or hundreds of hidden layers**.
- Every hidden layer **increases the complexity of the learned image features**.
- For example, the first hidden layer could learn how to detect edges
- the last learns how to detect more complex shapes specifically catered to the shape of the object we are trying to recognize.
- Bottom line is that the role of the ConvNet is to reduce the **images into a form that is easier to process, without losing features that are critical for getting a good prediction**.

Generic Steps in CNN

- The kernel is nothing but a filter **that is used to extract the features from the images.**
- Convolutional neural networks are composed of **multiple layers of artificial neurons.**
- Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the **weighted sum of multiple inputs and outputs an activation value**.
- When you input an image in a convnet, each layer generates several activation functions that are passed on to the next layer

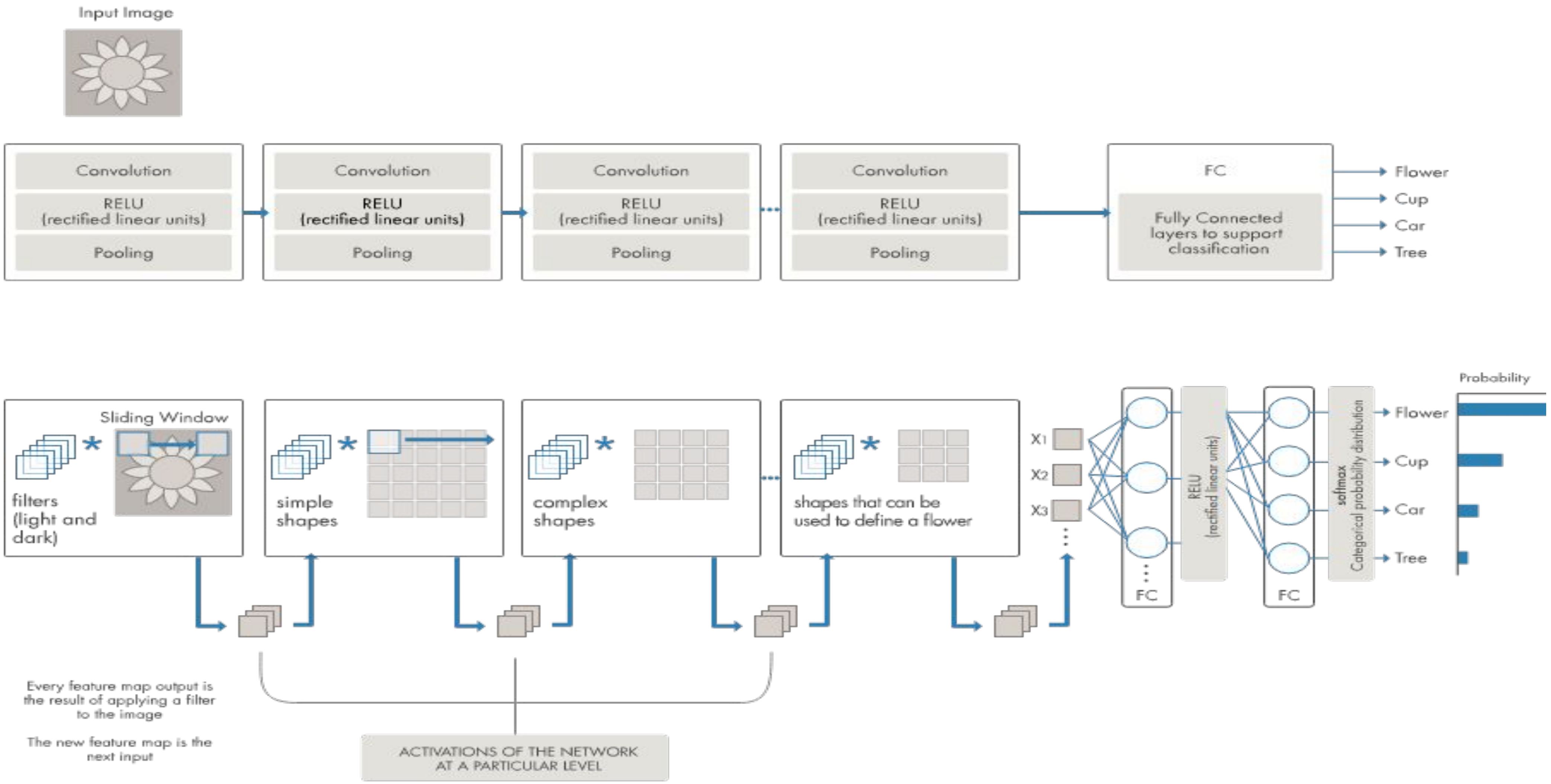
- The first layer usually **extracts basic features** such as *horizontal or diagonal edges*.
- This output is passed on to the next layer which **detects more complex features such as corners or combinational edges**.
- As we move deeper into the network it can identify even more complex features such as **objects, faces**, etc.
- Based on the activation map of the final convolution layer, the classification layer outputs a set of confidence scores (values between 0 and 1) that specify how likely the **image is to belong to a "class."**
- For instance, if you have a ConvNet that detects cats, dogs, and horses, the output of the final layer is the possibility that the input image **contains any of those animals**.


max pooling
average pooling
12 20 30 0
8 12 2 0
34 70 37 4
112 100 25 12
20 30
112 37
13 8
79 20


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)

- The pooling layer is responsible for **reducing the spatial size of the convolved feature.**
- This is to decrease the computational power required to process the data by reducing the dimensions.
- There are two types of pooling **AVERAGE POOLING AND MAX POOLING**
- Max Pooling is we find the **maximum value of a pixel** from a portion of the image covered by the kernel.
- Max Pooling also performs as a **Noise Suppressant**.
- It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction

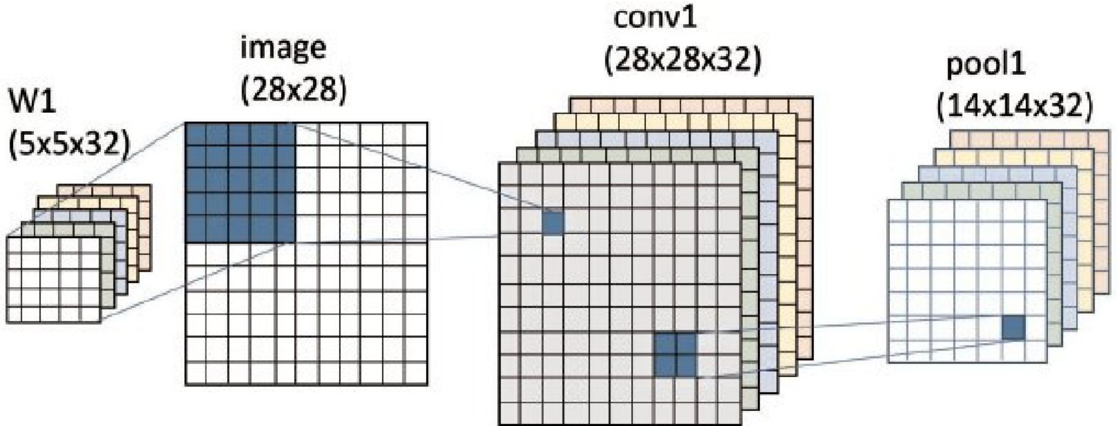
- Average Pooling returns the **average of all the values from the portion of the image covered by the Kernel**.
- Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism.
- Hence, we can say that Max Pooling performs a lot better than Average Pooling.


Input Image
Convolution
RELU
(rectified linear units)
Pooling
FC
Fully Connected layers to support classification
Flower
Cup
Car
Tree
Sliding Window
filters (light and dark)
simple shapes
complex shapes
shapes that can be used to define a flower
X1
X2
X3
FC
RELU (rectified linear units)
softmax
Categorical probability distribution
Probability
Every feature map output is the result of applying a filter to the image
The new feature map is the next input
ACTIVATIONS OF THE NETWORK AT A PARTICULAR LEVEL

- Suppose you have thousands of 28×28 black and white images of handwritten digits and you want to build a system that can identify each.
- Images are strings of bits, but they also have a lot of local two dimensional structure such as edges and holes.
- In order to find these patterns, we examine each of the many **5×5 windows** in each image individually.
- Train the system to build a 5×5 template **array W1** and a scalar **offset b** that together can be used to reduce each 5×5 window to a point in a new array conv by the following formula

$$\mathtt{conv}_{p,q} = \sum_{i,k=-2}^{2} W_{i,k}\,\mathtt{image}_{p-i,q-k} + b$$

- Next modify the conv array by applying the **relu function** to each x in the conv array so that it has **no negative values.**
- The final step, max pooling, simply computes the maximum value in each 2×2 block and assigns it to a smaller 14×14 array.
- The most interesting part of the convolutional network is that we do not use one 5×5 W1 template but 32 of them in parallel, producing 32 14×14 results, pool1

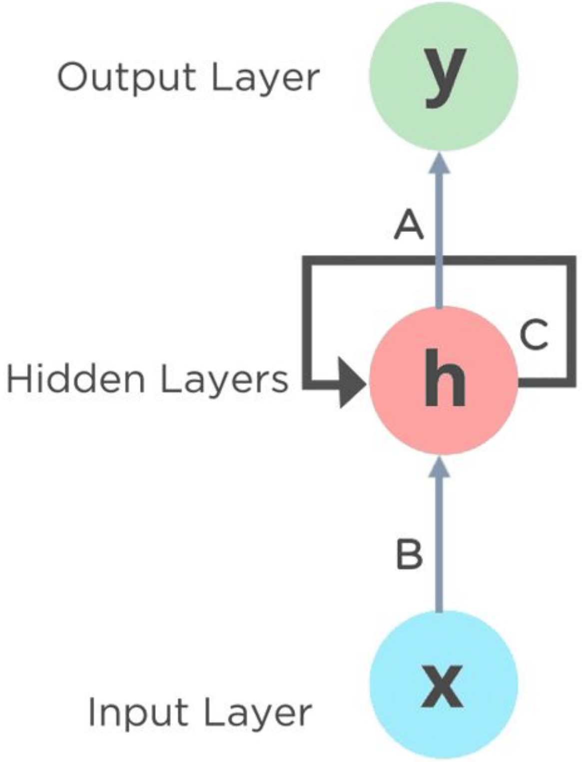


- When the network is fully trained, each of the 32 **5×5 templates in W1** is somehow different, and each selects for a different set of features in the original image.
- One can think of the resulting stack of 32 **14×14 arrays (called pool1) as a type of transform of the original image**, which works much like a **fourier transform** to separate a signal in space and time and transform it into frequency space.
- Second convolutional layer to pool1, but this time we apply 64 sets of 5×5 filters to each of the 32 pool1 layers and sum the results to obtain 64 new 14×14 arrays

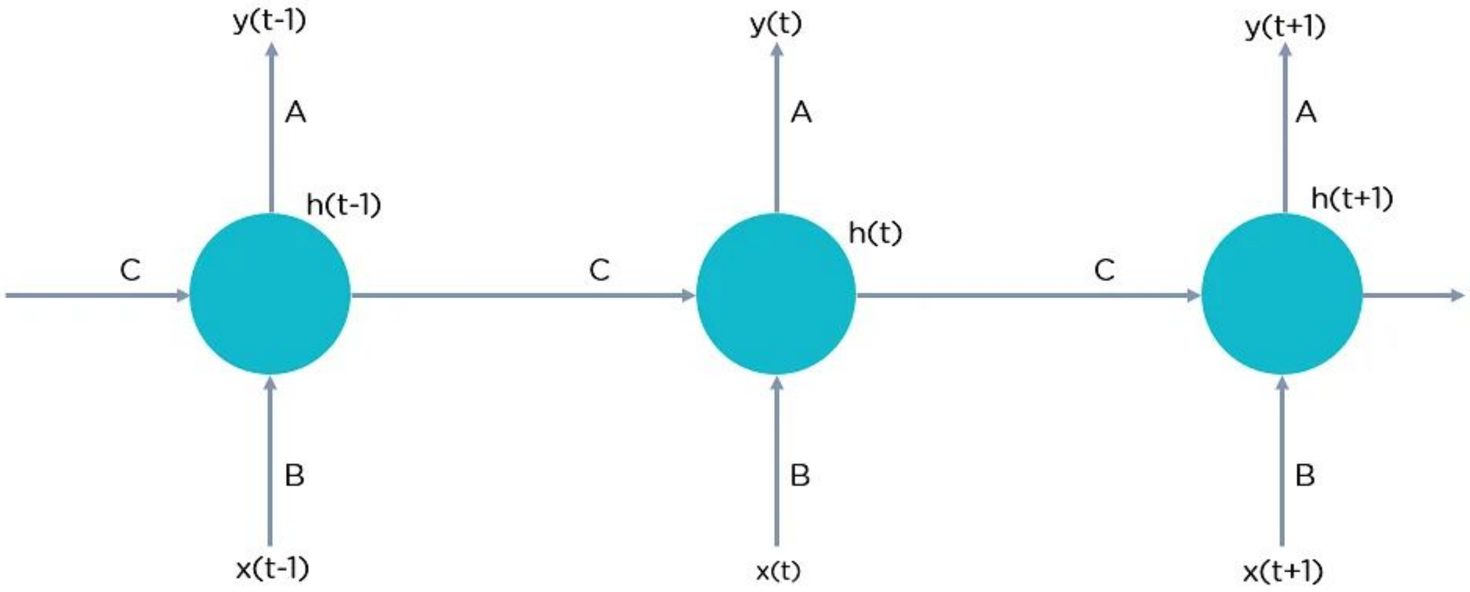
```
input_layer = tf.reshape(features, [-1, 28, 28, 1])

conv1 = tf.layers.conv2d(
      inputs=input_layer,
      filters=32,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)

pool1 = tf.layers.max_pooling2d(inputs=conv1, \
                              pool_size=[2, 2], strides=2)

conv2 = tf.layers.conv2d(
      inputs=pool1,
      filters=64,
      kernel_size=[5, 5],
      padding="same",
      activation=tf.nn.relu)

pool2 = tf.layers.max_pooling2d(inputs=conv2, \
                             pool_size=[2, 2], strides=2)
pool2_flat = tf.reshape(pool2, [-1, 7 * 7 * 64])

dense = tf.layers.dense(inputs=pool2_flat, \
                        units=1024, activation=tf.nn.relu)

logits = tf.layers.dense(inputs=dense, units=10)
```

Recurrent Neural Networks

- Recurrent neural networks (RNNs) are widely used in **LANGUAGE MODELING PROBLEMS**, such as *predicting the next word to be typed when texting or in automatic translation systems.*
- RNNs can learn from sequences that have repeating patterns.
- RNN works on the principle of saving the output of a particular layer and **feeding this back to the input in order to predict the output of the layer**.

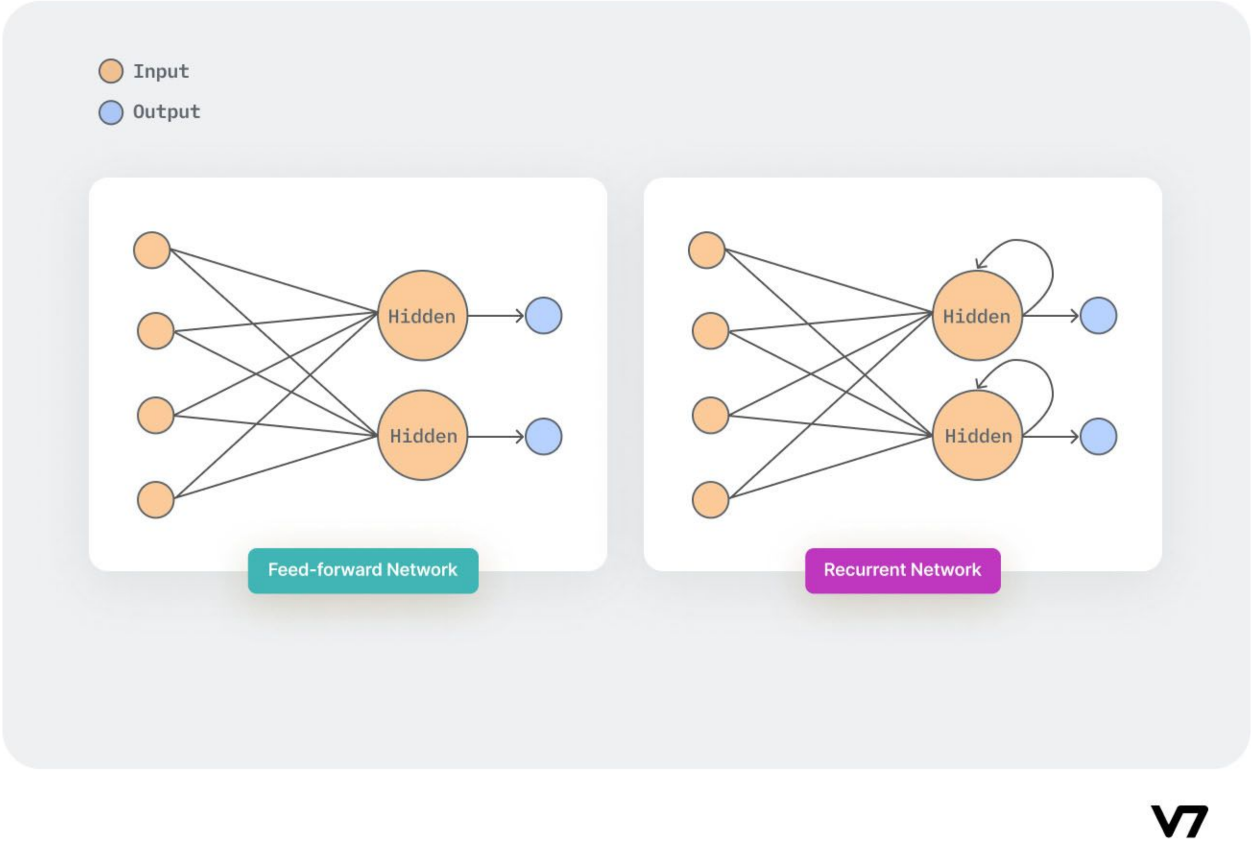


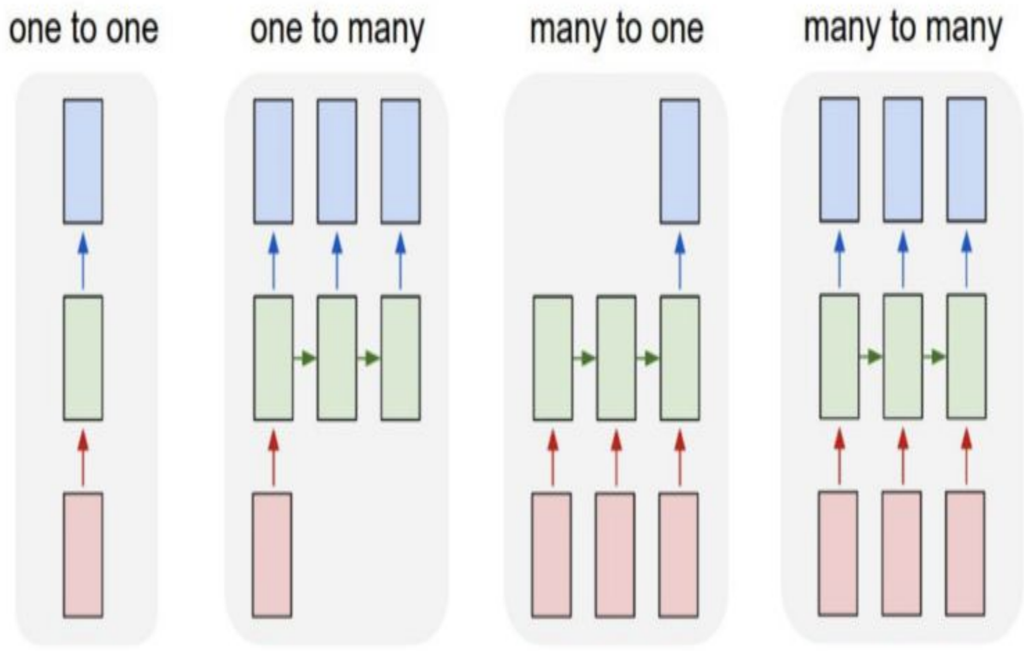
A, B and C are the parameters



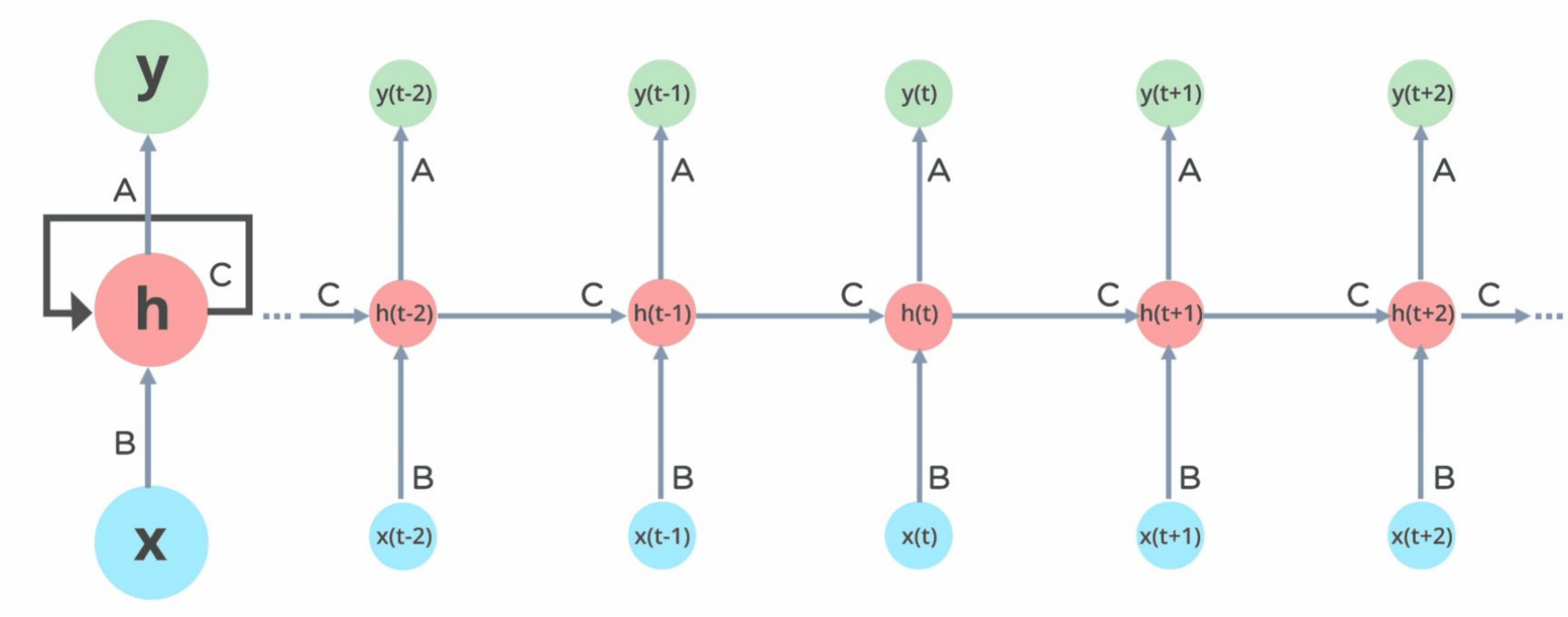
$h(t) = f_c\,(h(t-1),\, x(t))$

$h(t)$ = new state
f_c = function with parameter c
$h(t-1)$ = old state
$x(t)$ = input vector at time step t


Input
Output
Hidden
Hidden
Feed-forward Network
Hidden
Hidden
Recurrent Network
V7


one to one
one to many
many to one
many to many

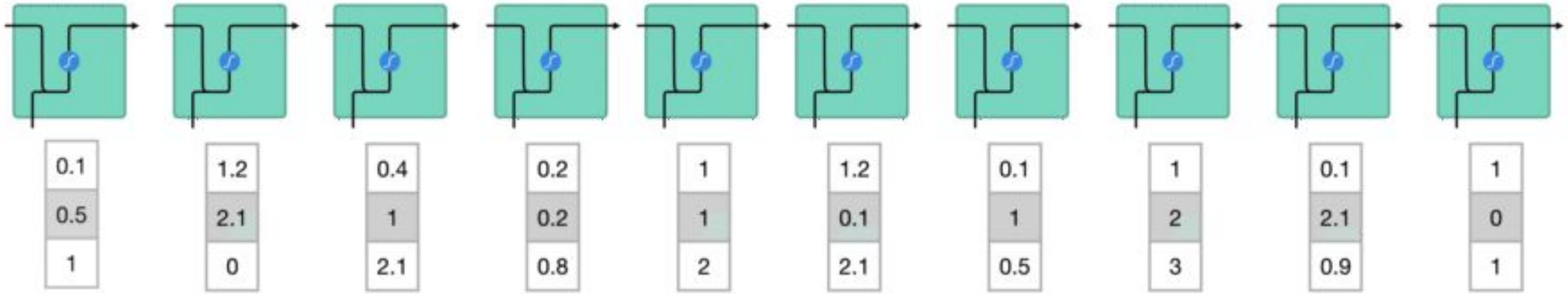
- The nodes in different layers of the neural network are compressed to form a single layer of recurrent neural networks.
- **A, B, and C are the parameters of the network.**
- Here, "x" is the input layer, "h" is the hidden layer, and "y" is the output layer.
- A, B, and C are the network parameters used to **improve the output of the model.**
- At any given **time t**, the current input is a combination of input at **x(t) and x(t-1).**
- The output at any given time is **fetched back** to the network to **improve on the output**.


y
A
h
C
B
x
y(t-2)
y(t-1)
y(t)
y(t+1)
y(t+2)
A
A
A
A
A
C
C
C
C
C
C
h(t-2)
h(t-1)
h(t)
h(t+1)
h(t+2)
B
B
B
B
B
x(t-2)
x(t-1)
x(t)
x(t+1)
x(t+2)

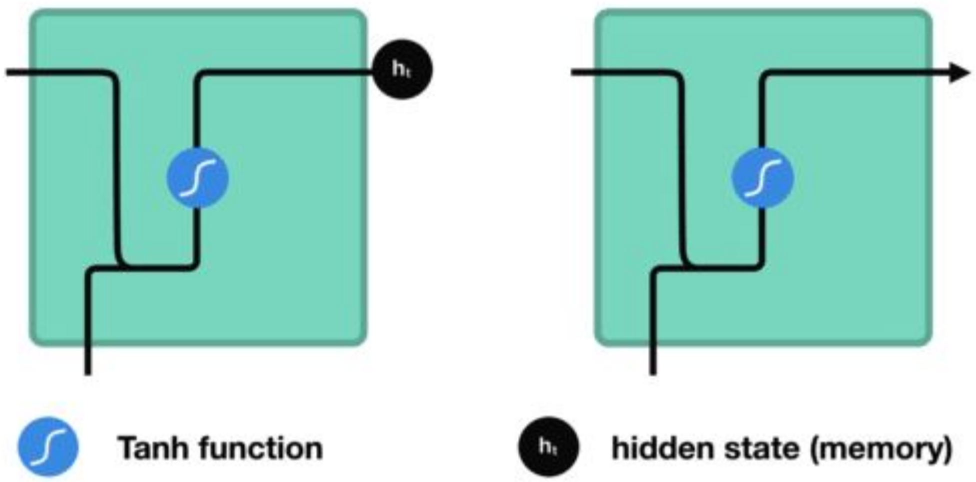
- The input layer **'x' takes in the input to the neural network** and processes it and passes it onto the **middle layer**.
- The middle layer **'h' can consist of multiple hidden layers**, each with its own ***ACTIVATION FUNCTIONS AND WEIGHTS AND BIASES.***
- The Recurrent Neural Network will standardize the **different activation functions and weights and biases**
- Then, instead of creating multiple hidden layers, it will create **one and loop over it as many times as required**.

Need to know about RNN

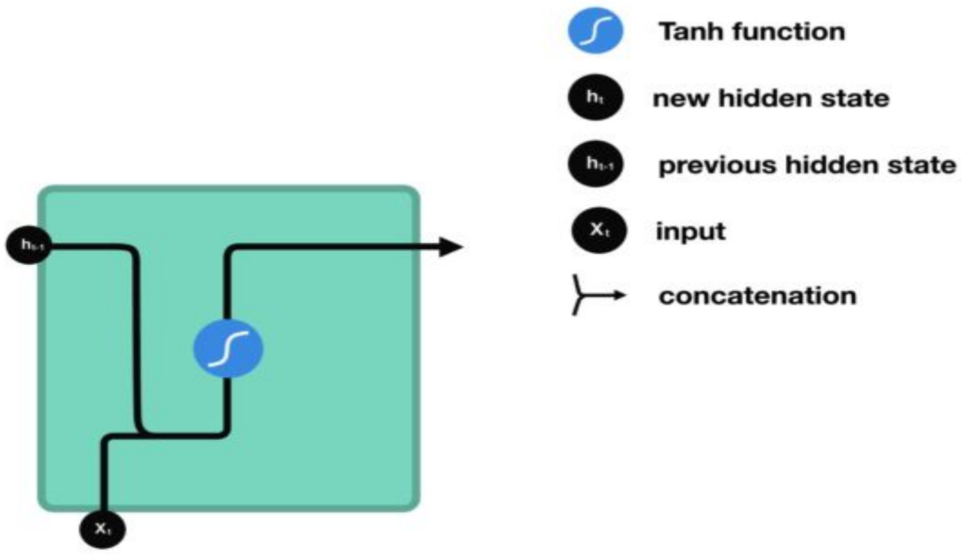
- First words get transformed into **machine-readable vectors.**
- Then the RNN processes the sequence of vectors one by one



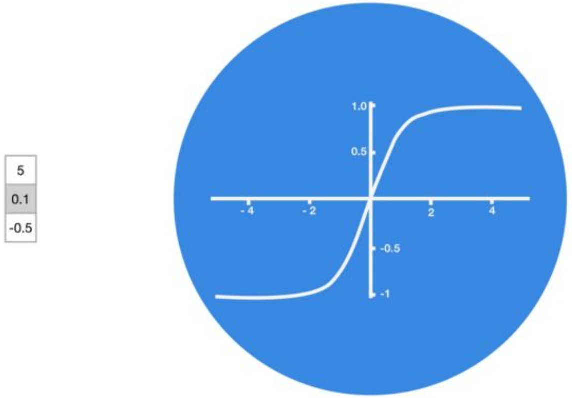
- While processing, it passes the **previous hidden state** to the **next step of the sequence**.
- The hidden state acts as the **NEURAL NETWORKS MEMORY**.
- It holds information on previous data the network has seen before



- First, the **input and previous hidden state** are combined to form a vector.
- That **vector** now has information on **the current input and previous inputs.**
- The vector goes through the **tanh activation**, and the output is the new hidden state, or the memory of the network.



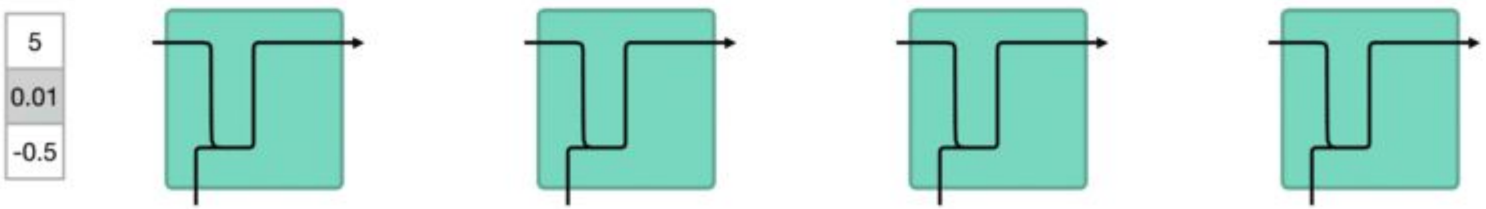
- **Tanh activation**
- The tanh activation is used to help **regulate the values** flowing through the network.
- The tanh function squishes values to always be between
- **-1 and 1.**

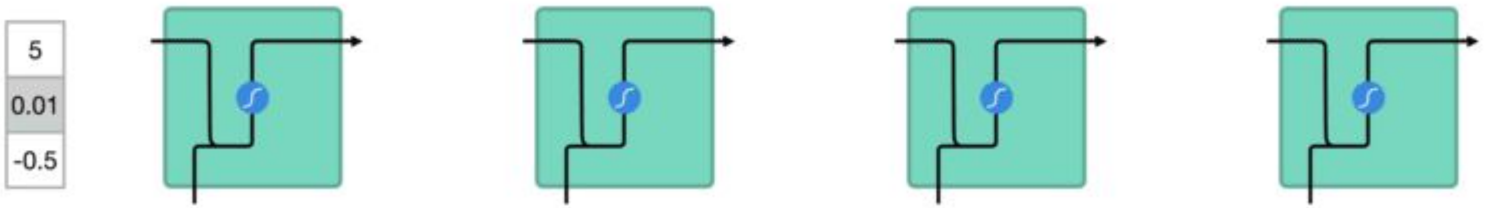


- When vectors are flowing through a neural network, it undergoes **many transformations due to various math operations.**

- A tanh function ensures that the values stay between **-1 and 1**, thus regulating the output of the neural network.
- You can see how the same values from above remain between the boundaries allowed by the tanh function.

- vector transformations without tanh

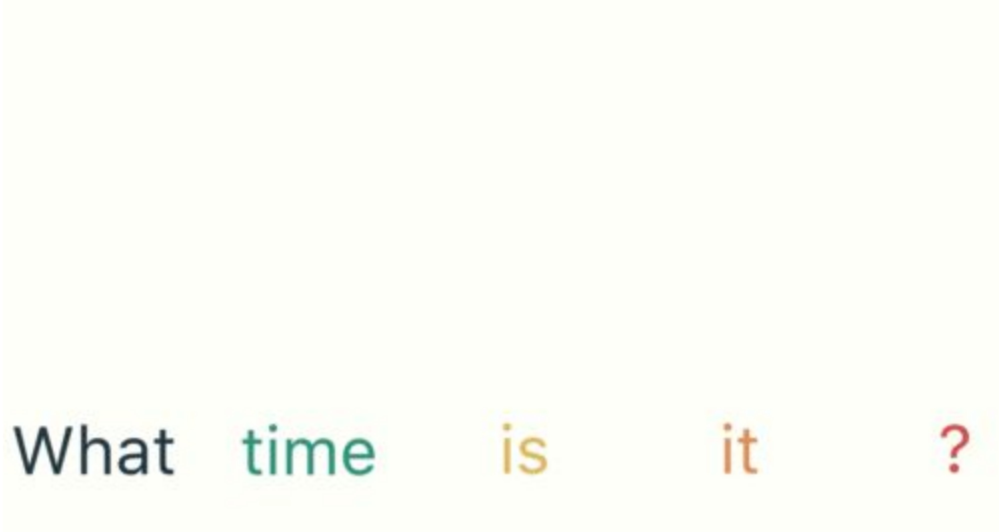


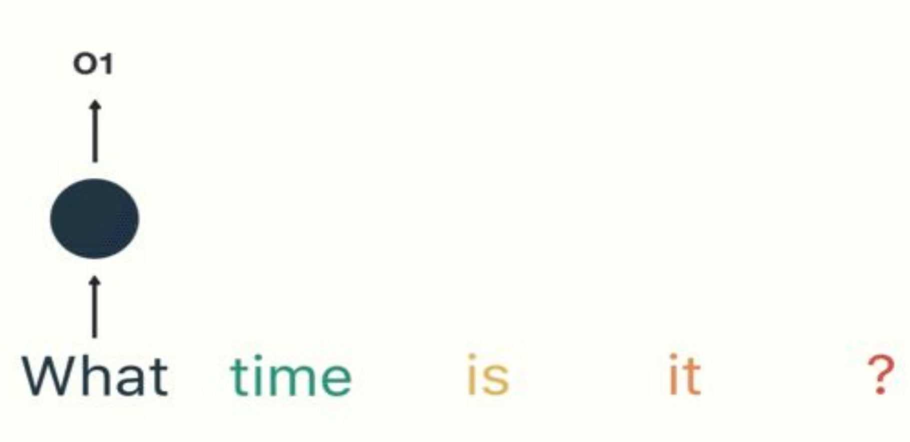


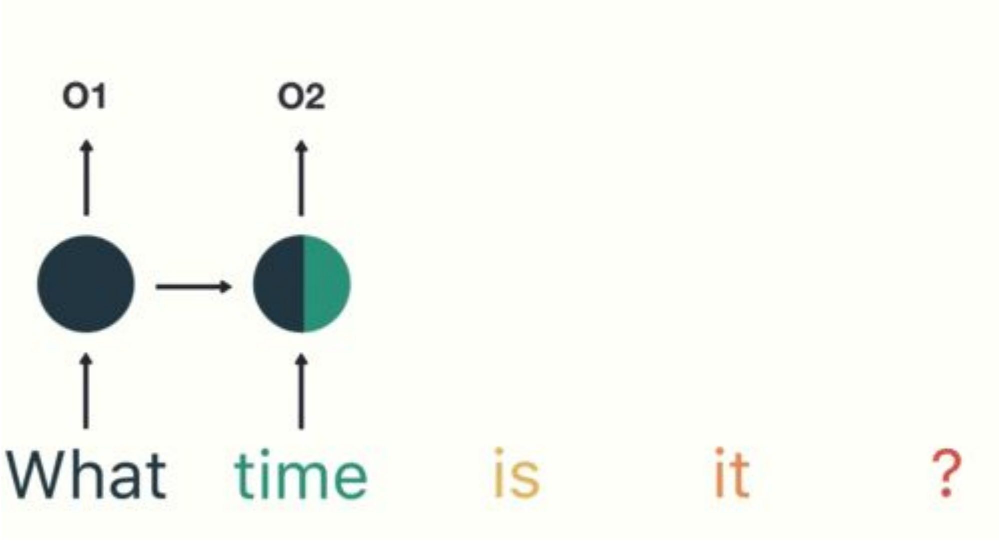
vector transformations with tanh

Example


What time is it?


What time is it ?


O1
What time is it ?


O1 O2
What time is it ?

Process of the example

Input Sequence

- tokenized into **individual words or characters**
- Each word is **converted into a vector** (using word embeddings or one-hot encoding

Hidden State Update

- **T1:** RNN processes the word "**What**", updates the hidden state h1.
- **T2:** the RNN processes the word "**time**", which, **combined with the hidden state from t1** updates the hidden state h2

Hidden State Update

- **T3**:RNN processes "is", **updating the hidden state to h3**
- **T4**: RNN processes "it", **updating the hidden state to h4**
- **T5**: RNN processes "?", and updates **the final hidden state h5**

Process of the example

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Understanding the Question

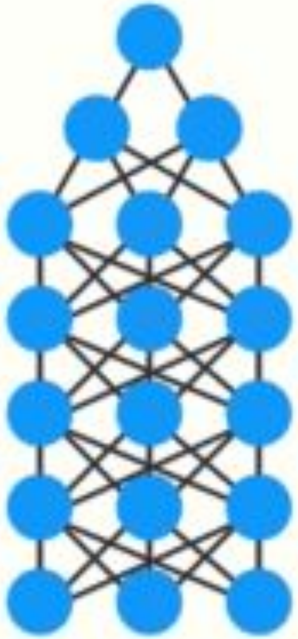
- RNN has processed the entire input sequence, **the hidden state contains a representation of the whole sentence**.
- This hidden state holds the **context of the sentence "What time is it?",** meaning the RNN understands this is a question asking for the current time.

Output Prediction

- The RNN can generate an output.

Issues and solution in RNN

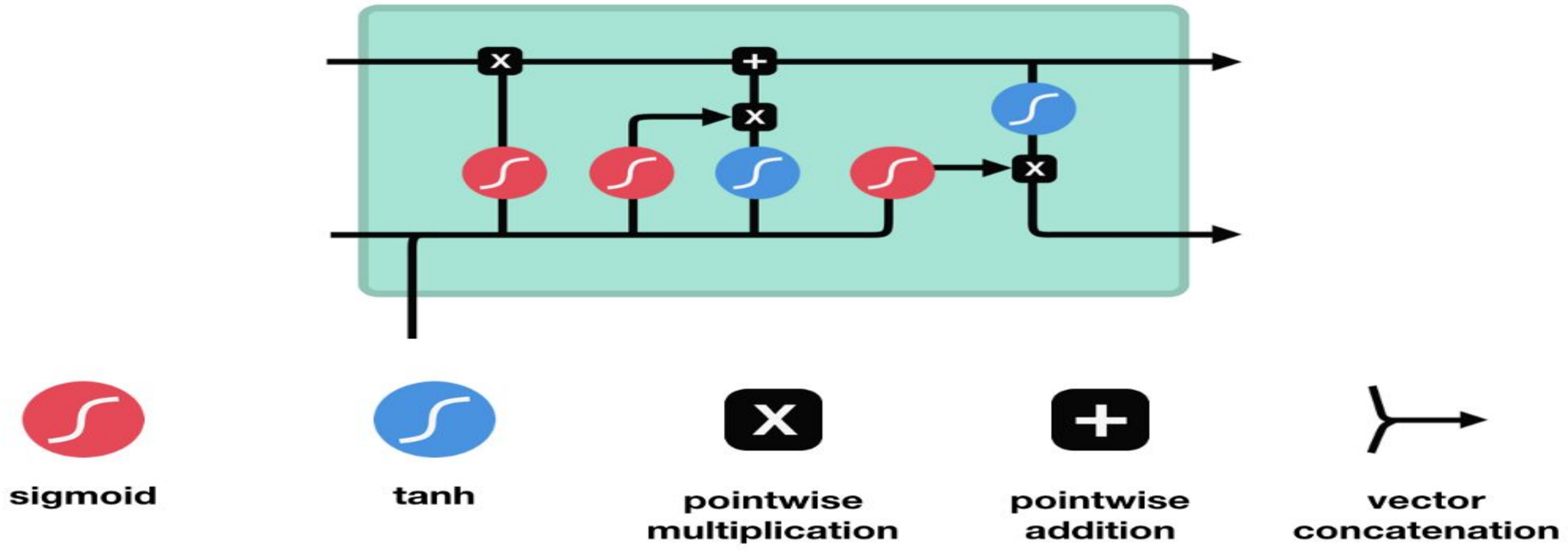
- **FORWARD PASS AND MAKES A PREDICTION**
- Using **loss function** it compares the **ground truth(Real or True value)**
- Loss function contains **ERROR VALUE** explains the bad performance of the network.
- With error value **back propagation** to evaluate the **GRADIENT for each node in the network**
- Gradient is value to do some **ADJUSTMENTS IN WEIGHTS** makes the network to learn the issues.
- Gradient gets **shrink when it carries back propagation** (**VANISHING GRADIENT PROBLEM**)
- **Small gradients means small adjustments**
- To overcome above issues the RNN uses LSTM(**Long Short Term Memory:**input, output, and forget**)**
- GRU(**Gated Recurrent Unit:update gate and a reset gate.)**



- A gradient measures how much ***the output of a function changes if you change the inputs a little bit***
- A gradient simply measures the change in all **weights** with regard to the **change in error**
- **VANISHING GRADIENT** occurs when the gradient is **smaller than expected**. It causes the earlier layers to start degrading before the later ones do, causing a decrease in the overall learning rate of that subset of layers.(LSTM, GRU)
- **EXPLODING GRADIENT** can happen when a gradient is **too big and creates an unstable model**. This means the weight of the model will be large and they may appear as "NaN".(RMSPROP:Root Mean Squared Propagation)

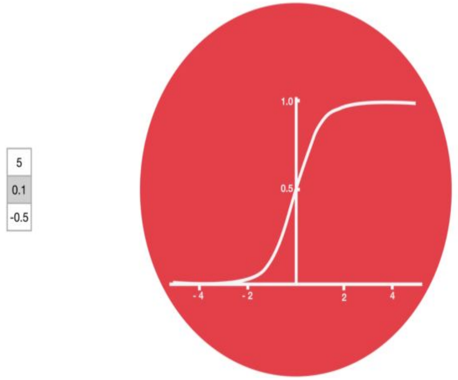
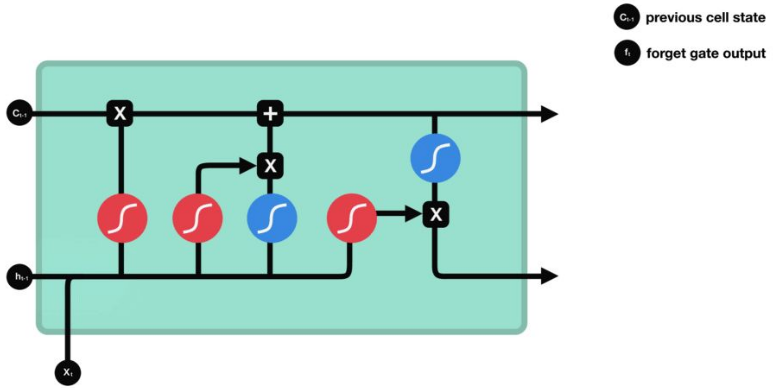
LSTM

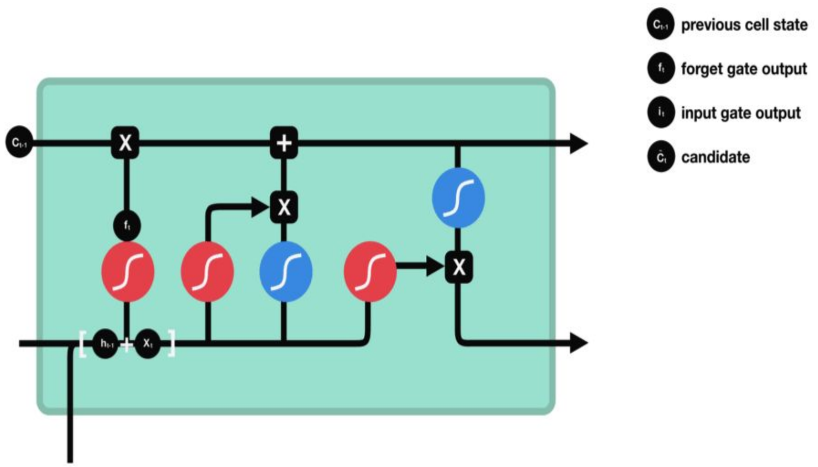
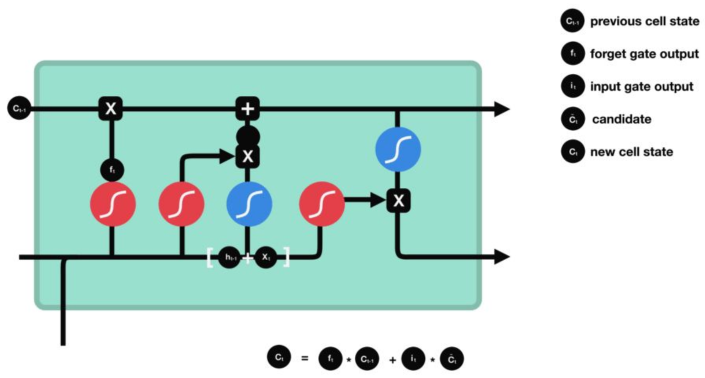
- An LSTM has a similar **control flow as a recurrent neural network.**
- It **processes data passing on information** as it propagates forward.
- The differences are the operations within the LSTM's cells.

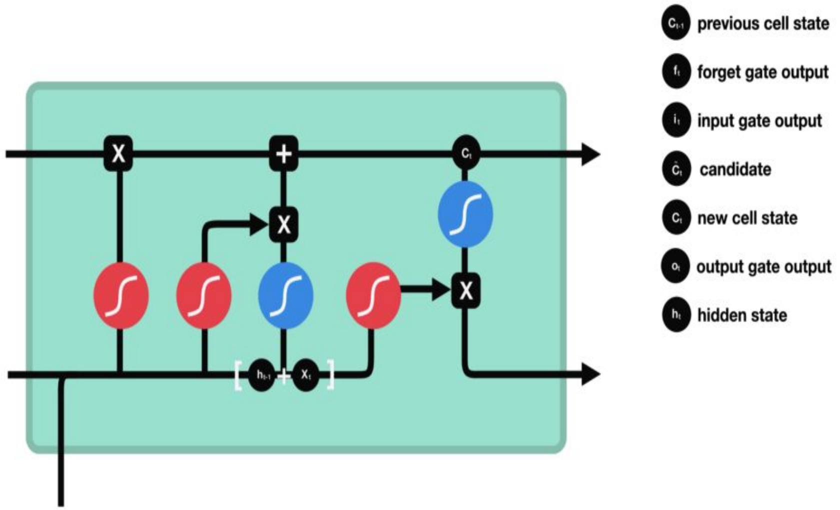


Core concept of LTSM

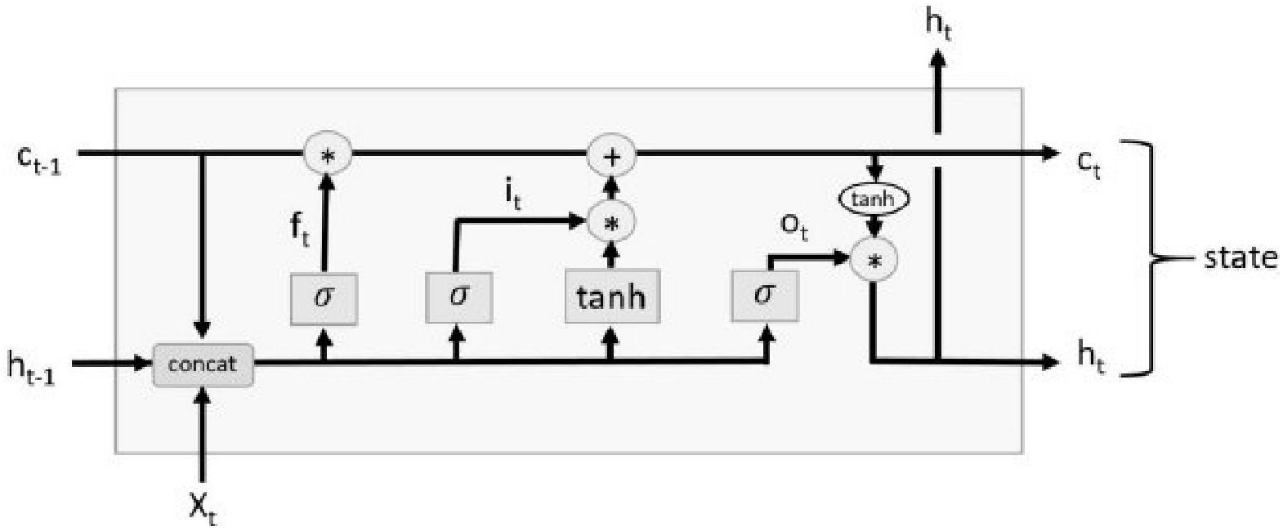
- The core concept of LSTM’s are the **cell state, and it’s various gates**.
- The cell state act as a transport highway that **transfers relative information all the way down the sequence chain**.
- You can think of it as the **“MEMORY” OF THE NETWORK**.
- The cell state, in theory, can carry **RELEVANT INFORMATION** throughout the processing of the sequence.
- So even information from the earlier time steps can make it’s way to later time steps, reducing the effects of short-term memory.
- As the cell state goes on its journey, information **get’s added or removed to the cell state via gates.**
- The gates are different neural networks that decide which **information is allowed on the cell state**.
- The gates can learn what information is **relevant to keep or forget during training.**

Sigmoid	• Gates contains **sigmoid activations**. • A sigmoid activation is similar to the tanh activation. • Instead of squishing values **between -1 and 1**, it squishes values **between 0 and 1.** • That is helpful to **update or forget data** because any number getting multiplied by 0 is 0, **causing values to disappears or be "forgotten."**	
Forget gate	• This gate decides what **information should be thrown away or kept**. • Information from **the previous hidden state** **and** **information from the current input** **is passed through the sigmoid function.** • Values come out between **0 and 1**. • The closer to **0 means to forget,** and the closer to 1 means to keep.	

Input Gate	• To **UPDATE THE CELL STATE**, we have the input gate. • First, we pass the **previous hidden state and current input** into a sigmoid function. • That decides which values will be updated by transforming the values to be **between 0 and 1.** • **0 means not important, and 1 means important**.	
Cell State	• The cell state gets **pointwise multiplied by the forget vector.** • This has a possibility of dropping values in the cell state if it gets multiplied by values near 0	

Output Gate	• The output gate decides what **the next hidden state should be.** • Remember that the **hidden state contains information on previous inputs.** • The hidden state is **also used for predictions.** • First, we pass the previous hidden state and the current input into a sigmoid functions. • Then we pass the **newly modified cell state to the tanh function**. • We multiply the **tanh output with the sigmoid output** to decide what information the hidden state should carry. • The output is **the hidden state**. • The new cell state and the **new hidden is then carried over to the next time step.**	

- LSTM information flow, adapted from Christopher Olah's famous blog post to fit equations in the text



$$i_t = \sigma(W^{(xi)}x_t + W^{(hi)}h_{t-1} + W^{(ci)}c_{t-1} + b^{(i)})$$

$$f_t = \sigma(W^{(xf)}x_t + W^{(hf)}h_{t-1} + W^{(cf)}c_{t-1} + b^{(f)})$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W^{(xc)}x_t + W^{(hc)}h_{t-1} + b^{(c)})$$

$$o_t = \sigma(W^{(xo)}x_t + W^{(ho)}h_{t-1} + W^{(co)}c_t + b^{(o)})$$

$$h_t = o_t \cdot \tanh(c_t)$$

$$\sigma(\mathrm{concat}(x, h, c)) = \sigma(W[x, h, c] + b) = \sigma(W^{(x)}x + W^{(h)}h + W^{(c)}c + b)$$

Steps for below code

1. Start to Initialize **h, c to o**
2. Set initial word (**passing the text**)
3. Carry the iteration for word process (inside iteration carry the below steps)
   - **Convert word to vector** (Xvec)
   - Compute LSTM gates (i, f, c, o, h) (I- INPUT, F-FORGET,O-OUTPUT,H-HIDDEN,C-CELL STATES)
   - Predict next possible **words from h**
   - Randomly select **one word from top**
   - Append the **chosen word to sentence**
   - **Update hidden state h** and cell state c
   - **Repeat the above Process** until all the words are formed
   - Print the complete sentence as a final output

4. END

- Microsoft Cognitive Toolkit (formerly known as the Computational Network Toolkit CNTK)
- Notations used: trained W and b arrays and two other arrays that together define its structure
- arrays into the Python version of the RNN

def rnn(word, old_h, old_c): **Xvec = getvec(word, E)** i = **Sigmoid(**np.matmul(WXI, Xvec) + np.matmul(WHI, old_h) + WCI*old_c + bI)	input a text string for the input word, while the equations take a vector encoding of the word as input I- INPUT, F-FORGET,O-OUTPUT,H-HIDDEN,C-CELL STATES

f = Sigmoid(np.matmul(WXF, Xvec) + np.matmul(WHF, old_h) + WCF*old_c + bF)
c = f*old_c + i*(np.tanh(np.matmul(WXC, Xvec)
+
np.matmul(WHC, old_h) +
bC))
o = Sigmoid(np.matmul(WXO, Xvec)+ np.matmul(WHO, old_h)+ (WCO * c)+
bO)
h = o * np.tanh(c)
# Extract ordered list of five best possible next words 	
q = h.copy()
q.shape = (1, 200)
output = np.matmul(q, W2)
outlist = getwordsfromoutput(output)
return h, c, outlist
	1. The RNN training generated the **encoding matrix E**, which has the nice property that the ith column of the matrix corresponds to the word in the ith position in the vocabulary List.
	2. The function **getvec(word, E)** takes the embedding tensor E, looks up the **position of the word in the vocabulary list**, and returns the column vector of E that corresponds to that word.
	3. The output of **one pass through the LSTM cell is the vector h**

c = np.zeros(shape = (200, 1)) h = np.zeros(shape = (200, 1)) output = np.zeros(shape = (10000, 1)) word = 'my' sentence= word for _ in range(40): h, c, outlist = rnn(word, h, c) word = outlist[randint (0,3)] sentence = sentence + " " +word print(sentence+".")	LSTM is truly a recurrent network, we provide the network with a starting word
my new rules which would create an interest position here unless there should prove signs of such things too quickly although the market could be done better toward paying further volatility where it would pay cash around again if everybody can.	
the company reported third-quarter results reflecting a number compared between N barrels including pretax operating loss from a month	following fiscal month ending july earlier compared slightly higher while six-month cds increased sharply tuesday after an after-tax loss reflecting a strong.

- Scenario :Taking customers review
- Input :Excellent! The product gave me a proper results, as all things should be. I didn't use it completely but for sure ill recommend to my friends
- *Convert to machine understanding values*
- *Apply using LSTM logic*
- *Finally the Model remembers*
- Output: Excellent! The product gave me a proper results, as all things should be. I didn't use it completely but for sure ill recommend to my friends

Amazon MXNet Virtual Machine Image

- MXNet is an open source library for distributed parallel machine learning.
- Amazon Deep Learning AMI which includes not only MXNet but also CNTK and TensorFlow, plus other good toolkits

```
>cd .jupyter
>openssl req -x509 -nodes -days 365 -newkey
rsa:1024 \
-keyout mykey.key -out mycert.pem
>ipython
[1]: from notebook.auth import passwd
[2]: passwd()
Enter password :
Verify password :
Out[2]: 'sha1:---- long string -----------'
```

Amazon Deep Learning AMI

Configuring the Amazon AMI to use Jupyter

Deep Learning AMI (p2.16xlarge instances, m4.2xlarge)

password, and copy the long sha1 string

https://ipaddress:8888 in your browser, where ipaddress is the external IP address of your virtual machine

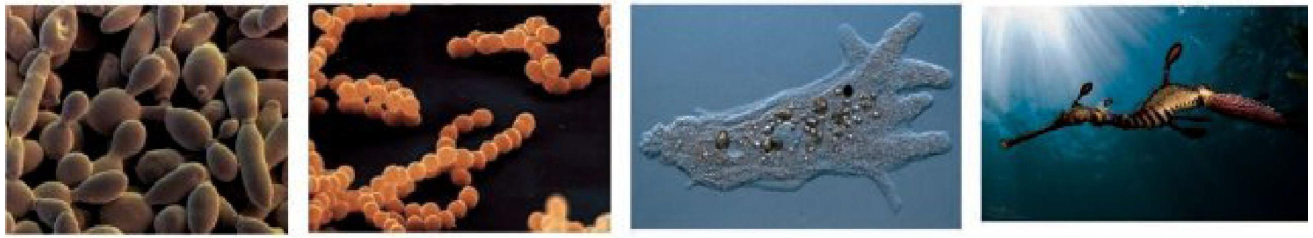
DEEP RESIDUAL LEARNING

Steps for below code

1. Start the process
2. Load **synsets and model data** (full-synset.txt, ResNet-152checkpoint)
3. Bind model to GPU context and set model parameters **((GPU) and initializing its weights using the preloaded parameters**)
4. Preprocess the image to shape(**input image is resized, normalized, and shaped to fit the model's expected input dimensions**)
5. Forward image through the model to compute predictions **ResNet-152 is a deep convolutional neural network (CNN) architecture**

6. Extract **output probabilities from model predictions**
7. Sort probabilities in **descending order**
8. Select **top 5 predictions**
9. Display top 5 classes and their probabilities
10. END

- resent-152-symbol.json, a complete description of the **network as a large json file**
- resnet-152-0000.params, a binary file containing all parameters for **the trained model**
- synset.txt, a text file containing the **1,121 image labels, one per line**



Yeast Streptococcus Amoeba Seahorse

```
import mxnet as mx
# 1) Load the pretrained model data
with open('full-synset.txt','r') as f:
synsets = [l.rstrip() for l in f]
sym,arg_params,aux_params=
mx.model.load_checkpoint('full-resnet-152',0)
# 2) Build a model from the data
mod = mx.mod.Module(symbol=sym, context=mx.gpu())
mod.bind(for_training=False, data_shapes=[('data', (1,3,224,224))])
mod.set_params(arg_params, aux_params)
# 3) Send JPEG image to network for prediction
mod.forward(Batch([mx.nd.array(img)]))
prob = mod.get_outputs()[0].asnumpy()
prob = np.squeeze(prob)
a = np.argsort(prob)[::-1]
for i in a[0:5]:
print('probability=%f, class=%s' %(prob[i],
synsets[i]))
```

Google TensorFlow in the Cloud

- Google's TensorFlow is a frequently discussed and used deep learning toolkit.
- **Logistic Regression Model**

```
import tensorflow as tf
import numpy as np
import csv
sess = tf.InteractiveSession()
x = tf.placeholder(tf.float32, shape=(None,3))
y = tf.placeholder(tf.float32, shape =(None,1))
# Set model weights
W = tf.Variable(tf.zeros([3, 1]))
b = tf.Variable(tf.zeros([1]))
pred = tf.sigmoid(tf.matmul(x, W) + b)
cost = tf.sqrt(tf.reduce_sum((y - pred)**2/batch_size))
opt = tf.train.AdamOptimizer() // stochastic gradient descent
method
optimizer = opt.minimize(cost)
```

$$cost = \sum_{i=0}^{1}(y - \sigma(W \cdot x + b))^2$$

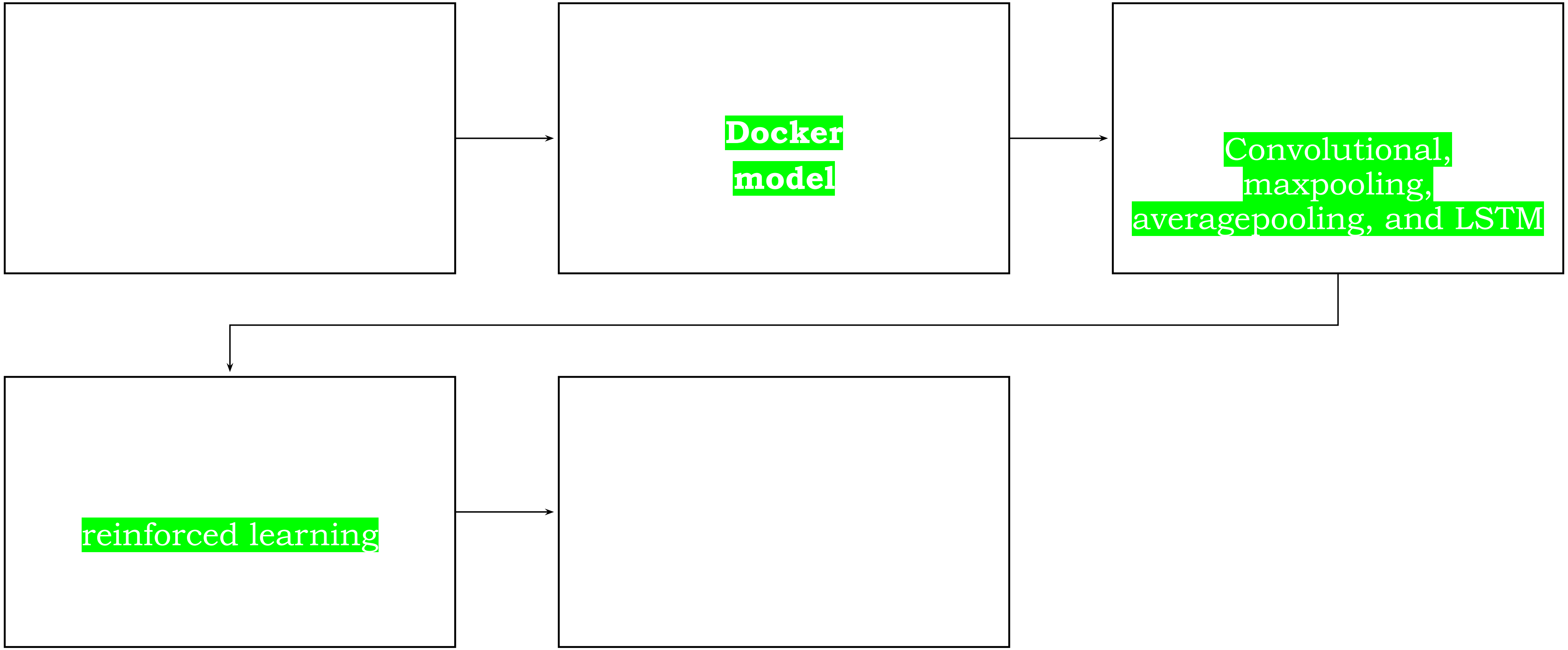
```
training_epochs = 100000
batch_size = 100
display_step = 1000
init = tf.initialize_all_variables()

sess.run(init)
# Training cycle
for epoch in range(training_epochs):
    avg_cost = 0.
    total_batch = int(len(train_data)/batch_size)
    # Loop over all batches
    for i in range(total_batch):
        batch_xs,batch_ys=get_batch(batch_size,train_data,train_label)
        # Fit training using batch data
        _,c=sess.run([optimizer,cost],
                     feed_dict={x:batch_xs, y:batch_ys})
        # Compute average loss
        avg_cost += c / total_batch
    # Display logs per epoch step
    if (epoch+1) % display_step == 0:
        print("Epoch:", '%04d' % (epoch+1), "cost=", str(avg_cost))
```

Steps for above code

1. Import **TensorFlow and NumPy**
2. Define **input data and labels**
3. Initialize model parameters (**Weights W, Bias b**)
4. Define **logistic regression model** using **sigmoid function**
5. Define cost function (**RMSE) between predicted and actual**
6. Initialize **Adam optimizer** to minimize the cost function
7. ***Initialize all variables (W, b,etc***)
8. Initiate the loop for **carrying iteration**
9. Calculate the **loss function**
10. END

Microsoft Cognitive Toolkit

- CNTK-CPU-InfiniBand-IntelMPI for **execution across multiple InfiniBand RDMA VMs**
- CNTK-CPU-OpenMPI for **multi-instance VMs**
- CNTK-GPU-OpenMPI for multiple **GPU-equipped servers** such as the NC class, which have 24 cores and 4 K80 NVIDIA GPUs


Docker model
Convolutional, maxpooling, averagepooling, and LSTM
reinforced learning

Steps in below code

1. Begin the process.
2. **Define Input Variables**: X, M, B (***input variables for the model X- input data to the computation graph, M-Matrix Variable for matrix multiplication, B-Bias Variable***)
3. **Define Computation Graph: Y = cntk.times(X, M) + B** // ***Specify how the inputs are processed to produce outputs*** (Represents the computation graph. cntk.times(X, M) performs matrix multiplication, and + B adds the bias term.)
4. **Input Values: x, m, b** // Provide actual data for evaluation
5. Evaluate Computation Graph result = Y.eval({X: x, M: m, B: b})// **Compute the output based on provided values**.
6. **Output Result:** array

Steps in below code

7. **Define CNN Model** (Convolutional Layers, MaxPooling, Dense Layers) for **feature extraction and classification**.
8. **Define RNN Model**(Embedding Layer ,LSTM Layer, Dense Output Layer) to handle sequential data and perform tasks like text classification.
9. END

```
import numpy as np
import cntk
X = cntk.input_variable((1,2))
M = cntk.input_variable((2,3))
B = cntk.input_variable((1,3))
Y = cntk.times(X,M)+B
```

- X is a 1×2-dimensional tensor, that is, a vector of length 2; M is a 2×3 matrix; and B is a vector of length 3.
- The expression **Y=X*M+B** yields a vector of length 3.
- However, no computation has taken place at this point: we have only constructed a graph of the computation.
- To execute the graph, we input values for X, B, and M, and then apply the eval operator on Y

```
x = [[ np.asarray([[40,50]]) ]]
m = [[ np.asarray([[1, 2, 3], [4, 5, 6]]) ]]
b = [[ np.asarray([1., 1., 1.])]]
print(Y.eval({X:x, M: m, B: b}))
----- output -------------
array([[[[ 241., 331., 421.]]]], dtype=float32)
```

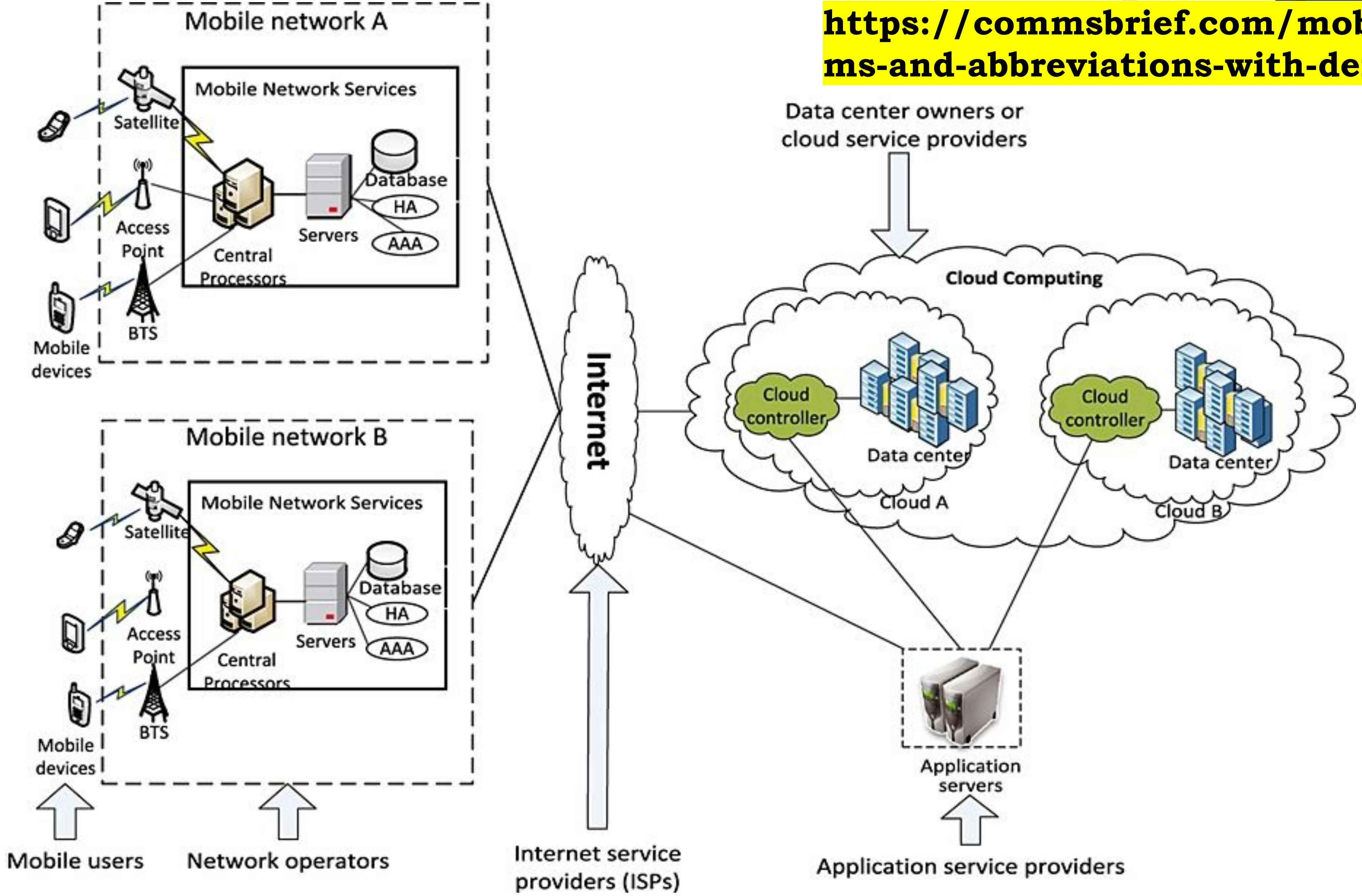
- Convolutional, MaxPooling, AveragePooling, and LSTM.
- Layers can also be stacked with a simple operator called sequential.

```
with default_options(activation=relu):
conv_net = Sequential ([
# 3 layers of convolution and dimension reduction by pooling
Convolution((5,5),32,pad=True),
MaxPooling((3,3), strides=(2,2)),
Convolution((5,5),32,pad=True),
MaxPooling((3,3), strides=(2,2)),
Convolution((5,5),64,pad=True),
MaxPooling((3,3), strides=(2,2)),
# 2 dense layers for classification
Dense(64),
Dense(10, activation=None)])
model = Sequential ([Embedding(150), # Embed into a
150-dimensional vector
Recurrence(LSTM(300)), # Forward LSTMDense(labelDim) # Word-wise
classification])
```

- The **Sequential operator** used in the same code can be thought of as a **concatenation of the layers** in the given sequence.
- The Recurrence operator is used to wrap the correct LSTM output back to the input for the next input to the network.
- The Cortana cognitive services

Wireless Internet and Mobile Cloud Computing

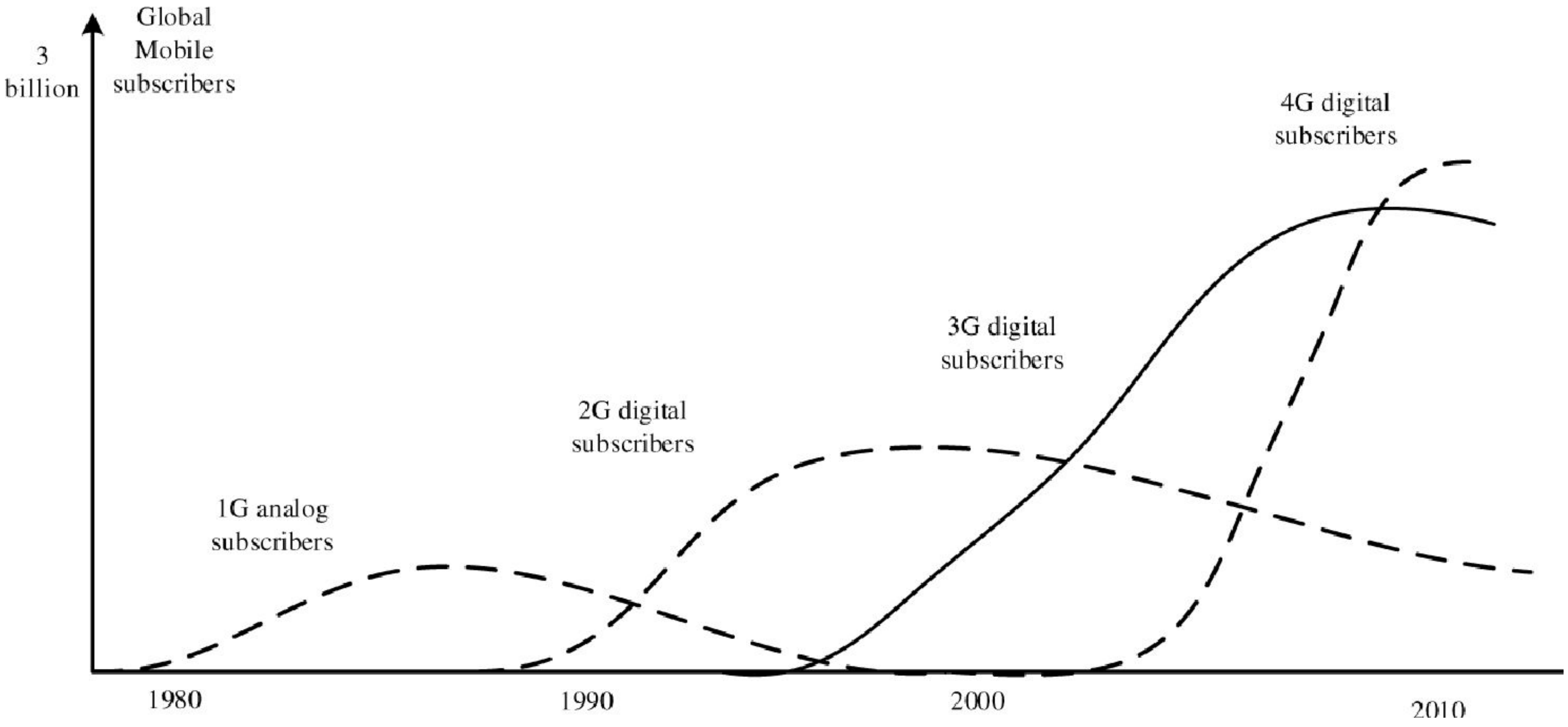
https://commsbrief.com/mobile-telecom-acronyms-and-abbreviations-with-definitions/



- Frequency Division Multiplexing (FDM)
- Time Division Multiplexing (TDM)
- Code Division Multiplexing (CDM)
- Space Division Multiplexing (SDM)

4G	Fourth Generation
AAA	Authentication, Authorization, Accounting
APDV	Application Protocol Data Unit
API	Application Programing Interface
ARM	Advanced RISC Machine
AV	Antivirus
B2B	Business to Business
B2C	Business to Customer
BTS	Base Transceiver Station
CC	Cloud Computing
CSP	Cloud Service Provider
EC2	Elastic Compute Cloud
GPS	Global Positioning System
HA	Home Agent
IaaS	Infrastructure as a Service
IA	Integrated Authenticated
ID	Identifier
IMERA	French acronym for Mobile Interaction in Augmented Reality Environment
ISP	Internet Service Provider
IRNA	Intelligent Radio Network Access
JME	Java ME, a Java platform
LBS	Location Base Service
LTE	Long Term Evolution
LTS	Location Trusted Server
MAUI	Memory Arithmetic Unit and Interface
MC	Mobile Computing
MCC	Mobile Cloud Computing
MDP	Markov Decision Process
MSC	Mobile Service Cloud
P2P	Peer-to-Peer
PaaS	Platform as a Service
QoS	Quality of Service
RACE	Resource-Aware Collaborative Execution
REST	Repretational State Transfer
RFS	Random File System
RTP	Real-time Transport Protocol
S3	Simple Storage Service
SaaS	Software as a Service
TCC	Truster Crypto Coprocessor
URI	Uniform Resource Identifier

Mobile Devices and Internet Edge Networks



Mobile Core Networks

- A mobile core network is a central part of the overall mobile network that **allows subscribers** to get access to the services that they are entitled to use.
- It is responsible for critical functions such as **subscriber profile information, location, service authentication and necessary switching tasks.**

Generation	1G	2G	3G	4G	5G
Radio and Networks Technology	Analog phones, AMPS, TDMA	Digital phones, GSM, CDMA	CDMA2000, WCDMA, and TD-SCDMA	LTE, OFDM, MIMO, software-steered radio	LTE, Cloud-based RAN
Peak Mobile Data Rate	8 Kbps	9.6 ~ 344 Kbps	2 Mbps	100 Mbps	10 Gbps–1 Tbps
Driving Applications	Voice Communication	Voice/Data Communication	Multimedia Communication	Wideband Communication	Ultra-speed Communication

Mobile core networks

- Cellular radio access networks (RANS)
- Global system for mobile communications (GSM)
- Code division multiple access (CDMA)
- Orthogonal frequency-division multiplexing (OFDM)
- Multiple-input and multiple-output (MIMO)
- Remote radio head (RRH)
- Cloud-based radio access networks (C-RAN).

Name	Description
Radio access network	• A radio access network (RAN) is the part of a mobile network that connects **end-user devices, like smartphones, to the cloud**. • This is achieved by sending information via **radio waves** from end-user devices to a RAN's transceivers, and finally from the transceivers to the core network which connects to the global internet.
Global system for mobile communication	• GSM stands for **Global System for Mobile Communication**. • It is a digital cellular technology used for **transmitting mobile voice and data services.** • It uses 4 different frequency bands of 850 MHz, 900 MHz, 1800 MHz and 1900 MHz . It uses the combination of FDMA and TDMA.
Code Division Multiple Access	• Code Division Multiple Access (CDMA) is a sort of **multiplexing that facilitates various signals to occupy a single transmission channel**. • It optimizes the use of **available bandwidth**. • The technology is commonly used in **ultra-high-frequency (UHF)** cellular telephone systems, bands ranging between the 800-MHz and 1.9-GHz.
Orthogonal Frequency Division Multiplexing	• OFDM, Orthogonal Frequency Division Multiplexing is a form of **signal waveform or modulation that provides some significant advantages for data links.** • Accordingly, OFDM, Orthogonal Frequency Division Multiplexing is used for many of the latest wide bandwidth and high data rate wireless systems including Wi-Fi, cellular telecommunications and many more

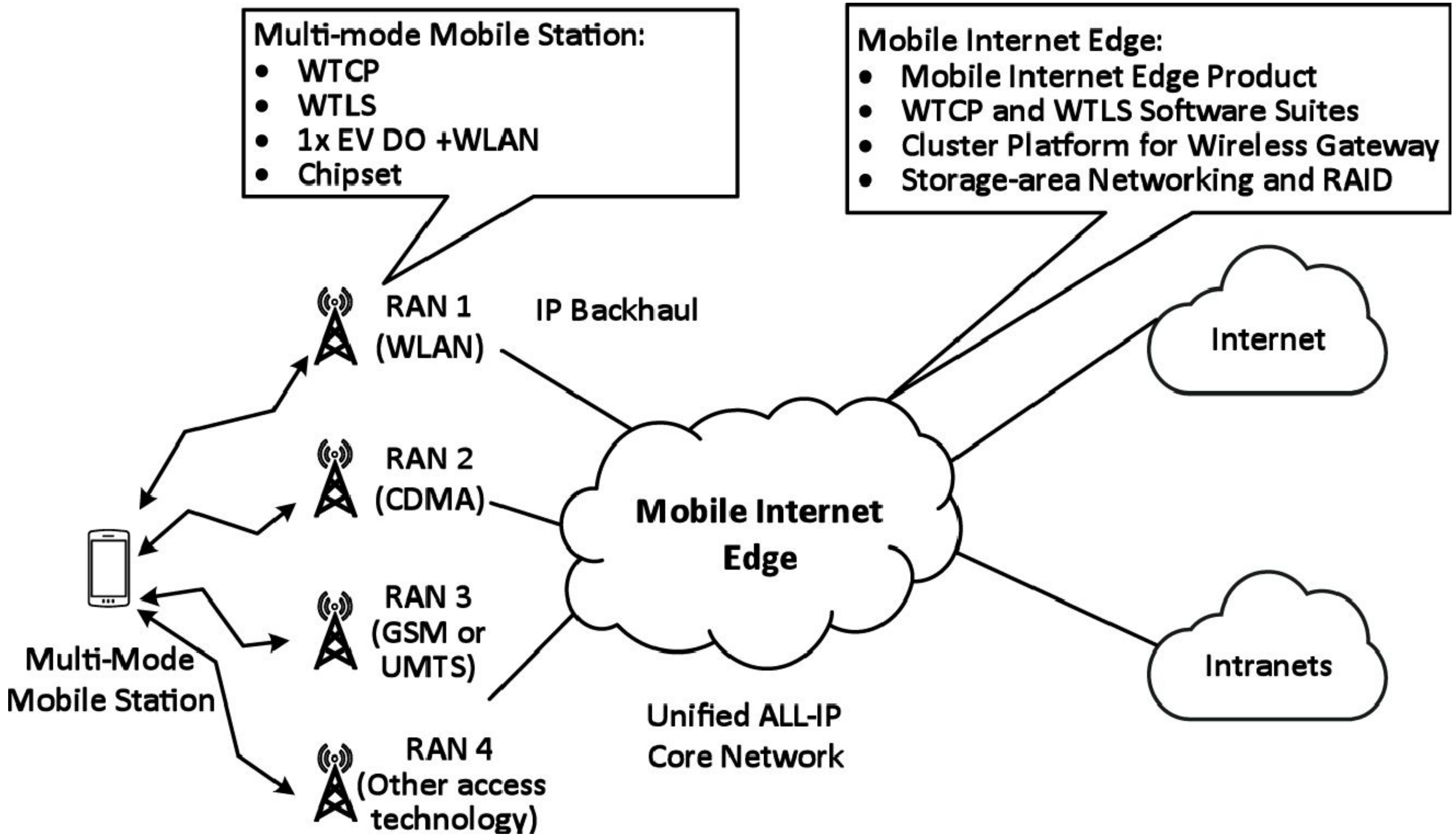
Name	Description
Multiple Input Multiple Output	MIMO: **Multiple Input Multiple Output** technology uses multiple antennas to make use of **reflected signals to provide gains in channel robustness and throughput**.
Remote Radio Head	• A **Remote Radio Head (RRH) or Remote radio unit (RRU)** is the RF circuitry of a base station enclosed in a small outdoor module. • The RRH performs all RF functionality **like transmit and receive functions, filtering, and amplification**. • It also contains **analog-to-digital or digital-to-analog converters** and up/down converters.
Cloud Radio Access Network	• Cloud/Centralized Radio Access Network (**C-RAN**) is a next-generation RAN architecture in which the **Baseband Units (BBUs)** are moved from cell sites to a centralized location. • The centralization of **BBUs also enables network virtualization, in which the BBU hotel can be replaced by a server**.

Mobile Internet Edge Networks

- Wireless local-area Network (WLAN)
- Wireless home-area network (WHAN)
- Personal-area Network (PAN)
- Body-area network (BAN)

Name	Description
Wireless local-area Network (WLAN)	• WLAN stands for **Wireless Local Area Network**. • WLAN is a wireless network that **allows devices to associate and communicate wirelessly.** • Just like a traditional wired LAN in which the device transmits over Ethernet cables, the devices using the WLAN transmit through WiFi. • Mobile users can connect to a WLAN through wireless connection. • The **IEEE 802.11 group** of standards describe the technologies for LANs
Wireless home-area network (HAN)	• **Home Area Network (HAN)** is a network in a user's home where all the laptops, computers, smartphones, and other smart appliances and digital devices are connected into a network. • This facilitates communication among the **digital devices within a home which are connected to the Home network.** • Home Area Network may be **wired or wireless**. • Mostly **wireless network is used for HAN**. • One centralized device is there for the function of **Network Address Translation (NAT).** • This Home Area Network enables communication and sharing of resources between the smart devices over a network connection.

Name	Description
Personal-area Network (PAN)	• **Personal Area Network (PAN)** is the computer network that connects computers/devices within the range of an individual person. • As PAN provides a **network range within a person's range typically within a range of 10 meters(33 feet)** it is called a Personal Area Network. • A Personal Area Network typically involves a computer, phone, tablet, printer, PDA (Personal Digital Assistant) and other and other entertainment devices like speakers, video game consoles, etc.
Body-area network (BAN)	• A **BAN (body area network) or a WBAN (wireless body area network)** is a wireless n/w of the wearable computing device. • These devices may be placed in the **human body or surface mounted on the human body in a particular position.** • The growth of attention in wearable technologies such as **glasses, watches have meant an improved focus on wireless networking.** • The term BAN (body area networks) have been invented to refer to the wireless network technology used in combination with wearables. • The main purpose of these networks is **to transmit data produced by wearable devices outside to a WLAN or the Internet**. In some cases, wearables can also exchange the data directly with each other


Multi-mode Mobile Station:
• WTCP
• WTLS
• 1x EV DO +WLAN
• Chipset
Mobile Internet Edge:
• Mobile Internet Edge Product
• WTCP and WTLS Software Suites
• Cluster Platform for Wireless Gateway
• Storage-area Networking and RAID
RAN 1 (WLAN)
IP Backhaul
RAN 2 (CDMA)
RAN 3 (GSM or UMTS)
RAN 4 (Other access technology)
Multi-Mode Mobile Station
Mobile Internet Edge
Unified ALL-IP Core Network
Internet
Intranets

- **Bluetooth Devices and Networks**
- **Wi-Fi Networks**
- **Wi-Fi, Bluetooth, and Wireless Sensor Networks**
  - *Wireless metropolitan area networks* (WMAN)
  - *Wireless local area networks* (WLAN)
  - *Wireless personal area networks* (WPAN)

Network Types	Cellular WAN	WMAN	WLAN	WPAN	WPAN
Market Name Standard	**GSM/GPRS** **CDMA/1XRTT**	**WiMaX** **802.15.6**	**Wi-Fi** **802.11n**	**ZigBee** **802.15.4**	**Bluetooth** **802.15.1**
Application focus	Wide Area Voice and Data	Data, Trans. Bandwidth	Web, E-mail, Video	Monitoring & Control	Cable Replace
Memory (MB)	18+	8+	1+	0.004–0.032	0.25+
Battery (days)	1–7	1–7	0.5–5	100–1000+	1–7
Network Size	1	1	32	2^{64} or more	7
Bandwidth (KBs)	64–128+	75,000	54,000+	20–250	720
Range (KM)	1000+	40 ~ 100	1–100	1–100+	1–10+
Success Metric	Coverage	Speed	Flexibility	Power, Cost	Low cost

MCC

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

- Mobile cloud computing (MCC) is the method of using cloud technology to deliver mobile apps.
- Complex mobile apps today perform tasks such as authentication, **location-aware functions**, and providing targeted content and communication for end users.
- Hence, they require extensive computational resources such as **data storage capacity, memory, and processing power.**
- Mobile cloud computing takes the pressure off mobile devices by harnessing the power of cloud infrastructure.
- Developers build and update rich mobile apps using cloud services and then deploy them for remote access from any device.

- **Frontend web and mobile on AWS offers** a broad set of tools and services to support development workflows for mobile application developers.
- With the speed and reliability of AWS infrastructure, you can develop, deploy, and secure your applications at the scale you require.
- For example, you can use the following AWS services and resources:

- **AWS Amplify** to build full-stack and scalable applications powered by AWS scalable applications with authentication, storage, analytics, and artificial intelligence capabilities.
- **AWS Amplify Hosting** for hosting fast, secure, and *reliable static and server-side rendered apps that scale with your business.*
- **AWS Device Farm** to improve the **quality of your web and mobile applications** by testing across desktop browsers and real mobile devices hosted on AWS.
- **Amazon Chime SDK** to add **audio, video, and desktop sharing capabilities to your applications.**

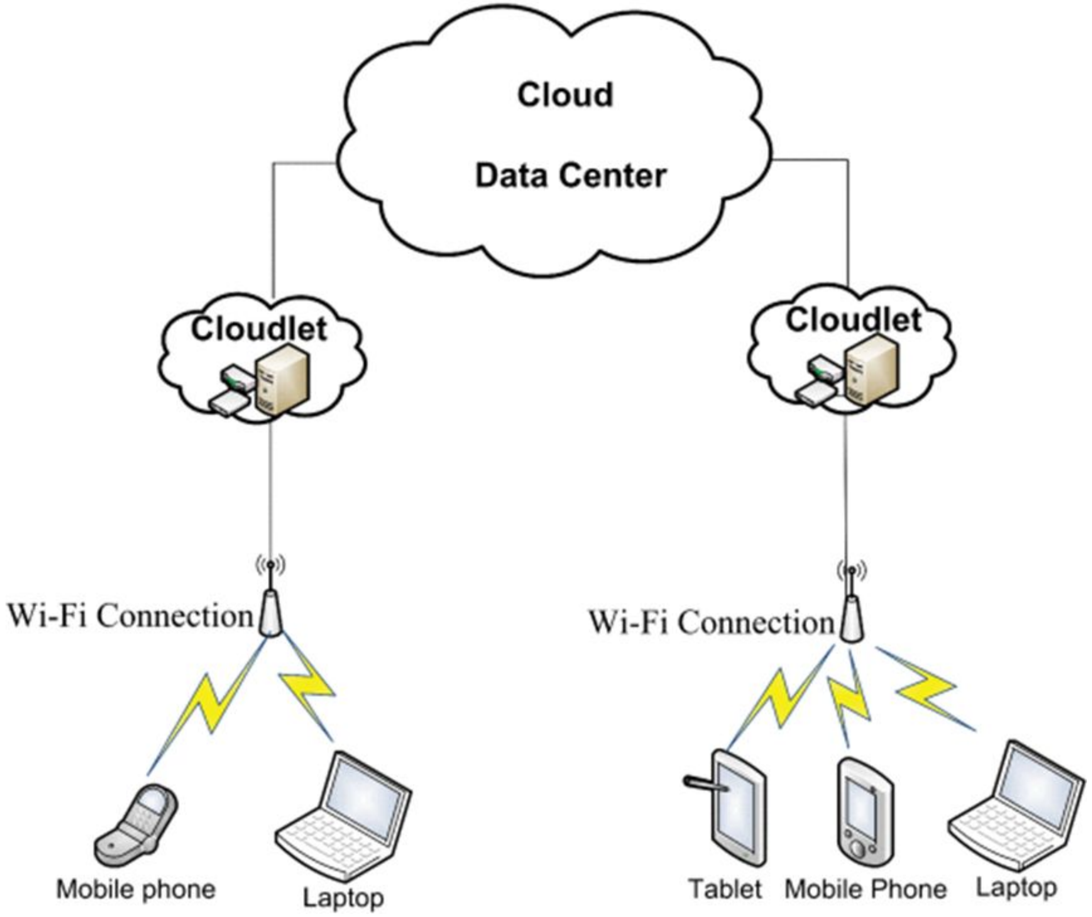
Sample Diagram for AWS amplify

CLOUDLET

- A cloudlet is an emerging computing paradigm that is designed to meet the requirements and expectations of the **Internet of things (IoT) and tackle the conventional limitations of a cloud (e.g., high latency).**
- The idea is to bring computing resources (i.e., **storage and processing) to the edge of a network.**
- Moreover, a cloudlet computation offloading application for ***augmenting resource-constrained IoT devices, handling compute-intensive tasks, and minimizing the energy consumption of related devices is explored***
- Cloudlet is a platform that uses **special containers to run software applications**

CLOUDLET Architecture

- A cloudlet is a small-scale, decentralized data center located at the edge of the network, designed to provide **cloud-computing services closer to end users.**
- The primary purpose of a cloudlet is to bring the **computational power of the cloud closer to mobile devices or other edge devices,** reducing latency and improving the performance of applications that require real-time processing

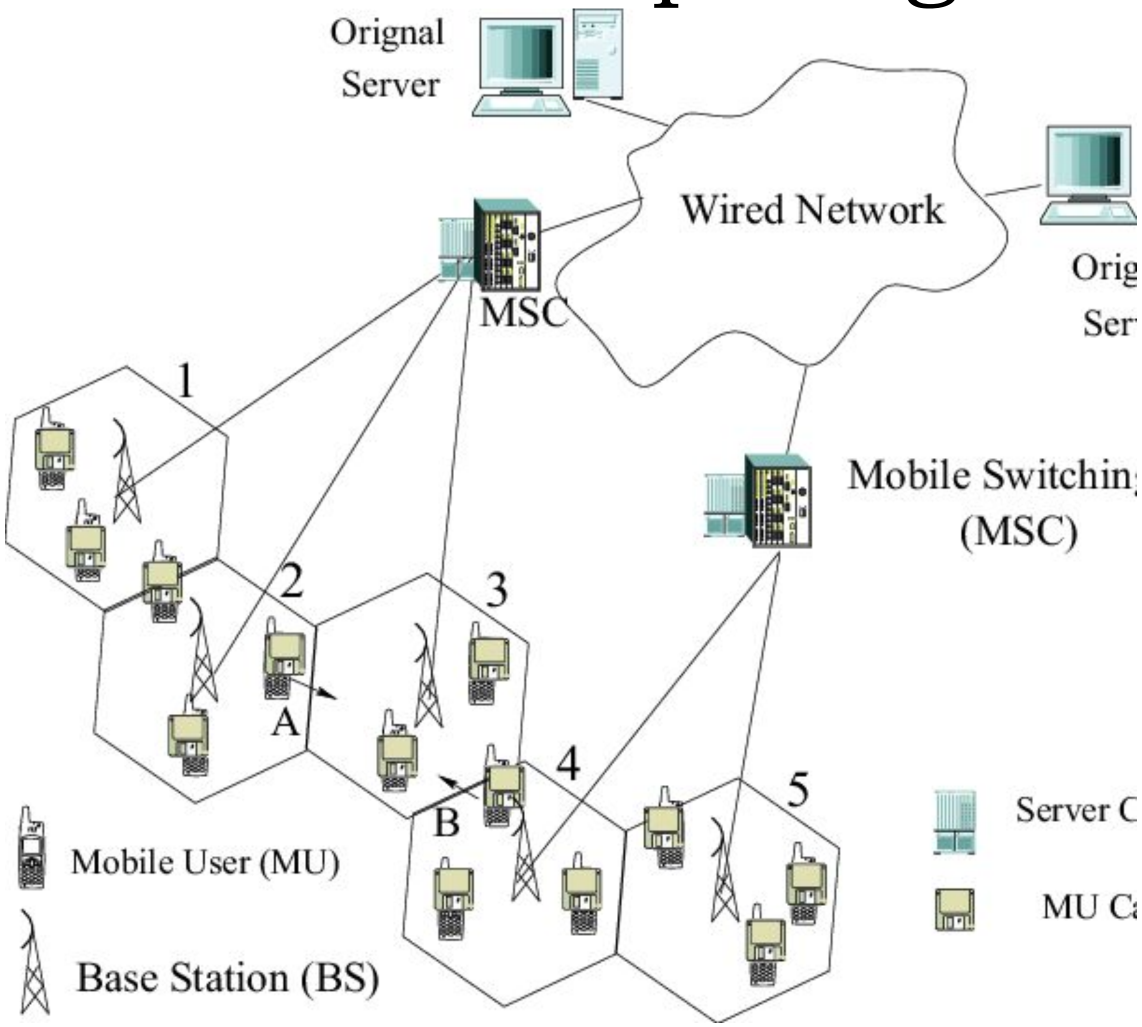


Features of **CLOUDLET**

- **Strong isolation** between untrusted user-level computations
- Mechanisms for **authentication, access control, and metering**
- **Dynamic resource allocation** for user-level computations
- The ability to support **a very wide range of user-level** computations, with minimal restrictions on their process structure, programming languages or operating systems.

Cloudlet Mesh for Mobile Cloud Computing

- Mobile device users may move across a mobile cellular network or another wireless network to access remote clouds.
- Due to limited resources available on a smartphone or tablet computer, the remote cloud access could be constrained by **limited battery life, low CPU power, and small storage capacity**.
- Cloudlet, which offers a resource-rich portal for upgrading mobile devices with **cognitive abilities to access distance clouds**

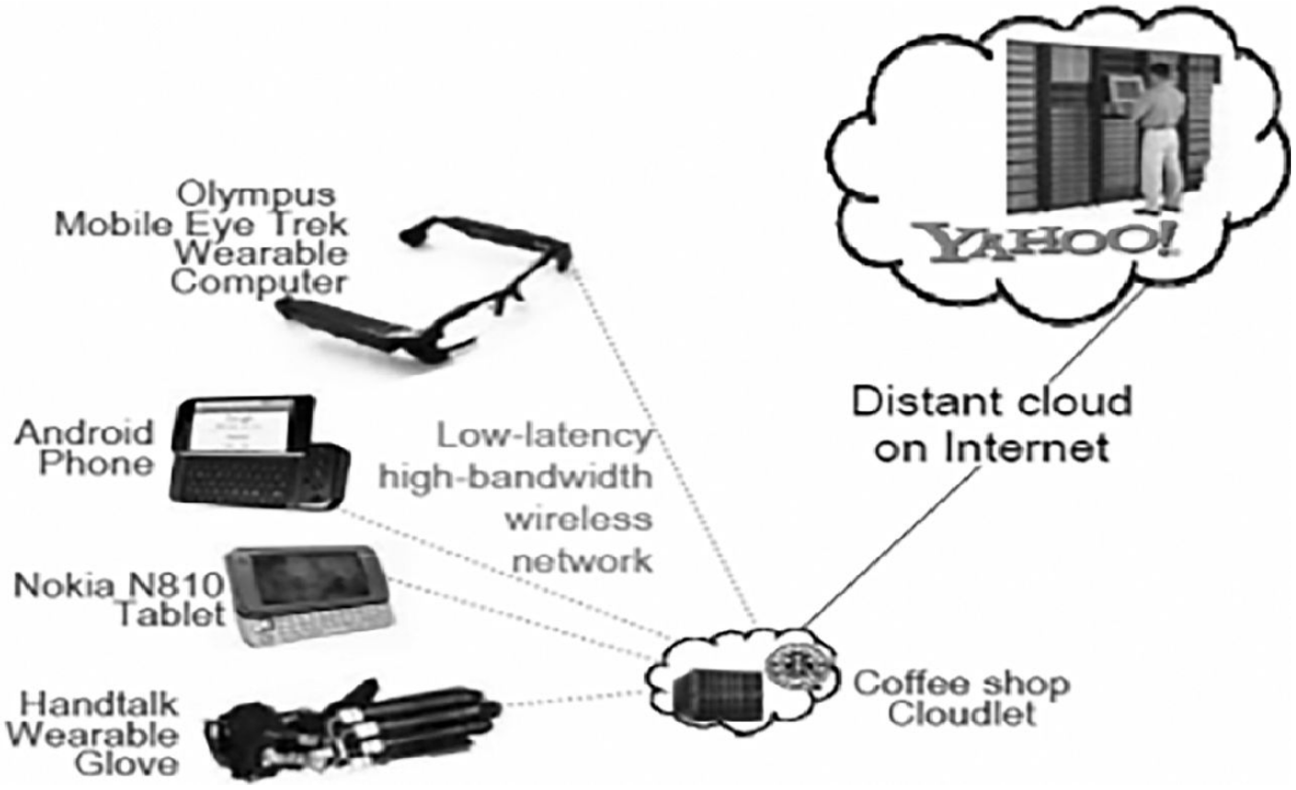


- **Trustworthy fashion**, using virtual machines (VMs) to explore location-aware cloud applications.
- Cloudlet makes it possible for mobile devices to offload a **heavy database search or machine learning jobs**
- **T**o nearby or remote clouds for **processing or analysis and return with short answers or predictions quickly**.
- A **Cloudlet Mesh** for **Mobile Cloud Computing (MCC)** refers to a **decentralized framework of interconnected cloudlets**, which are small-scale data centers deployed at the edge of the network, closer to mobile users.
- In mobile cloud computing, resource-intensive tasks like data **processing, storage, and computational tasks are offloaded from mobile devices to the cloud, reducing battery usage and enhancing performance.**

Scenario based question for cloudlet

- **Scenario:** Abundant data collected by **IoT sensing devices could be passed through a smartphone** to a remote cloud for processing or **machine learning systems**.
- The cloudlet can help to link the users to remote clouds in performing **DATA MINING AND MACHINE LEARNING OPERATIONS.**
- This cloudlet portal is designed to be **TRUSTABLE AND USES VMS** to explore location aware cloud applications.

Cloudlet Gateway for Mobile Devices to Access Remote Clouds



Fast VM Synthesis in Cloudlets

- **VM synthesis** involves the **creation and booting of a virtual machine from a VM image**.
- The goal of fast VM synthesis is to **minimize the time required to instantiate a new VM** and make it ready to perform tasks.

- Techniques used : **Overlay-based VM Images**,

  Delta-based Synthesis

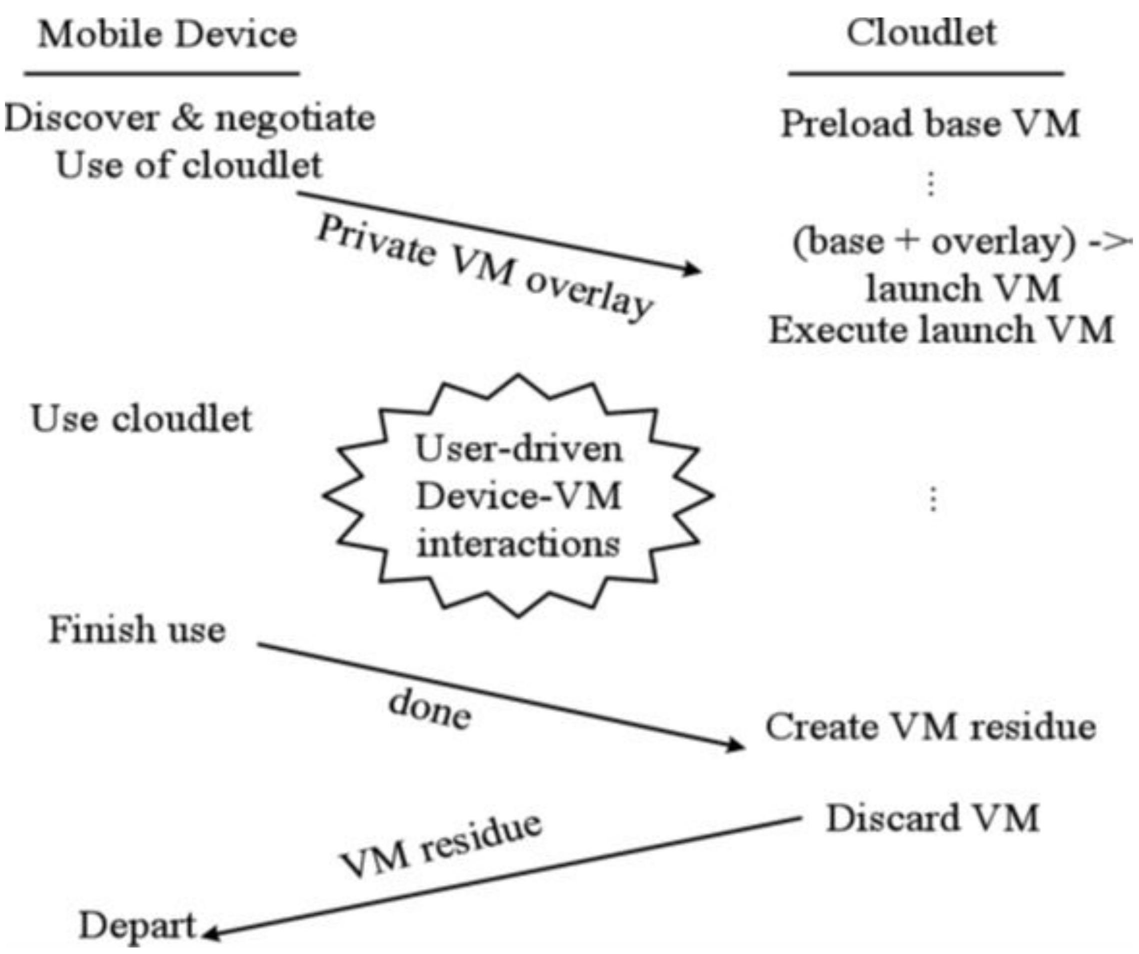
  Pre-fetching and Caching

  Container-based

Fast VM Synthesis in Cloudlets

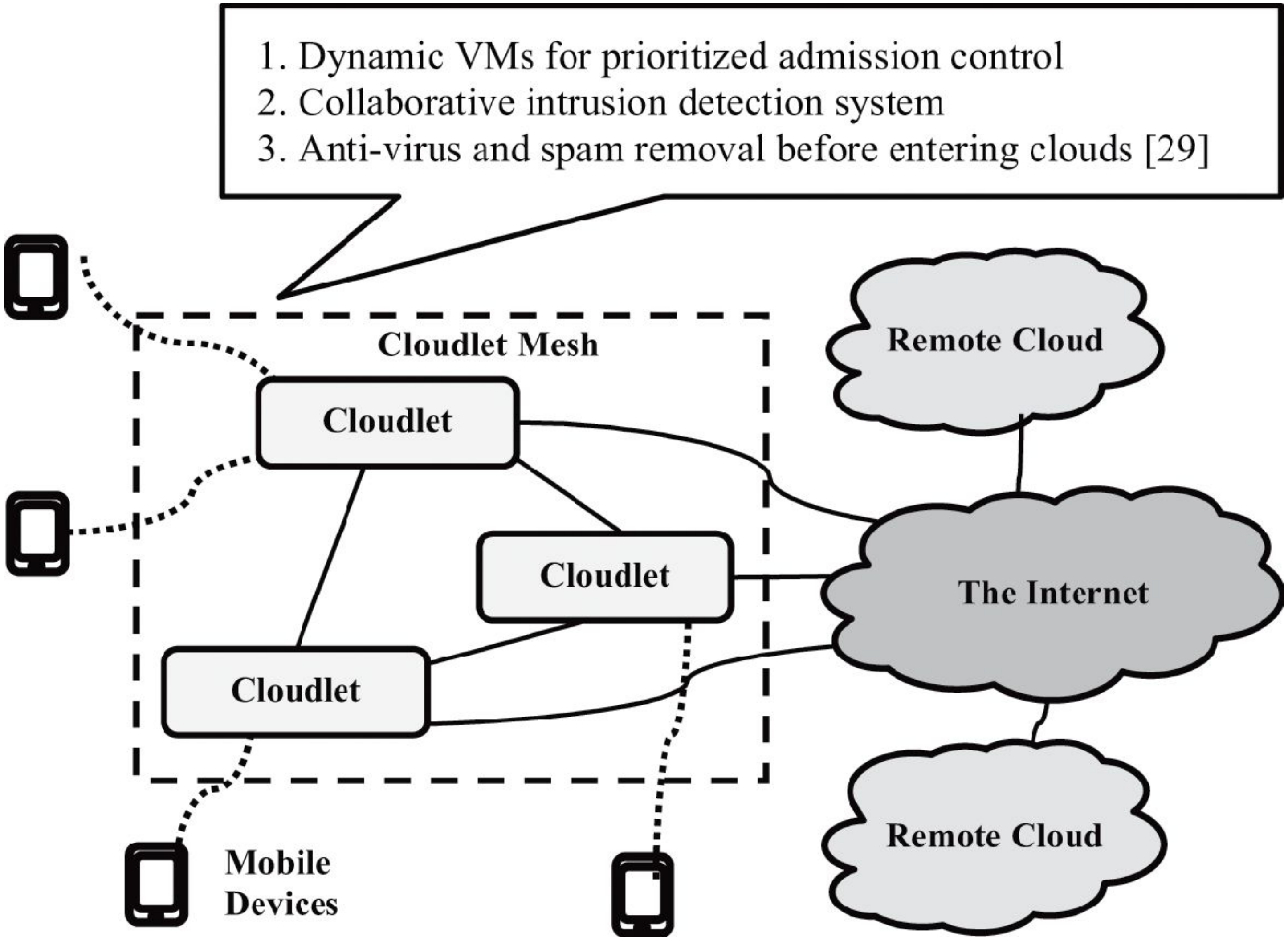
- Overlay virtualization is a method for **creating traffic isolation** within a **MULTITENANCY INFRASTRUCTURE**.
- Using a form of **tunneling between isolated network segments**, it allows for **scalability and ease of use** while providing for separation between the virtual network and the underlying physical environment.
- A small VM overlay is delivered by the mobile device to a cloudlet that already possesses a base VM.
- The **VM overlay** combined with the base VM creates a special execution environment for the mobile device to launch its cloud applications through the cloudlet portal.
- **Trust and security** issues are also major factors in cloudlet deployment.

VM overlay provides an additional abstraction layer that allows virtual machines to be managed, deployed, and moved easily between different data centers, cloud platforms, or edge locations without being tied to the underlying hardware infrastructure.



- **DATA PROTECTION** includes file/log access control, data coloring, and copyright compliance.
- **DISASTER RECOVERY** is also needed to secure data from being lost due to hardware/software failures.
- **Cloud security** can be enforced with establishing the root of trust, securing the VM provisioning process, software watermarking, and the use of firewalls and IDSs at host and network levels.
- Recently, **trust overlay networks and reputation systems** were suggested to protect data centers from trusted cloud computing

- All cloudlets are Wi-Fi-enabled.
- Each cloudlet server has an embedded **Wi-Fi access point.**
- Each cloudlet **connects many mobile devices** within the Wi-Fi range.
- The cloudlets are interconnected by **wireless links to form the mesh.**
- All cloudlets operate essentially as gateways at the edge network of the Internet


1. Dynamic VMs for prioritized admission control
2. Collaborative intrusion detection system
3. Anti-virus and spam removal before entering clouds [29]
Cloudlet Mesh
Cloudlet
Cloudlet
Cloudlet
Remote Cloud
The Internet
Remote Cloud
Mobile
Devices

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

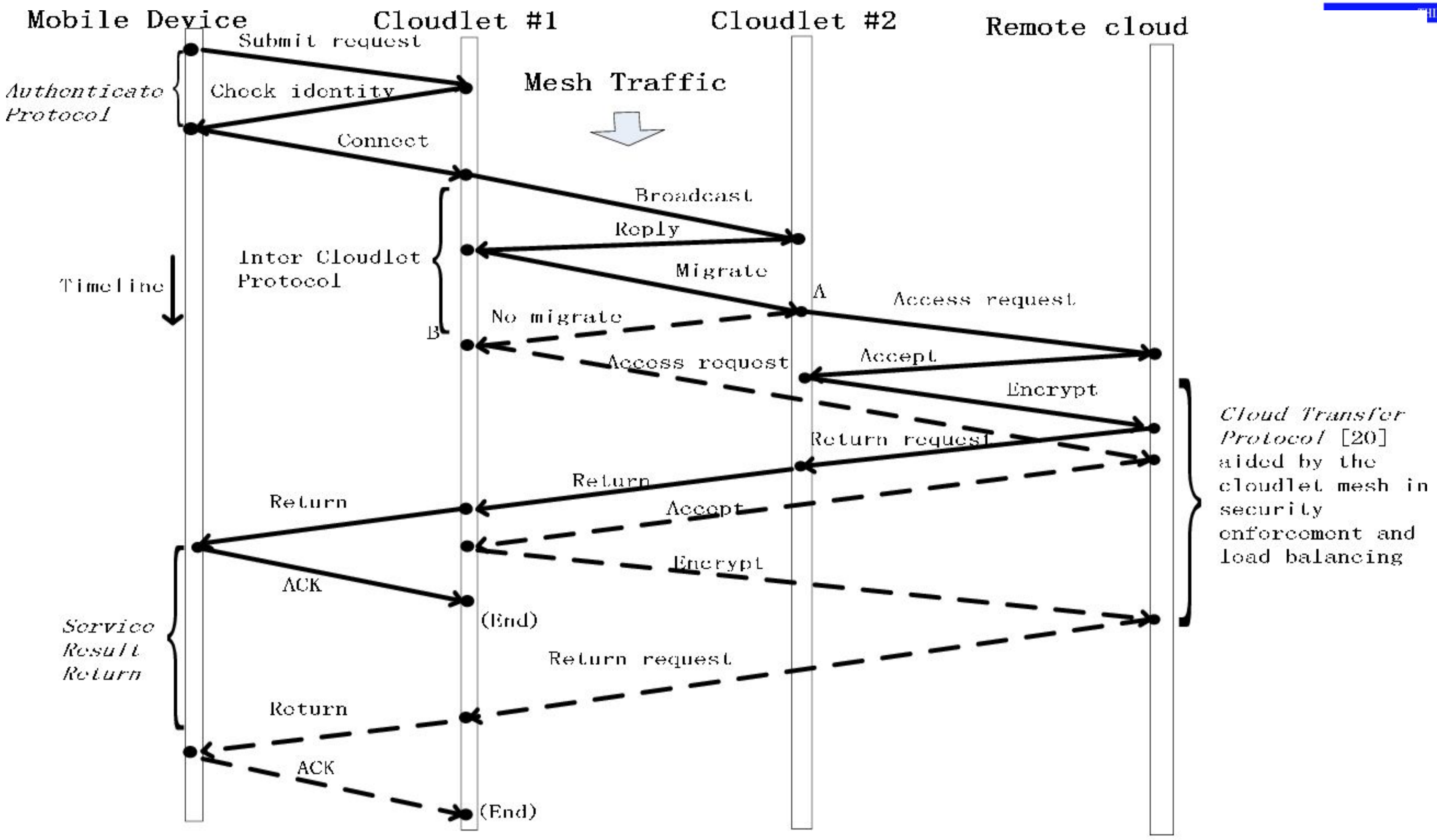


Figure 3. Security protocols applied for interactions among mobile devices, cloudlets and remote clouds

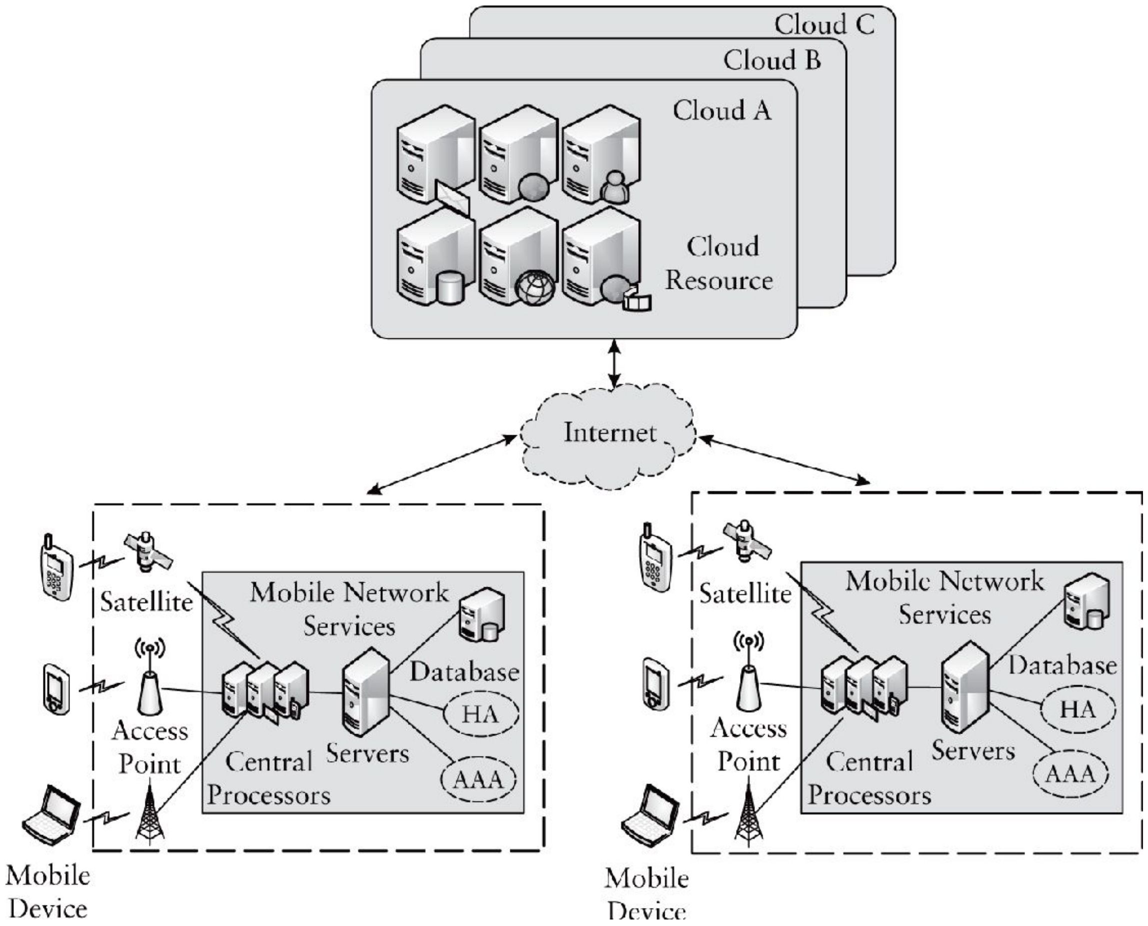
Threats/Defense	Mobile Device	Cloudlet Mesh	Remote Clouds
Encryption for Data Protection	Energy cost for encryption is high on mobile devices	Encryption to secure the access of a remote cloud	Encryption fully supported to protect user data lost
Virus, Worms, or Malware Attacks	Privacy and energy cost for detecting malware is high	Protect mobile device by verifying files and content	Perform analytics on the cloud to detect new types of malware
Identity Theft and Authentication	User authentication before offloading to clouds	Need to authenticate all three parties involved	Authentication as a service (AaaS) is needed
Cloud Offloading and File Transfer	Offloading tasks in security- enforced cloudlet mesh	Data caching at cloudlet to improve performance	High latency to offload may create a QoS problem
Data Integrity and Storage Protection	May use secure storage outsourcing protocols	Data stored by the cloudlet is vulnerable to attacks	Clouds may compromise user data through phishing attacks
URL and IP and Spam Filtering	Checking blacklist of IP addresses and URLs	Alert mobile devices with intrusive attacks on clouds	Performs predictive analytics and provides database updates

Scenario based for CLOUDLET

- A mobile user playing an augmented reality game may need a **cloudlet to quickly instantiate a VM to perform complex computations** (e.g., rendering 3D graphics or processing sensor data).

Mobile Device Request → **VM Synthesis** → **Computation** → **Result communication** → **VM Termination**

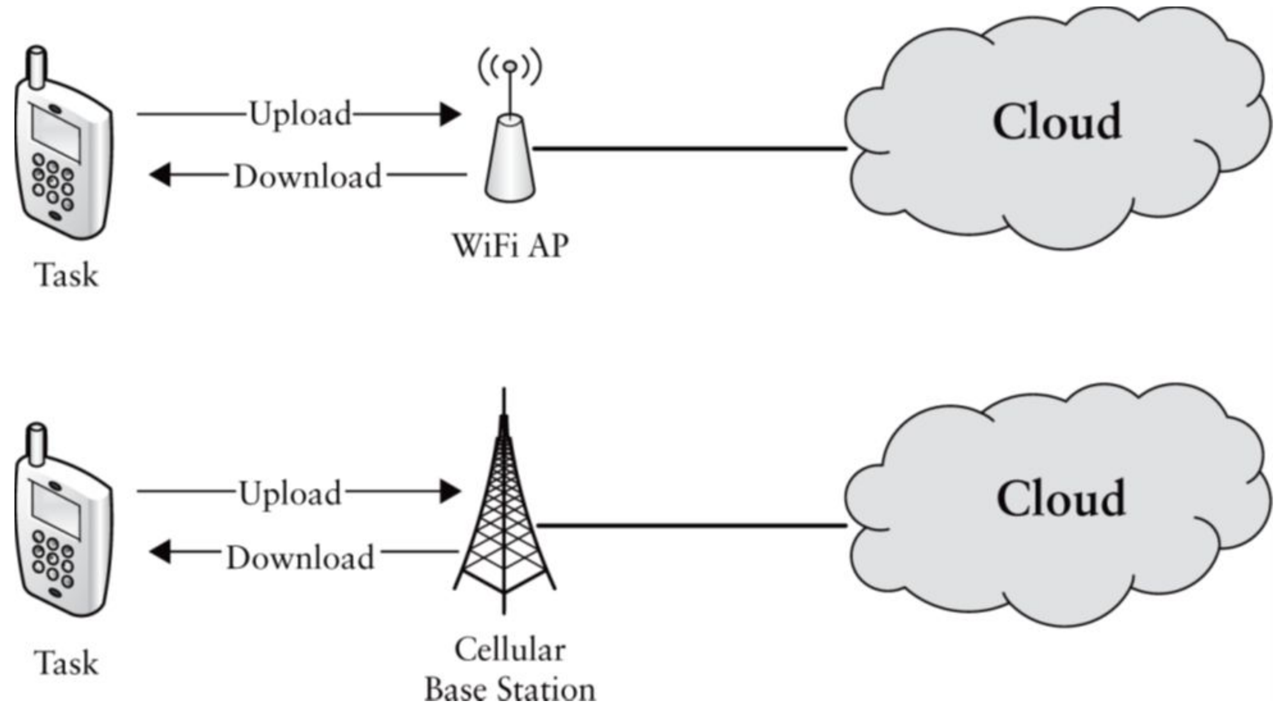
Mobile Clouds and Colocation Clouds

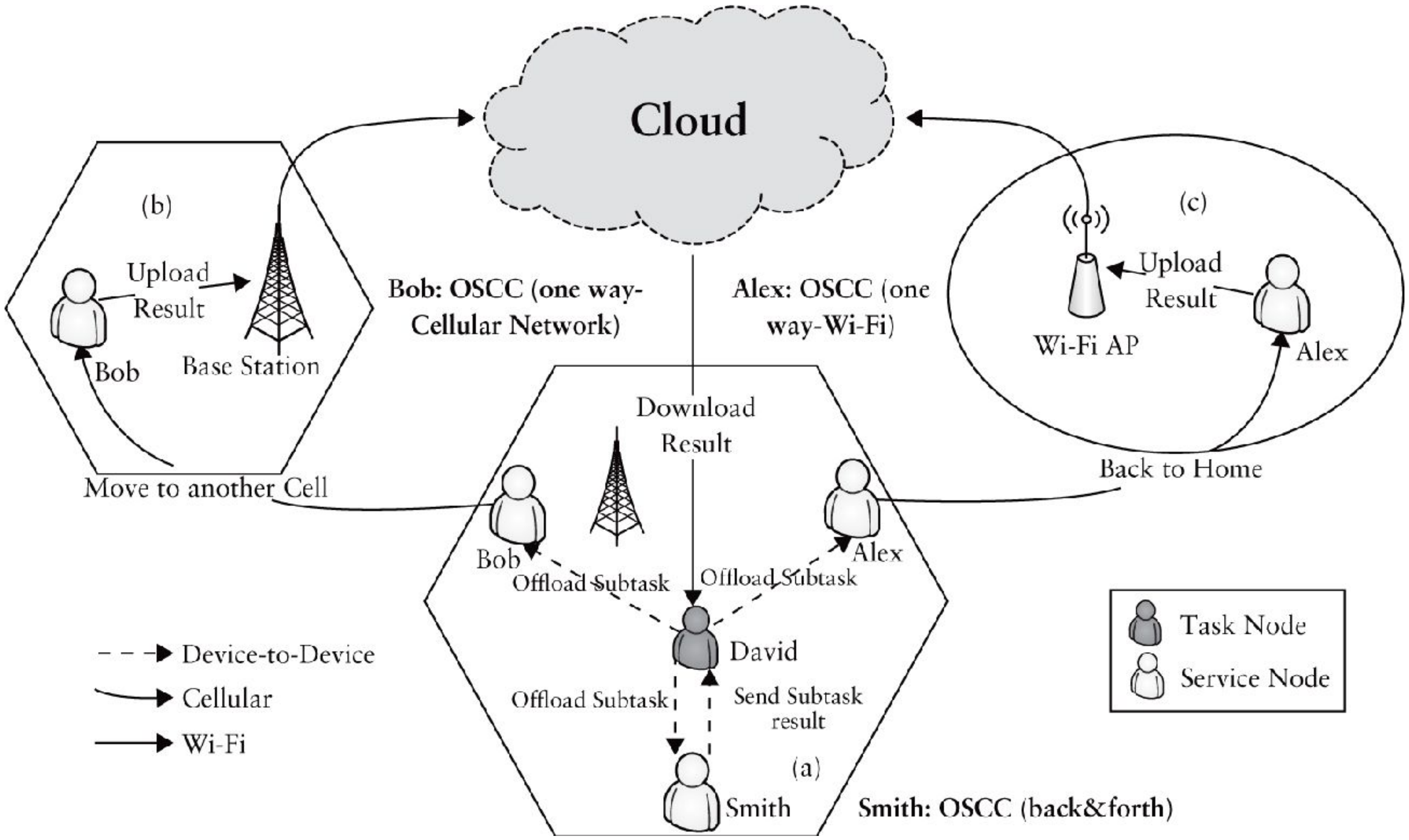


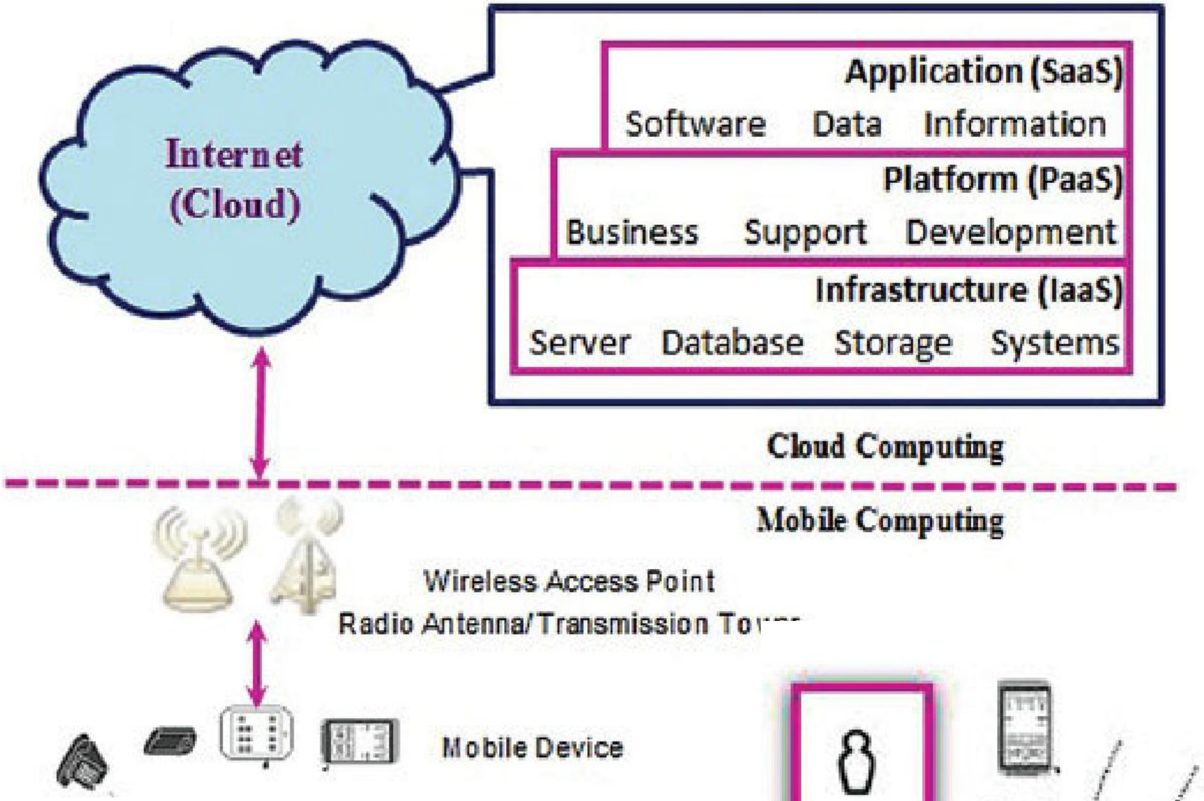
- Mobile cloud computing (MCC)
  - A mobile user basically has a new cloud option to execute the application.
  - The user attempts to offload the computation through wi-fi, cellular network, or Satellite to the distant clouds.
  - The terminal devices at the user’s end have limited resources, i.E., Hardware, energy, or bandwidth.
  - The cell phone itself is **infeasible to finish some compute-intensive tasks**.
  - Instead, the data related to the computation task is offloaded to the remote cloud.

- Special cloudlets were introduced to serve as wireless gateways between mobile users and the internet.
- These cloudlets can be used to **offload computations or web services to remote clouds safely.**
- With the growing popularity of mobile devices, a new type of **peer-to peer (P2P) mode** for mobile cloud computing has been introduced.
- By applying short-range wireless networks, one can easily connect to nearby mobile devices using the cloudlets.
- Task scheduling scheme **OVER COLOCATED CLOUDS (OSCC)**

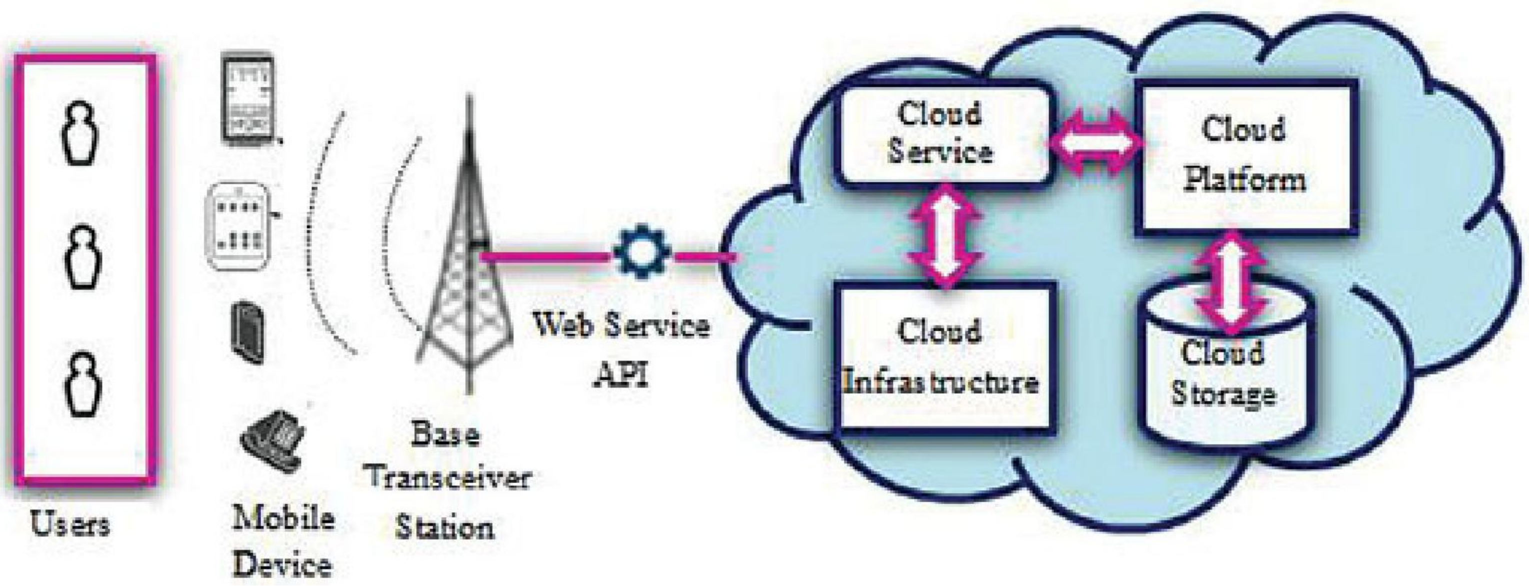
- A basic feature of OSCC is that the contact between ***the task node and the service node*** can be either short or long instead of limiting users' **mobility to guarantee the contact time for task completion in conventional cloudlet based service mode**
- **Based on the location of the service node upon the sub-task completion, there are three situations:**
  - Move close to the task node again within **D2D communication range**
  - Cannot to connect with the task node directly by D2D communications, but wifi can still work,
  - In no way to connect the task node by neither D2D links nor wifi, but the cellular network can still work


Upload
Download
Task
WiFi AP
Cloud
Upload
Download
Task
Cellular
Base Station
Cloud


Cloud
(b)
Upload Result
Bob
Base Station
Bob: OSCC (one way-Cellular Network)
Alex: OSCC (one way-Wi-Fi)
(c)
Upload Result
Wi-Fi AP
Alex
Download Result
Move to another Cell
Back to Home
Bob
Alex
Offload Subtask
Offload Subtask
David
Offload Subtask
Send Subtask result
(a)
Smith
Smith: OSCC (back&forth)
Device-to-Device
Cellular
Wi-Fi
Task Node
Service Node



USE IT FOR SCENARIO BASED QUESTION



Applications And Examples of Mobile Cloud Computing

- **Email:** This is a prominent example that lots of people use. Gmail, Outlook, and Yahoo Mail are numerous examples of mobile email.
- When you check your emails through your smartphone, you're using mobile cloud computing technology.

- **Social Media**: It enables quick sharing of real-time data on social media platforms like Twitter, Instagram, and Facebook.
- For example, a video recorded on a mobile device can be saved and shared with another mobile user.

- **Finance and Commerce:** Using your phone or tablet to track your account balance, making a purchase on ecommerce platforms such as Amazon, Shopify, etc., is an example of mobile cloud computing, and its scalability makes it ideal for commerce and social media as well.

- **Healthcare:** With cloud computing, accessing patient **records through a mobile device is simple**.
- Mobile healthcare also permits massive amounts of instantaneous data stored in the cloud, accessible via a mobile device.
- It enables convenience by allowing access to patient records when needed.

For scenario-based question (EX)

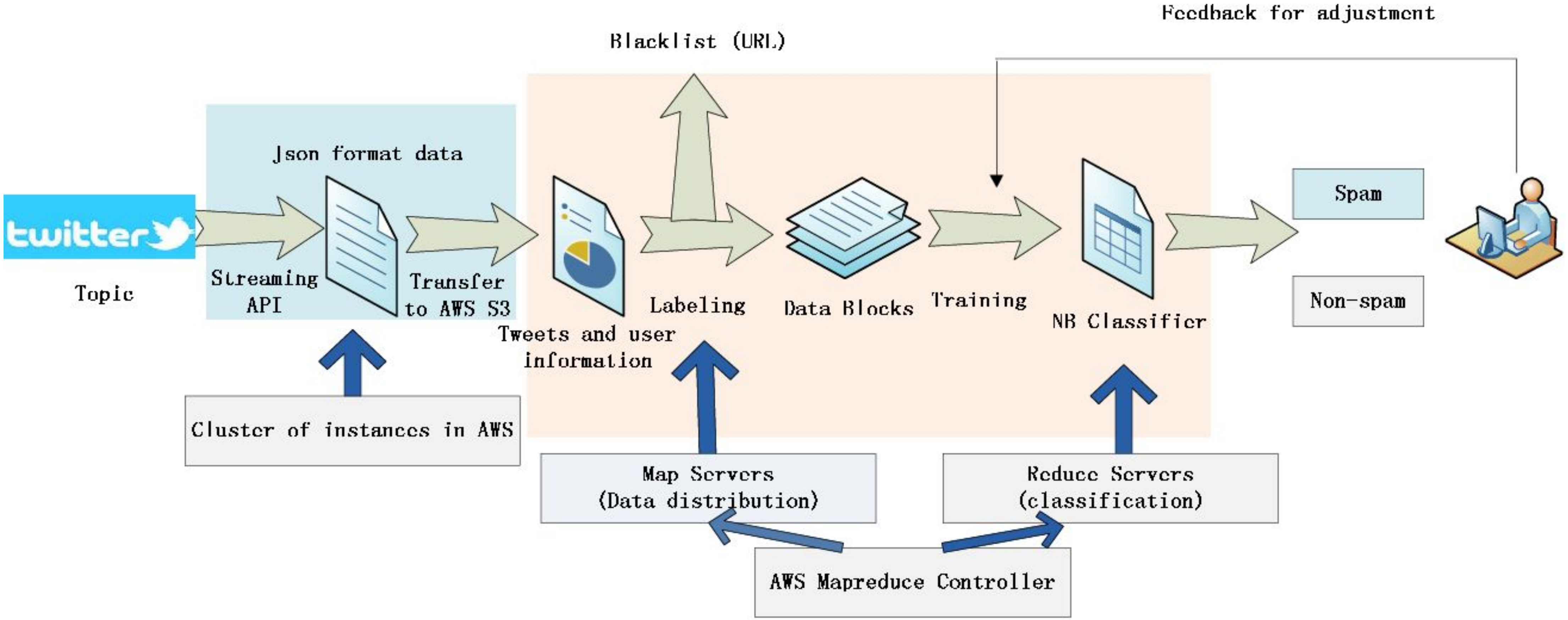


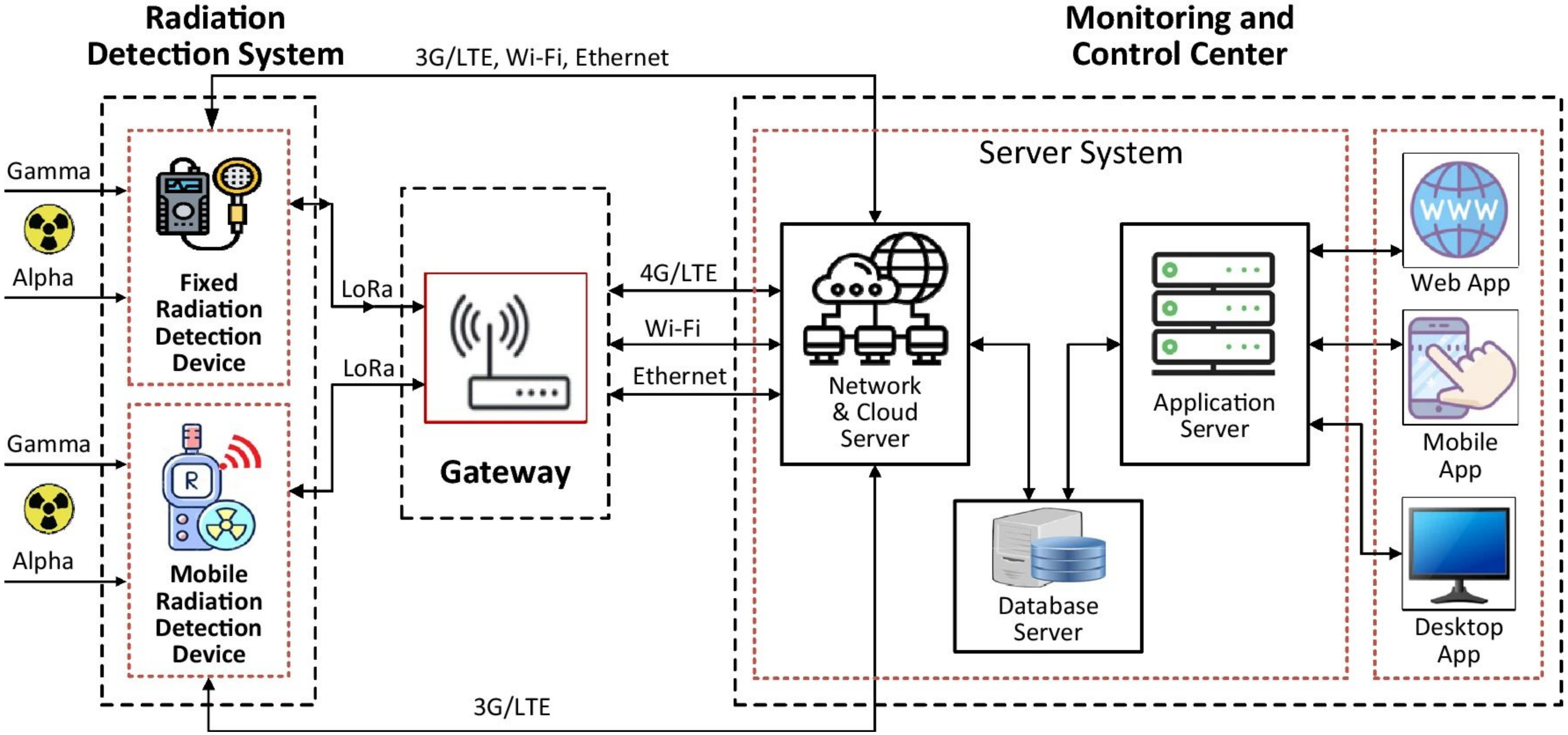
Figure 5 MapReduce spam filtering on remote cloud for securing mobile devices and the cloudlet mesh.

UNIT IV

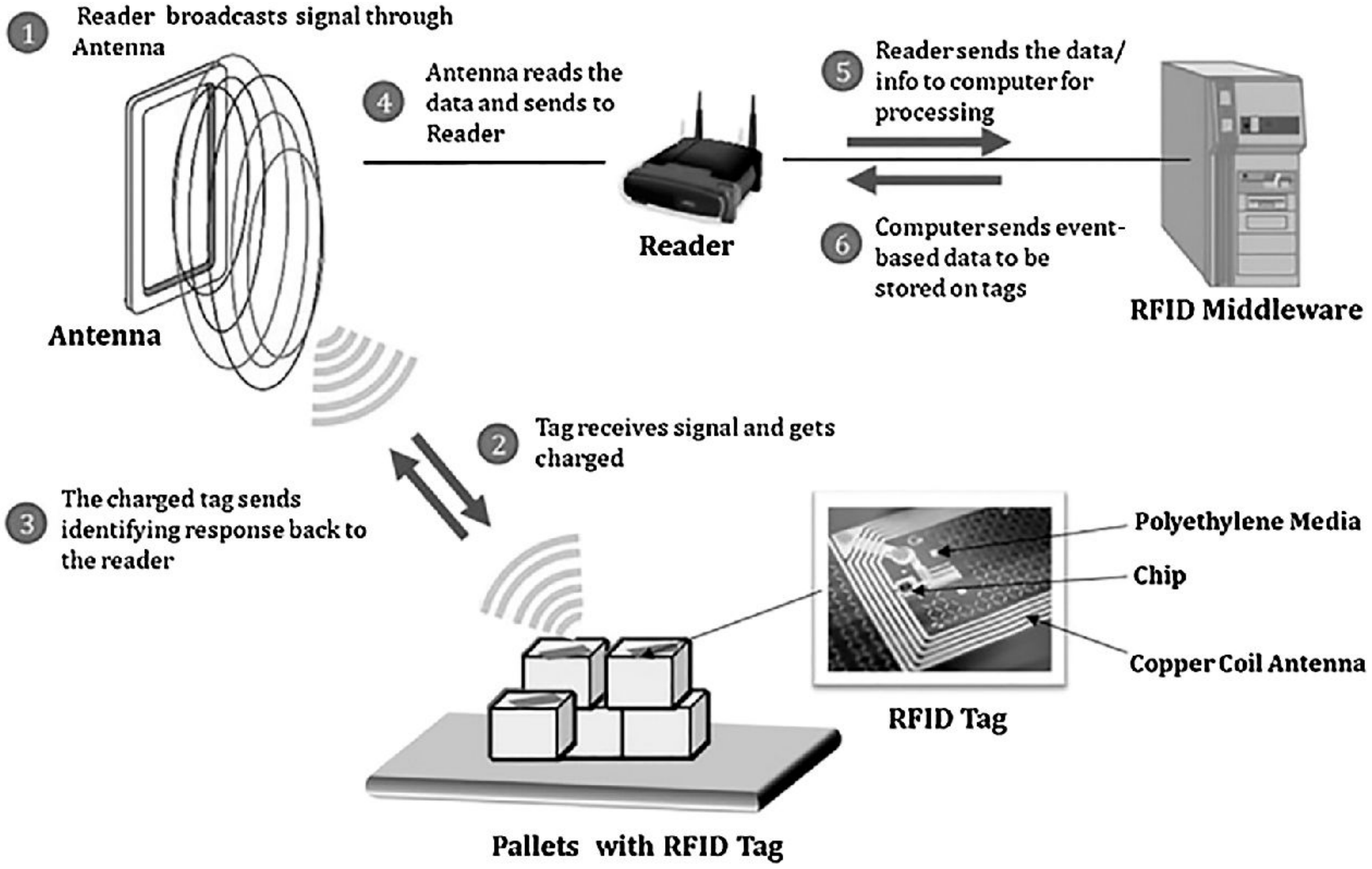
IoT Sensing and Interaction with Clouds

- Integrating the digital and physical worlds is the ultimate goal of IoT.

For Scenario based questions



RFID Technology for Merchandise Tagging or e-Labeling



Local and Global Positioning Systems

- Position techniques are applied to identify **location and measure distance, angle, area, hop count, neighborhood, etc.**
- Basically, **distance-related information** can be obtained by radio signal strength or radio propagation time
- **Angle information** by antenna arrays and area
- Hop count
- Neighborhood information by the fact that radios only exist for nodes in the vicinity.

Local Positioning Technology

- Localization solutions consist of two basic stages:
- Measuring geographic information from the ground truth of network deployment
- Computing node locations according to the measured data.
- Geographic information includes a variety of geometric relationships from coarse-grained neighbor awareness to fine-grained internode ranges (**e.G., Distance or angle**).

Satellite Technology for Global Positioning

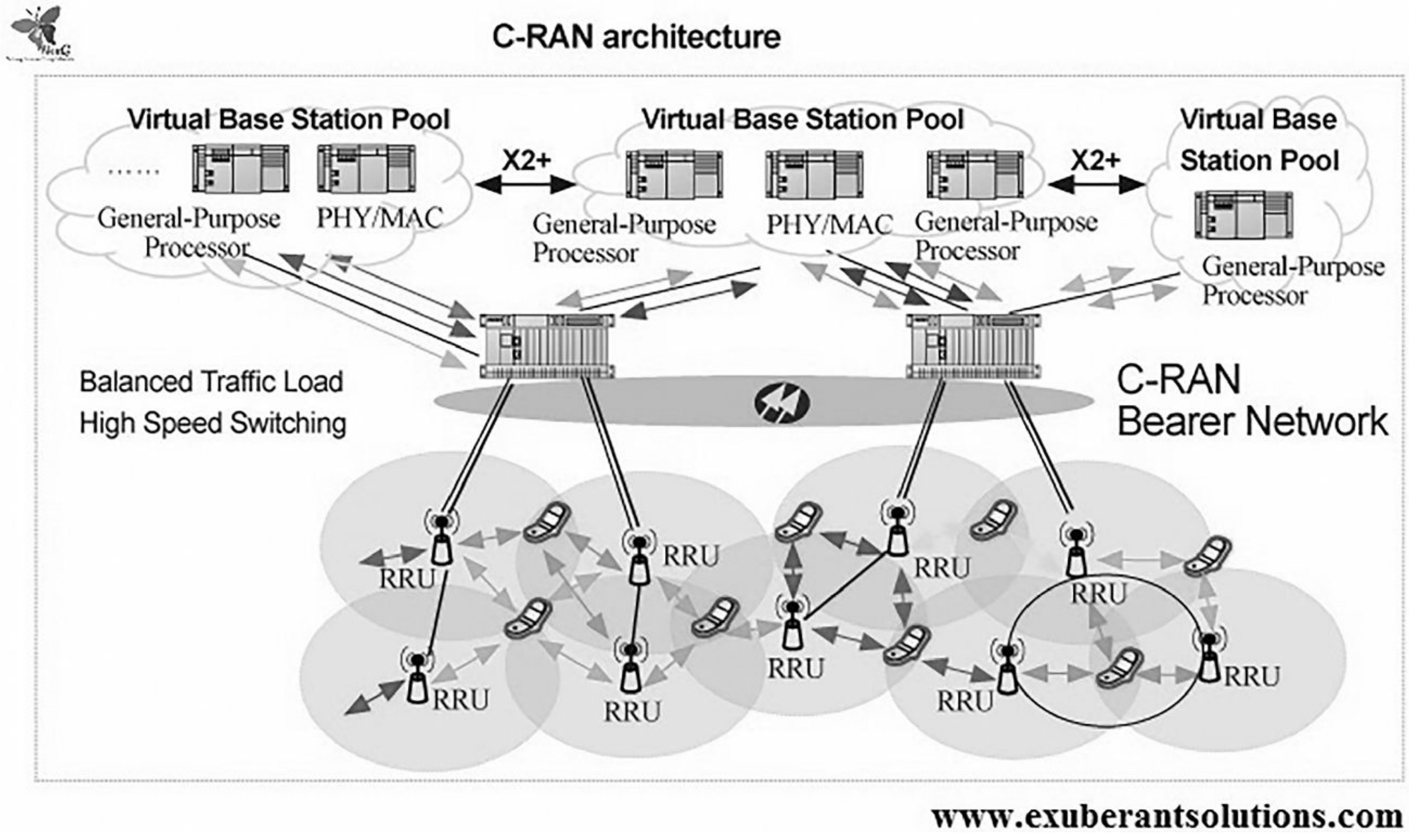
- **Global positioning** is done with multiple satellites deployed in outer space.
- Each satellite continually transmits messages that include the transmission **time and satellite position**.
- A GPS receiver calculates its **position by precisely timing the signals sent by satellites.**
- The receiver uses the messages it receives to determine the transit time and computes the distance d to each satellite using light speed.

Four Global Positioning Systems Deployed

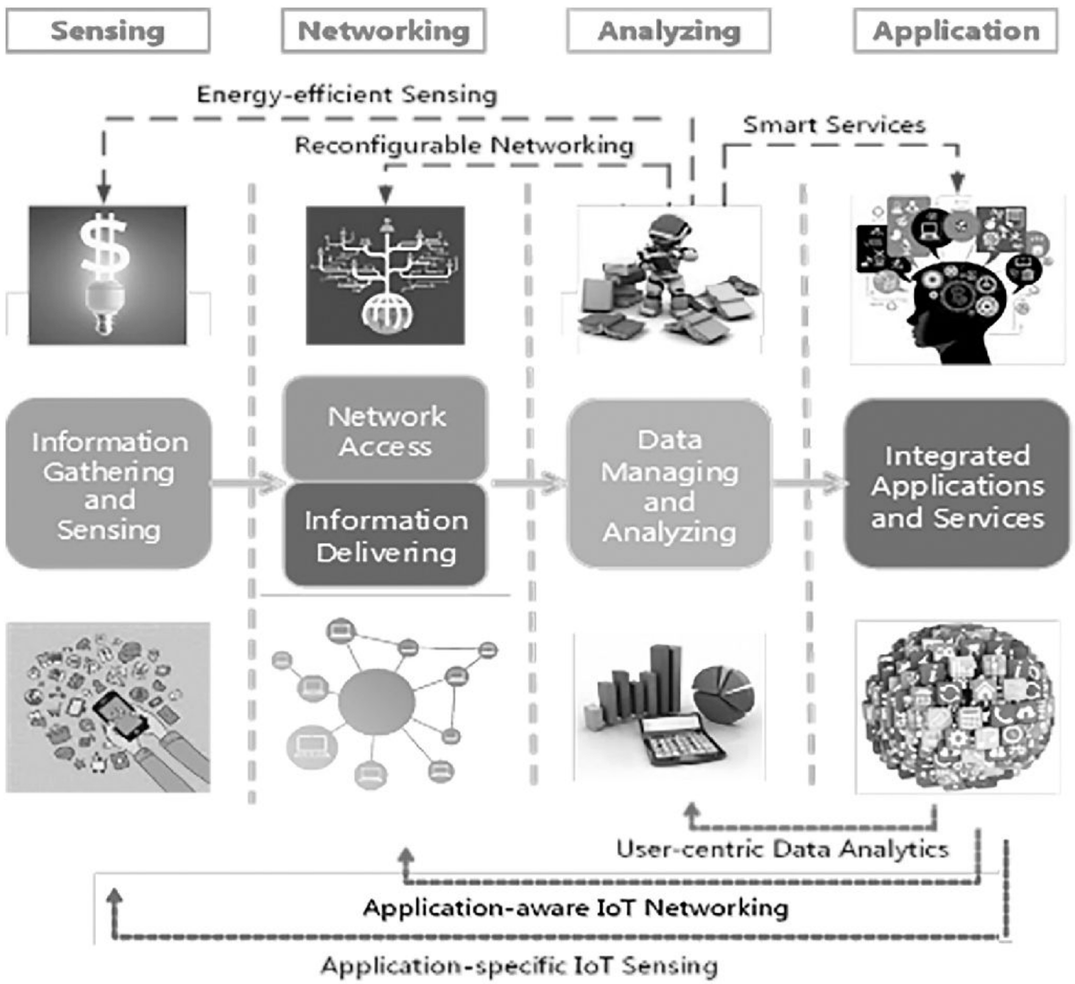
Features	GPS	GLONASS	Beidou	Galileo
Owner	United States	Russia	China	European Union
Coding	CDMA	FDMA/CDMA	CDMA	CDMA
Orbital Height	20,180 km (12,540 mi)	19,130 km (11,890 mi)	21,150 km (13,140 mi)	23,220 km (14,430 mi)
Period	11.97 h (11 h 58 min)	11.26 h (11 h 16 min)	12.63 h (12 h 38 min)	14.08 h (14 h 5 min)
Number of Satellites	At least 24	31 (24 operational)	5 GEO, 30 MEO satellites	22 operational satellites supported

- The bulky antenna towers used in conventional-based stations are replaced by a large number of small **remote radio units (RRU)** that operate with little power (even solar energy can do the job) and get easily distributed with high density in populated user areas.
- The control and processing in physical based stations are replaced by using **virtual base station (VBS)** pools housed in a hierarchy of **cloud-based switching centers**.

Cloud-Based RAN for Building Mobile Networks



IoT Interaction Frameworks with Clouds and Devices



Smart Interactions among IoT Applications and Sensing, Networking, and Analytics

Wireless sensor network (WSN)	• WSN consists of spatially **distributed autonomous sensors to monitor physical or environmental conditions** and to cooperatively pass their data through the network to a main location. • WSNs emphasizing information perception through all kinds of sensor nodes are the basis of IoT.
Machine to machine (M2M) communication	• Typically, M2M refers to **data communications with or without limited human intervention** among various terminal devices such as computers, embedded processors, smart sensors/actuators, and mobile devices. • The rationale behind m2m communication is based on three observations: • A **networked machine is more valuable** than an isolated one • When multiple machines are interconnected, more autonomous applications can be achieved • Smart and ubiquitous services can be enabled by machine-type devices intelligently communicating with other devices at any time and anywhere.

Body-area network (BAN)	• This sensor network is built with smart clothing using lightweight, small-size, ultra-low-power, and intelligent monitoring wearable sensors. • These devices monitor a person's physiological activities such as health status and motion pattern, which can be routed to distant doctors for help in telemedicine applications. • Smart cloth with **BAN is becoming a growing industry**.
Cyber-physical system (CPS)	A system of sensor networks, GPS devices, and computer systems that interact with humans or robots in the control loop.

For Scenario based question

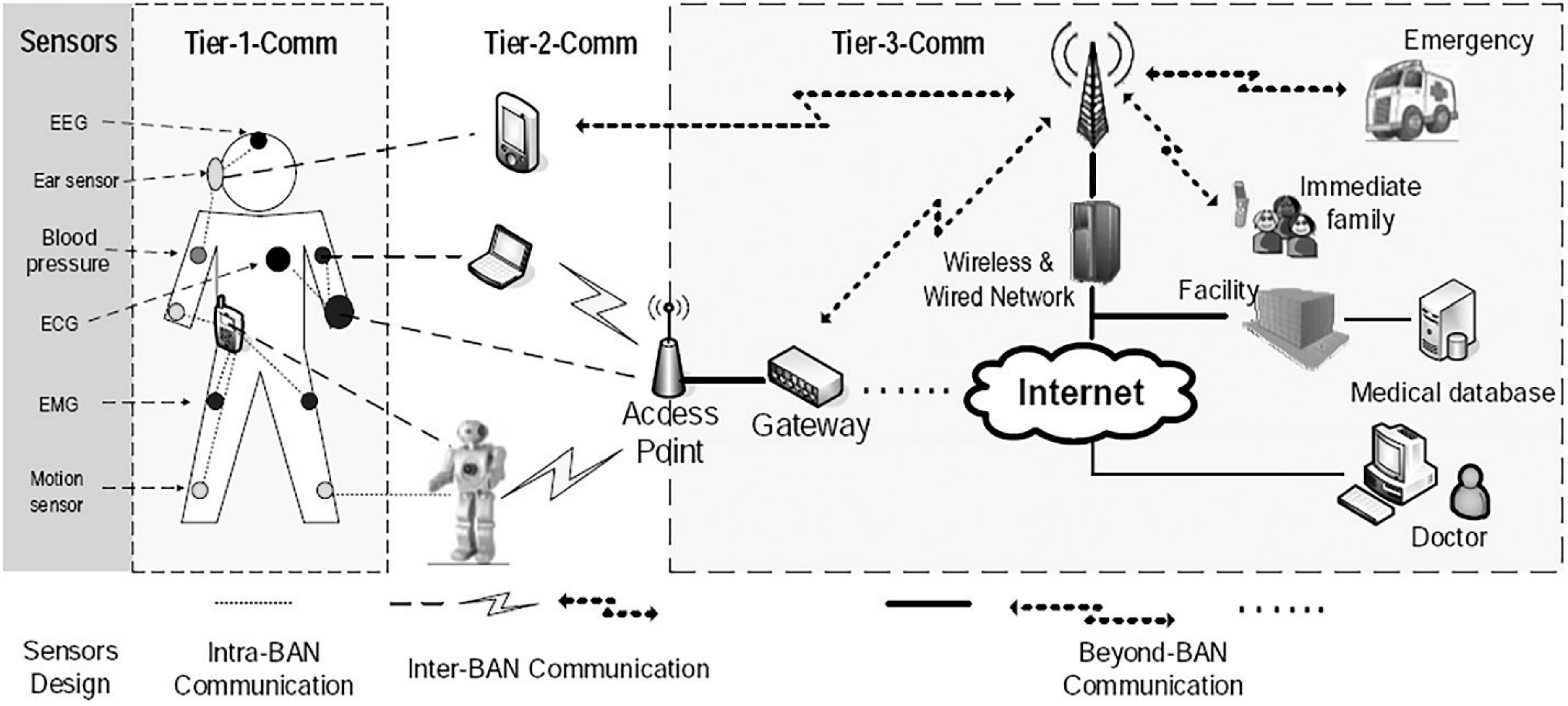


Figure 4.9 **A three-tier architecture based on a BAN communications system.**

Requirements of four IoT computing and communication frameworks

Framework	WSN	M2M	BAN	CPS
Sensing Requirement	XXXX	XX	XXX	XXX
Networking Demand	XX	XXXX	XX	XXXX
Analyzing Complexity	XX	XX	XXX	XXXX
Application Industrialization	XXXX	XXX	XX	X
Security Demand	X	XX	XXX	XXXX

X’s refer to a **higher demand** for that particular feature under the column framework

Cloud Computing in Social Media Applications

- Social media is a major source of **big data aggregation in our daily activities.**
- **Assess data analytics** technologies applied in the social media industry and its impact in all walks of life.

Social Media Big-Data Industrial Applications

- Marketing profits from *microblogs and video streaming, consumer services prefers using forums and mobile systems, sales enjoys products/service reviews, and human resources* prefers to leverage business networks.
- Most organizations apply enterprise social networks.
- Mobile social media users make use of the location- and/or time-sensitive features of the big data set collected

Social media corporate functions weighted by social-economic impact

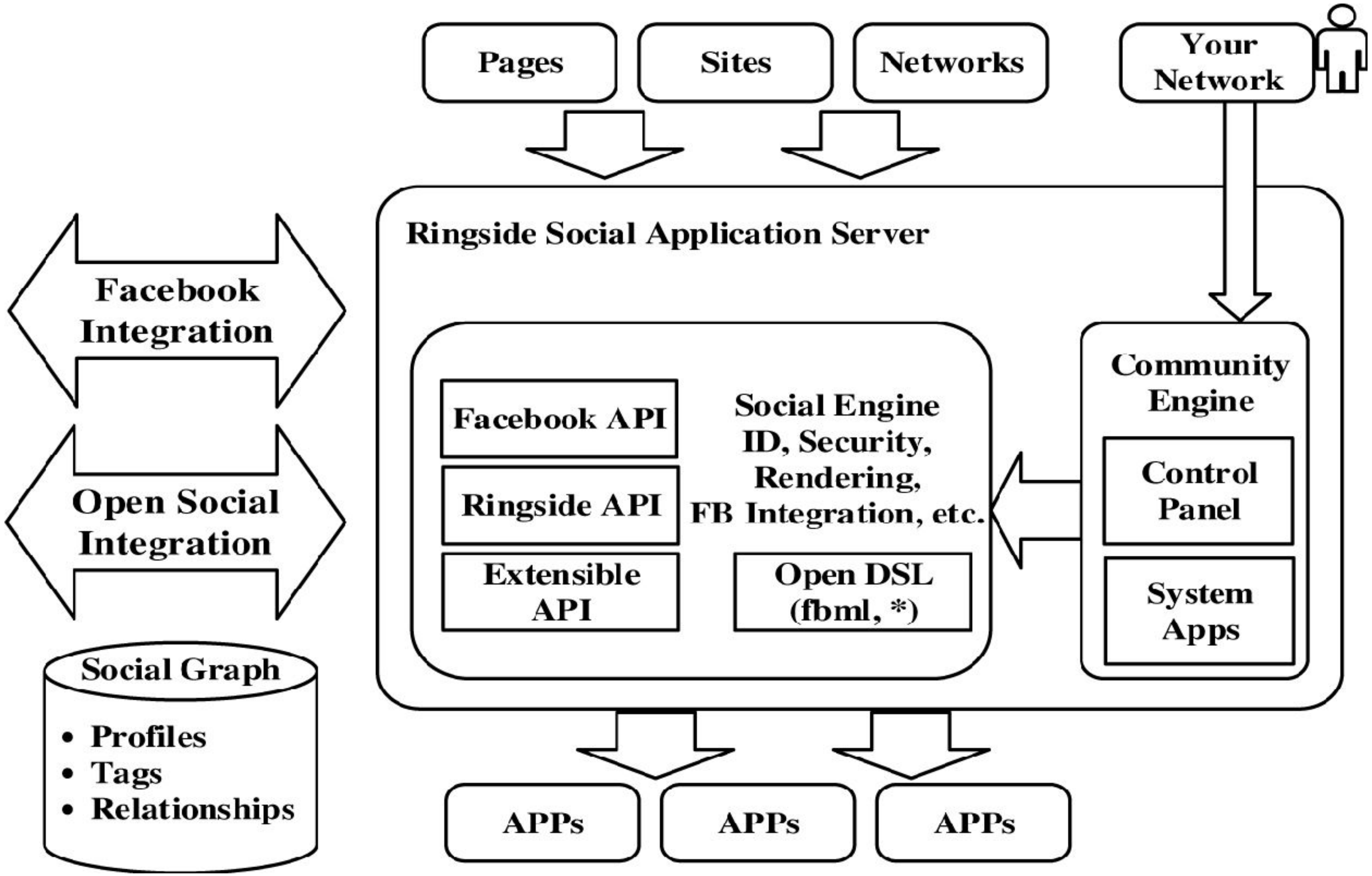
Corporate Function	Res. and Develop.	Marketing	Customer Service	Sales	Human Resources	Organization
Blogs	Low	Medium	Low			
Business Networks					Very high	Low
Collaborative Projects	Very high					Very high
Enterprise Networks	High				Medium	
Forums	Medium	Low	Very high		Low	
Microblogs		High				
Photo Sharing		Medium				
Products/Services Review	Low	Medium		Very high		
Social Bookmarking		Medium				
Social Gaming		Medium				
Social Networks	Low	Very high	Medium		Low	Low
Video Sharing		Very high	Low			
Virtual Worlds	Low	High		Low		

Marketing research	• **Online data collections** can escalate rapidly to a large amount. • They must be handled quickly and **continuously in a streaming mode**. • The requirement is to keep all concerned parties or firms well informed with the exact times of **transactions and the comments made during the transactions or social network visit**
Communication in social media exchanges	• Mobile social media communication takes the form of **business to consumer (B2C),** in which a company may establish a connection to a consumer based on its location and provide reviews about user-generated content
Sales promotions and discounts	• Although customers have had to use printed coupons in the past, **mobile social media allows companies to tailor promotions to specific users at specific times**
e-Commerce	• Mobile social media applications such as Amazon.com and Pinterest have started to influence an upward trend in the popularity and accessibility of e-commerce, or online purchases. • Such e-commerce events could be conducted as **B2B (business to business), B2C (business to customer), C2B (customer to business), or C2C (customer to customer) in a peer-to-peer (P2P) fashion**. • **O2O transactions online to offline or offline to online sales or business exchanges**

Social Networks and API for Social Media Applications

Social Network, Year, and Website	Registered Active Users	Major Services Provided
Facebook, *2004,* www.facebook.com	1.65 billion users, 2016	Content sharing, profiling, advertising, events, social comparison, communication, play social games, etc.
Tencent QQ *in China, 1999,* www.tencent.com	853 million users, 2016	An instant messaging service, online games, music, ebQQ, shopping, microblogging, movies, WeChat, QQ Player, etc.
Linkedin, *2002,* www.linkedin.com	364 million users, 2015	Professional services, online recruiting, job listings, group services, skills, publishing, advertising, etc.
Twitter, *2006,* www.twitter.com	320 million users, 2016	Microblogging, news, alerts, short messages, rankings, demographics, revenue sources, photo sharing, etc.

Facebook Platform Architecture



FLOW DIAGRAM for RINGSIDE

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

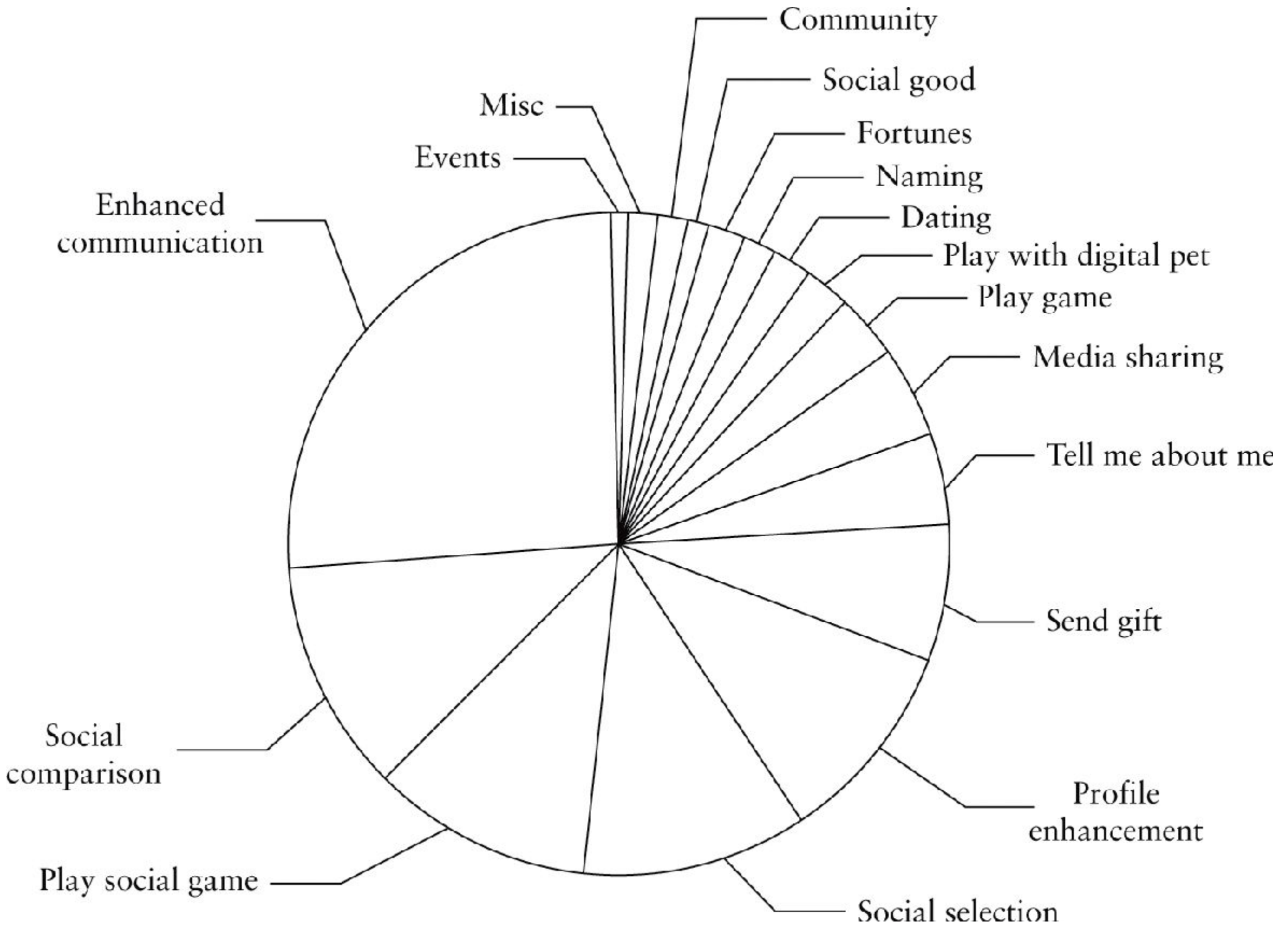
USERS
- Request initiation
- Select Applications

Social network
- Integrate
- Builds Context
- Access Social Graph

APPLICATIONS
- Process by executing thru Social network
- Passing the context
- Providing API for more information

RESPONSE
- Getting the output and passing to the social media application

Facebook platform offering over 2.4 million user applications



Service functionality of the Facebook platform

Function	Short Description
Profile Pages	Profile picture, bio information, friends list, user's activity log, public messages
Graph Traversal	Access through user's friends list on profile page, with access control
Communication	Send and receive messages among friends, instant messaging, and microblogging
Shared Items	Photo album with built-in access control, embedded outside videos on profile page
Access Control	Access control levels: only me, only friends, friends of friends, and everyone
Special APIs	Games, calendars, mobile clients, etc.

Programming Interfaces (APIs) for Social Media Applications

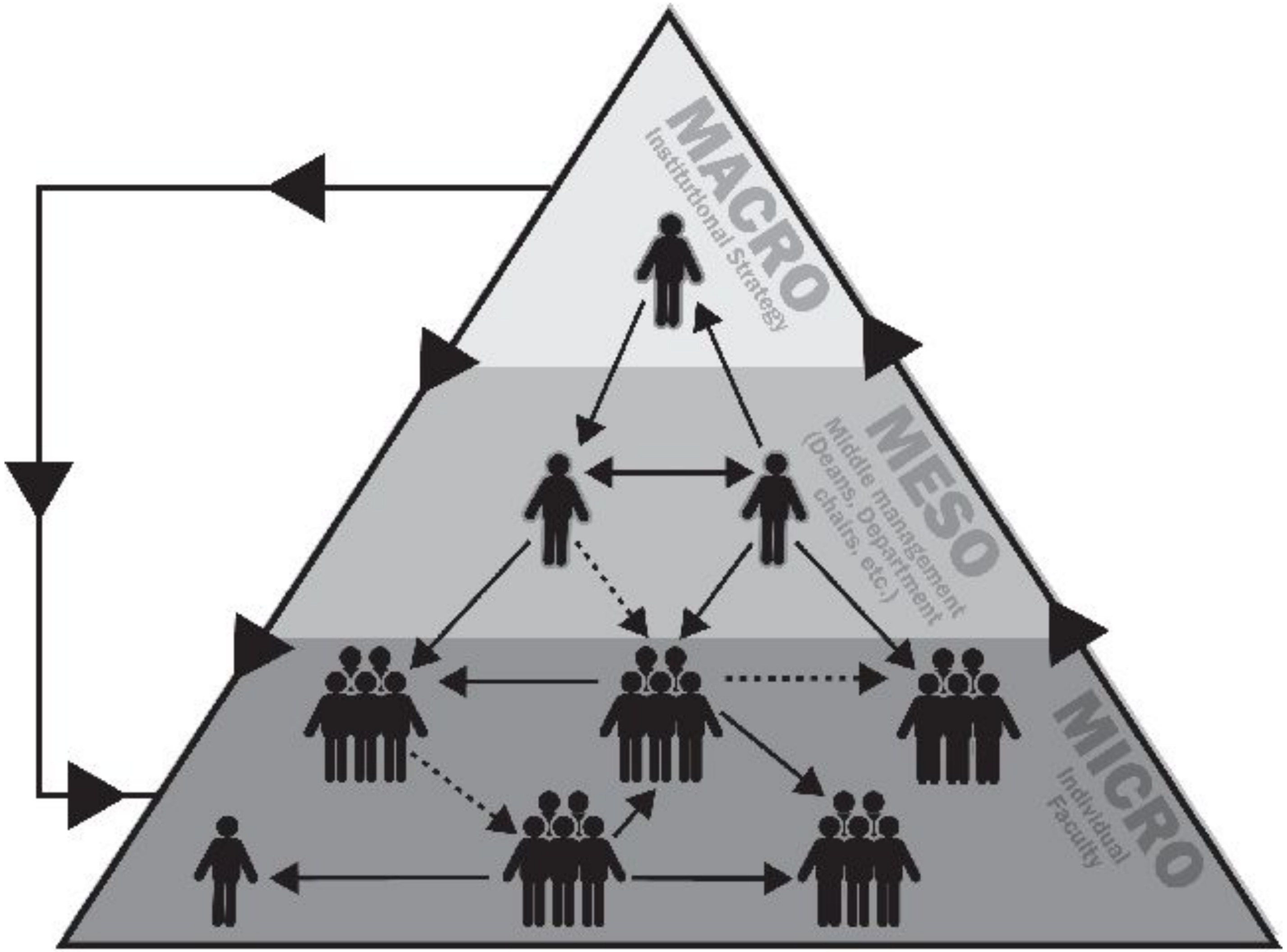
- **Application programming interfaces (APIs)** are the first software tools to access a computer, website, or cloud platform.
- These APIs enable users or **programmers to start using the system.**
- Social media APIs are used in **social networking, *instant messages, dating services, personal, location services, hobbies, travel, crowdsourcing, blogging, chat, messaging, and Avatars, etc***

Social media application programming interfaces (APIs)

API Name	Functionality	Protocol Applied	Data Format	Security
Facebook Graph API	Facebook social graph processing, community detection, finding friends, etc.	REST	JSON	OAuth
Google+API	To provide access to Google+, a social media website with links, status, and photo options	REST	JSON	API key, OAuth
Social Mention API	Programmatic access to interact with Social Mention website, a RESTful API	HTTP	PHP	API key
Delicious API	Allows users to access, edit, and search for bookmarks	REST	JSON, RSS	OAuth, HTTP/ Basic
MySpace API	To access various MySpace functions and integrate application into MySpace	Javascript	Unknown	OAuth
Meetup API	To use the topics, groups, and events created by Meetup in their own applications	REST	JSON, XML KML, RSS	PAith, API key
FindMeOn API v.1,0	Programmatic access to the social media search and management functions of FindMeOn.	HTTP	JSON	API key
Cisco JTAPI	Cisco Java Telephony API allows Java applications to interact with Telephony resources	SOAP, HTTP	XML	SSL Support
YouTube Data API v3.0	Perform actions available on the YouTube website	REST, HTTP	JSON	API key

Social Graph Properties and Representations

- LEVELS OF SOCIAL MEDIA NETWORKING
  - MICRO-LEVEL
  - MESO-LEVEL
  - MACRO-LEVEL



Macro Level: Sets the strategic direction

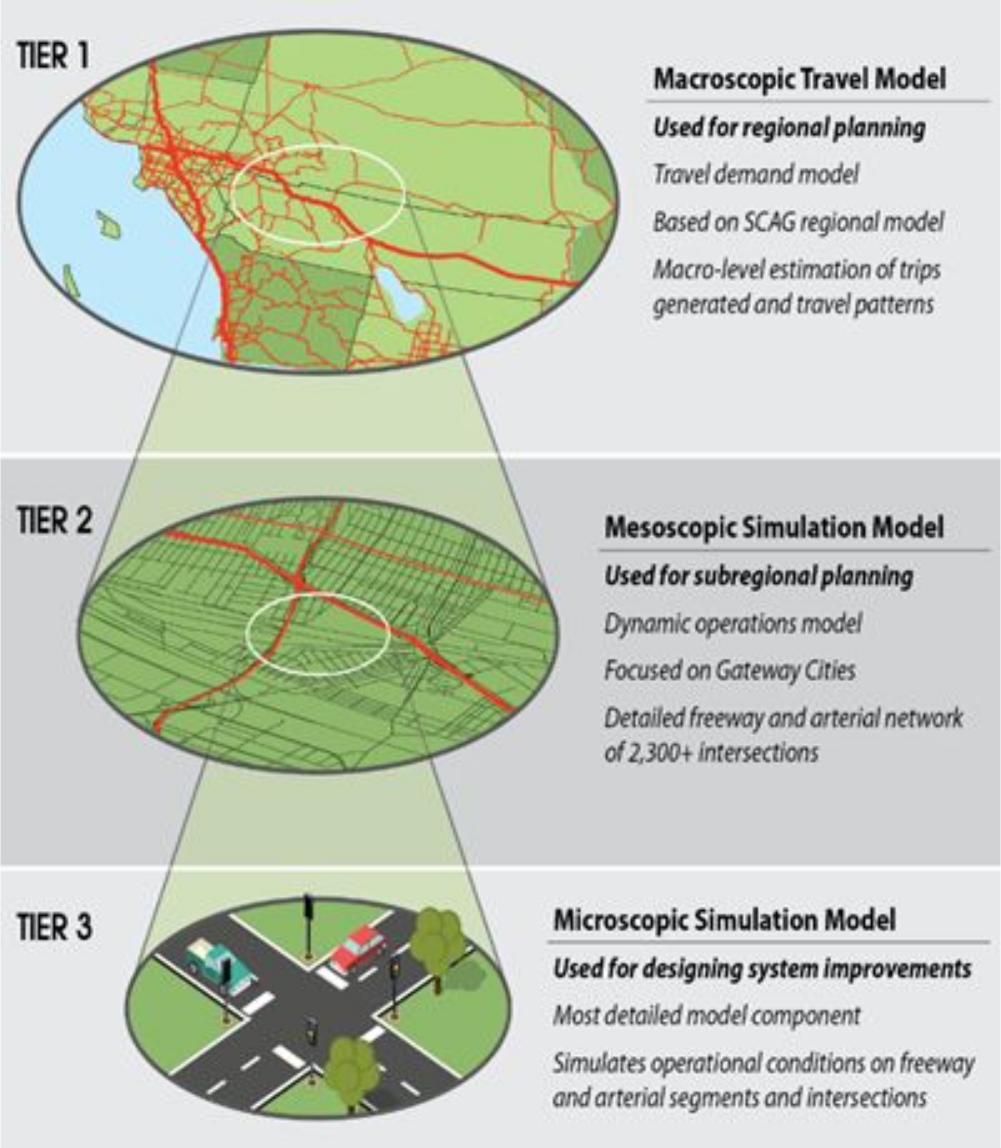
Meso Level: Interprets key issues and acts as conduit of information both upwards and down

Micro Level: Activities of communities of practice, individual faculty and students

Microsocial network	• The micro-level, social network research typically begins with an **individual**, snowballing as social relationships are traced, or may begin with a **small group of individuals** in a particular social context. • On average, the small group has a hundred or fewer peer nodes. • Member nodes in the **same group may have close** ties with many Edge connections. • Different communities are loosely connected with many fewer edge connections
Macrosocial networks	• Rather than tracing interpersonal interactions, macro-level social networks generally expand from the outcomes **of greater interactions, such as economic or other resources**. • Large-scale networks are a term somewhat synonymous with "macro-level" social networks. • These are often used in **social or behavioral sciences** in connection with economics classes, professional societies, or political affiliations

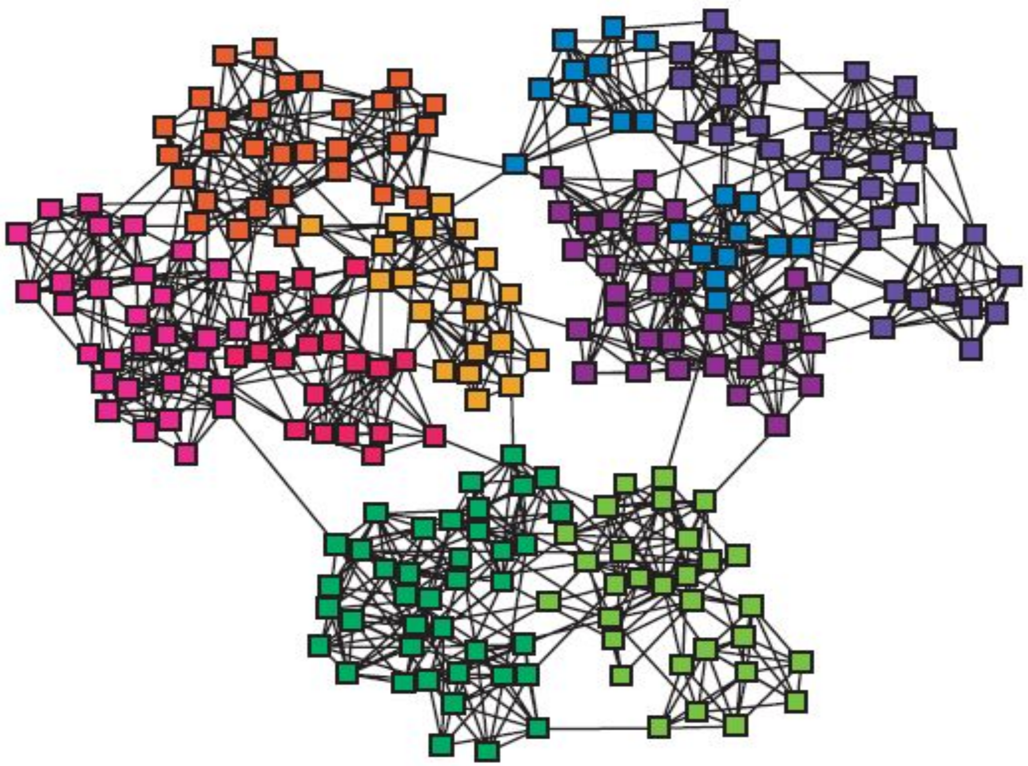
Mesosocial networks	• The meso-level theories begin with a population size that falls between the **micro and macro-levels**. • However, meso-level may also refer to networks that are specifically designed to reveal connections between micro- and macro-levels. • Meso-level networks are low density and may exhibit causal processes distinct from interpersonal micro-level networks. • The macro-level network graph may far exceed the cutoff boundary being shown at all sides of the network. • The circled network groups are at the micro-level, while • The thick linkages among micro groups correspond to the meso-level connections. • Several micro networks tied to a few central nodes form the so-called meso networks.

For scenario based questions

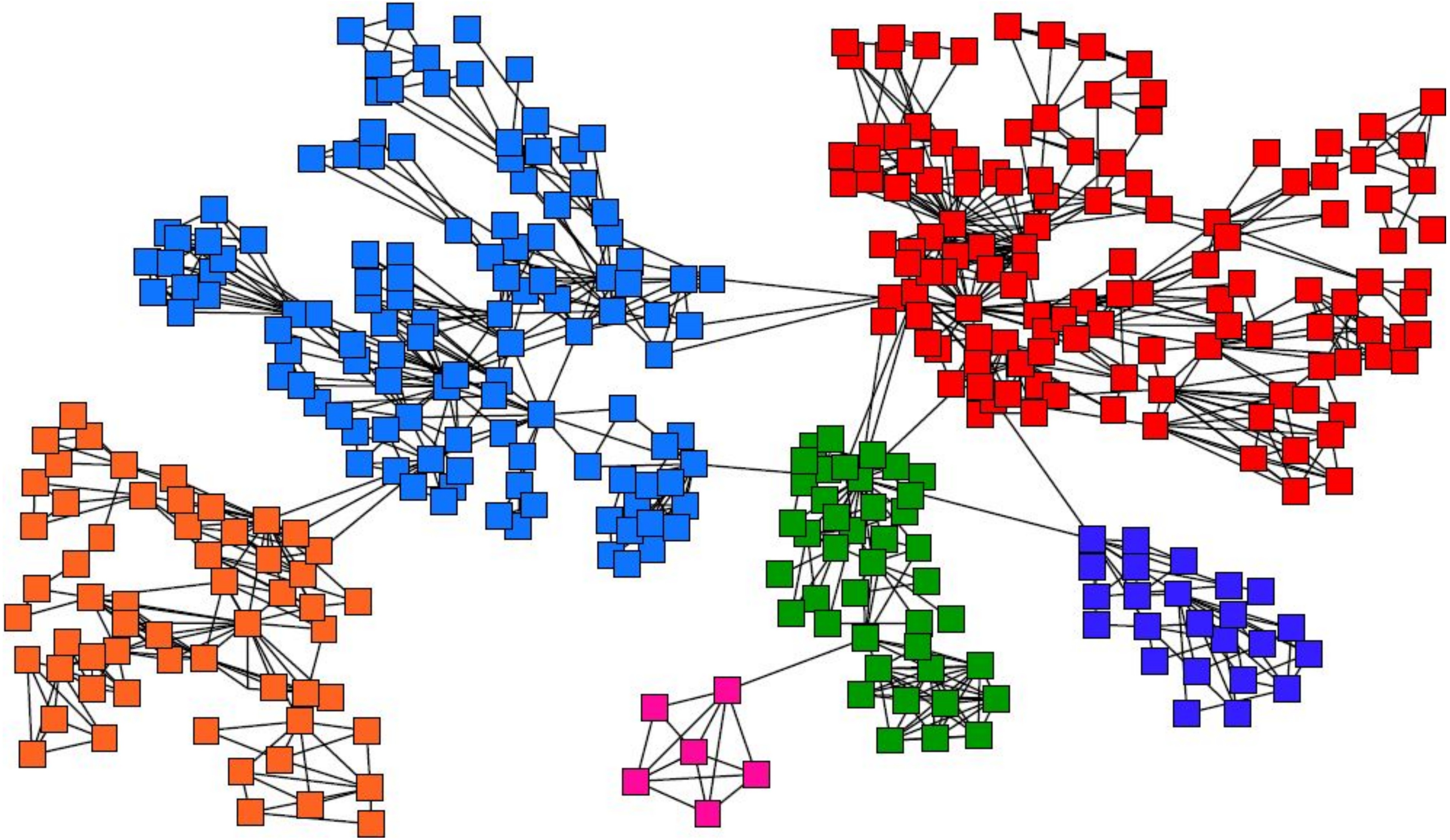


Networks & Communities

- **We often think of networks being organized into modules, cluster, communities:**

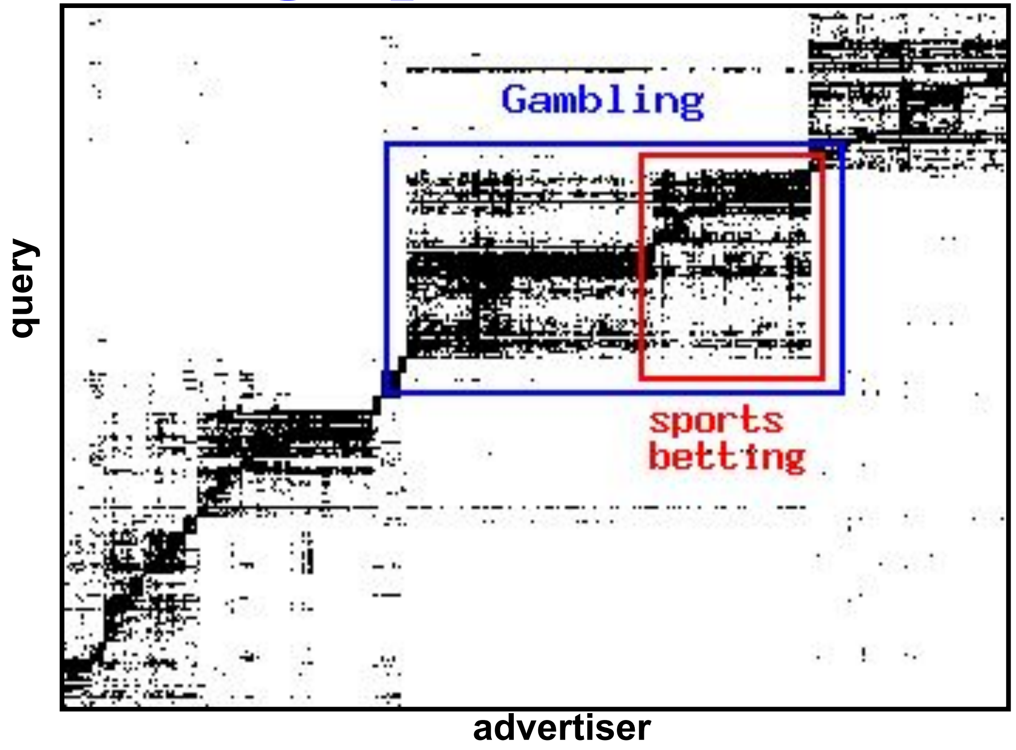


Goal: Find Densely Linked Clusters



Micro-Markets in Sponsored Search

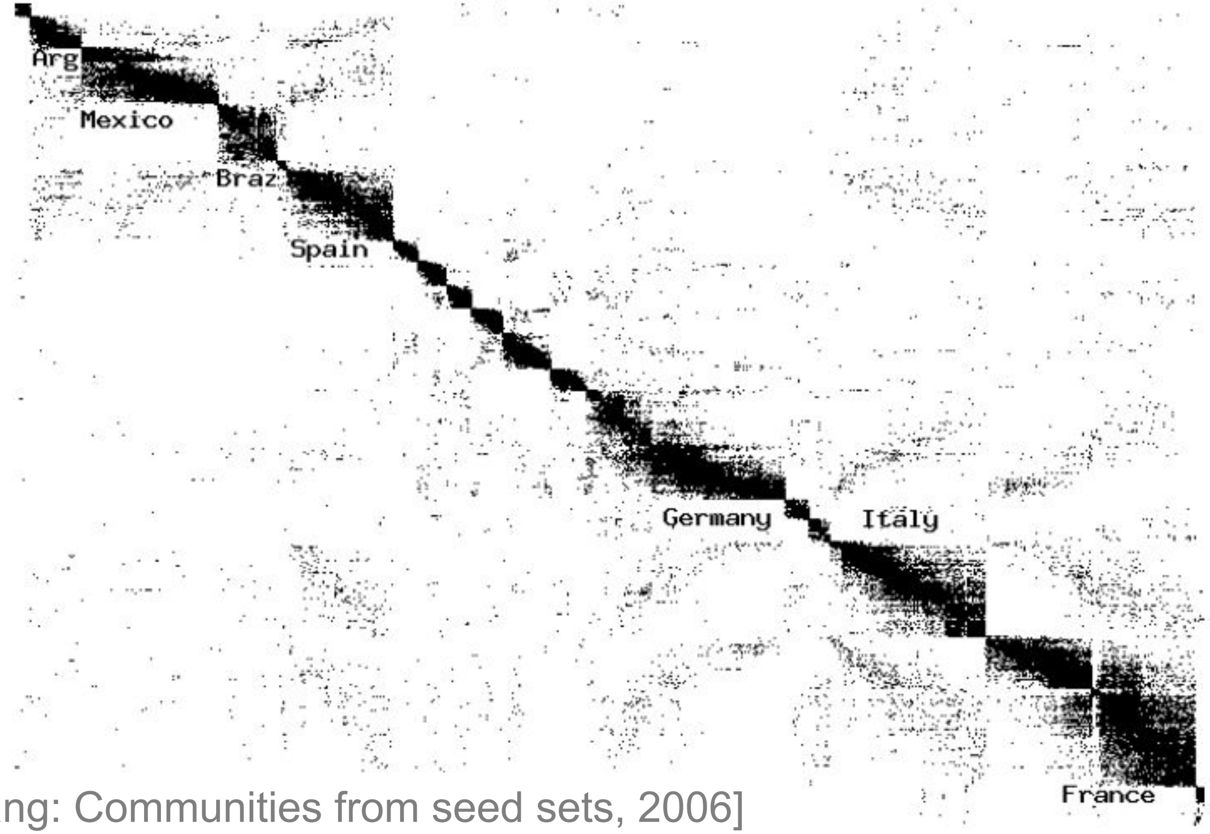
- **Find micro-markets by partitioning the query-to-advertiser graph:**



[Andersen, Lang: Communities from seed sets, 2006]

Movies and Actors

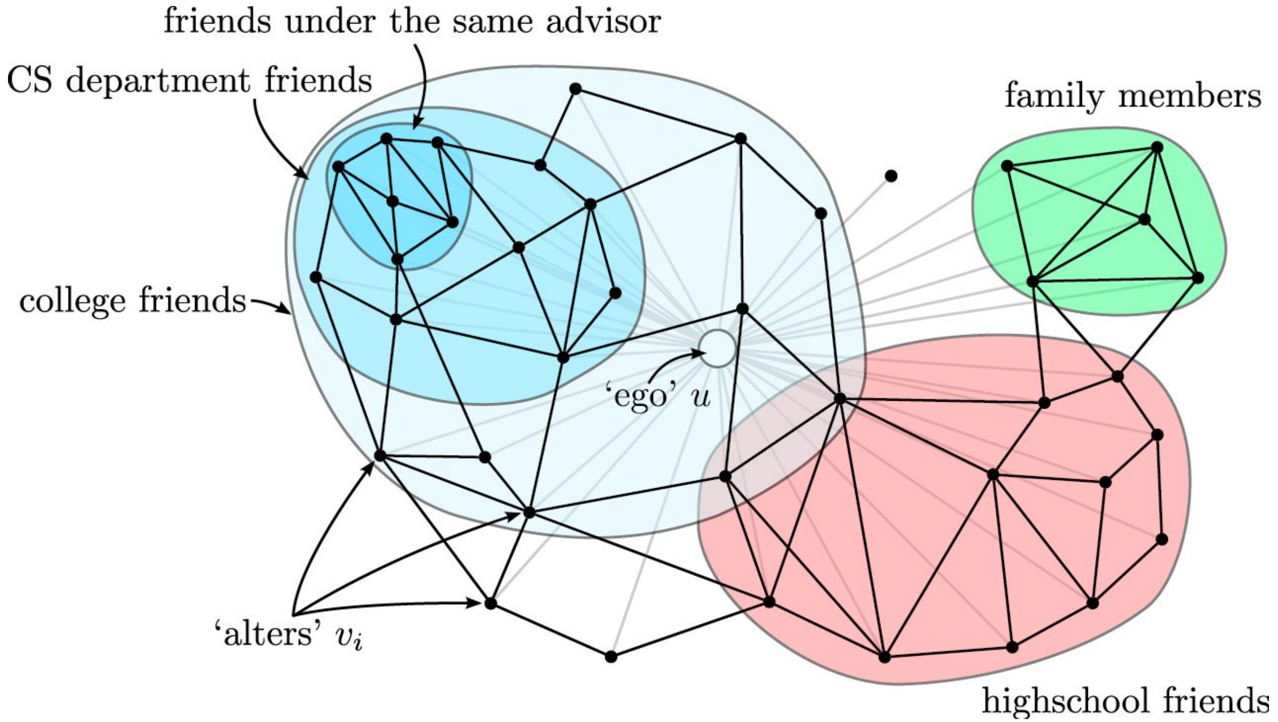
- **Clusters in Movies-to-Actors graph:**



[Andersen, Lang: Communities from seed sets, 2006]

Twitter & Facebook

- **Discovering social circles, circles of trust:**



[McAuley, Leskovec: Discovering social circles in ego networks, 2012]

Varieties of Social Networks

- Telephone Networks
- Email Networks
- Collaboration Networks

Social Graph Characteristics

- All social networks are not **so chaotic or random** as once assumed, but rather they have underlying structures.
- Social relationships are often mapped into **directed or undirected graphs**, sometimes called acquaintance graphs or simply social connection graphs.
- The nodes in a **social graph correspond to the users or actors and the graph edges or links** refer to the ties or relationships among the nodes.
- The graphs can be **complex and hierarchically** structured to reflect relationships at all levels.
- There can be many kinds of ties between the nodes.
- Social networks operate from the family level up to national and global levels.

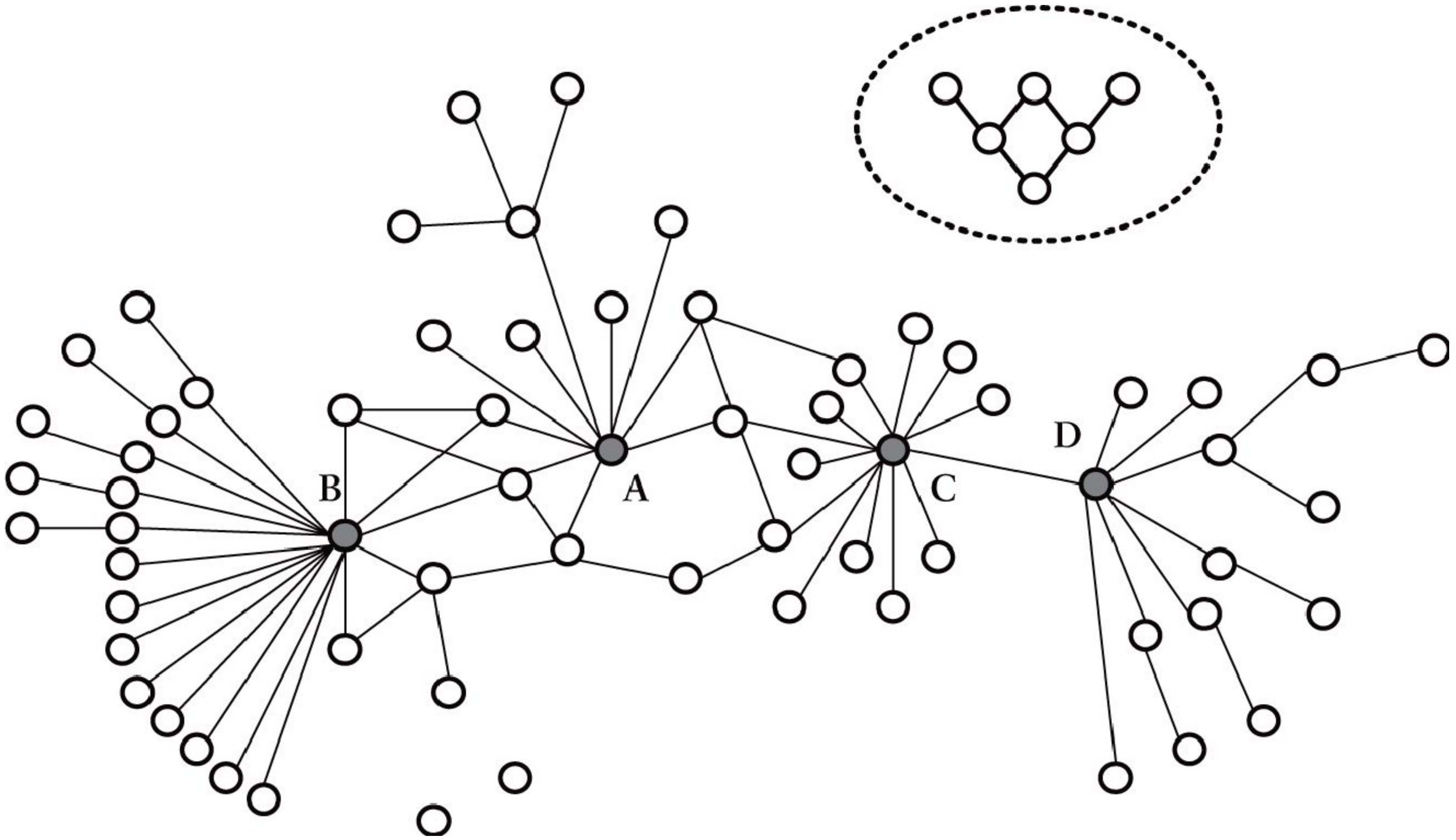
Social Network Graph Properties

- Social networks play a critical role in problem-solving, running an organization, and the **degree to which individuals succeed in achieving their goals**.
- A social network is simply a map of all **of relevant ties between all actor nodes.**
- The network can also be used to measure social capital—the value that an individual gets from the social network.
- These concepts are often displayed in a social network graph.
- The **BLACK DOTS ARE THE NODES (USERS)** and the **edges link the nodes under specified tie relationships**.

NODE DEGREE	The *node degree* is the number of **immediate node neighbors of a node**.
REACH	The *reach* is defined as the degree to which **any member of a network can reach other members of the network.**
PATH LENGTH	• *Path length* measures the **distance between pairs of nodes** in the network. • *Average path length* is the **average of these distances between all pairs of nodes**
BETWEENNESS	*Betweenness* reveals the extent to which a **node lies between other nodes in the network.**
CLOSENESS	Closeness is the **inverse of the sum of the shortest distances** between each individual and every other person in the network
COHESION	Cohesion is the degree to which actors are **connected directly to each other by cohesive bonds.**
CENTRALITY	Centrality indicates the social power of a node based on how well they **"connect" the network.**
SOCIAL CIRCLES OR CLUSTERS	If there is **less severity of direct contact or structurally cohesive blocks**, then a social circle can be created either **loosely or tightly**, depending on the stringency rules applied Clustering coefficient is the likelihood that two associates of a node are associates themselves.

CENTRALIZED VS. DECENTRALIZED NETWORKS	• Centrality gives a rough indication of the **social power of a node based on how well they "connect" the network.** • Betweenness, closeness, and degree are **all measures of centrality**. • Centralized networks have their links dispersed around one or a few nodes, while a decentralized network is one in which there is little variation between the number of links each node possesses
BRIDGE AND LOCAL BRIDGE	An **edge is a bridge** if deleting it would cause its endpoints to lie in different clusters or components of a graph
PRESTIGE	• Prestige describes a node's centrality. • **Degree prestige, proximity prestige, and status prestige** are all measures of prestige.
RADIALITY	Radiality is the **degree a network reaches out and provides novel information and influence**.
STRUCTURAL COHESION	Structural cohesion is the minimum number of members who, **if removed from a group, would disconnect the group**
EQUIVALENCE	• **Structural equivalence** refers to the extent to which nodes have a common set of linkages to other nodes. • These nodes do not have any ties to each other.
HOLE	A **structural hole can be filled by connecting one or more links to reach other nodes**.

The graph representation of account holders in a social network



Social Graph Analysis on Smart Clouds

- Ideas for providing online social networking services

- *Personal page or profiles for each user are linked by social connections.*
- *There is* ***social graph traversal*** *along specific social links or networks.*
- *Communication tools are shared between the participants or registered users.*
- *Special information like music, photos, videos, etc., is shared with friends or professional groups.*
- *Communities operate in special niche topic areas such as healthcare, sports, hobbies, etc.*
- *Customized software tools or databases are used to set up social network services.*
- *Strong customer loyalty creates viral membership growth.*
- *Social networks have revenues by selling premium memberships and access to premium content.*

ONLINE SOCIAL NETWORK (OSN)

- The social network provider should choose a brand name with its own **API interfaces and profile variables**.
- The chosen forum categories must be relevant to a **sufficiently large user community.**
- The **OSN platform** must include specific functionalities that make it easy for users to **join and enjoy the services**.
- The social network community must operate reliably with **high availability and performance**.

- Logically, the **OSN provides a P2P platform**.
- However, modern popular social networking services are all built with **client-server architecture** for easy management and maintenance.
- This means that all **blog entries, photos, videos and social network relations are stored and managed** by private clouds owned by the service providers

Filtering Techniques and Recommender Systems	• **Recommender systems** in order for movies, tourism, or restaurants to make our daily life activities better organized, convenient, and enjoyable. • **Social or collaborative filtering** of unwanted data can be Done by polling the opinions of the users to make decisions based on ratings. • **Content-based filtering** is needed in recommending items based on features of products and ratings by users. • **Demographic filtering** helps decision making based on the demographic information of the users. • Finally, **knowledge-based filtering** makes decisions based on expertise or peer reputation. • **Hybrid filtering** combines the advantages of the above filtering techniques to make even smarter decisions.
Pushing Data Analytics for Cloud/Network Security Enforcement	• **Big data analytics** is needed in network security, enterprise events analytics, and netflow monitoring to identify botnets, persistent threats, data sharing, provenance, and governance techniques that are often wanted for trust management with reputation systems.

Cloud Support of Social Network Applications	• **Cyber-physical systems (cps**), analytical algorithms can perform more accurately in system configuration, physical knowledge, and working Principles. • To **integrate, manage, and analyze machinery**, it is important to handle data more efficiently during the different stages of the machine life cycle. • The **coupling model between humans and machines** is greatly facilitated by the use of cloud storage and analytics systems. • This involves sensing, storage, synchronization, synthesis, and service operations
Smart and Pervasive Use of IoT-enabled Services	• These include coordinated calendar, itinerary, job management, events, and **consumer record management (CRM) services.** • Other interest areas include cooperative word processing, online presentations, web-based desktops, sharing online documents, data sets, photos, video, databases, and content distribution. • Deployment of conventional **clusters, grids, P2P, and social networking applications are very much in demand in cloud environments**

Multicloud Mashup Architecture and Service

- A cloud mashup is composed of **multiple services with shared data sets and integrated functionalities.**
- The **EC2 provided by Amazon Web Service (AWS**), the authentication and authorization services provided by Facebook, and the MapReduce service provided by Google can all be mashed up to deliver real-time, personalized driving route recommendation services.

Basic idea to address Scenario based Questions

- Cloud+(IOT or MCC or SM)=MASHUP

- CLOUD: Components, Deployment, Services, Storage, Computation

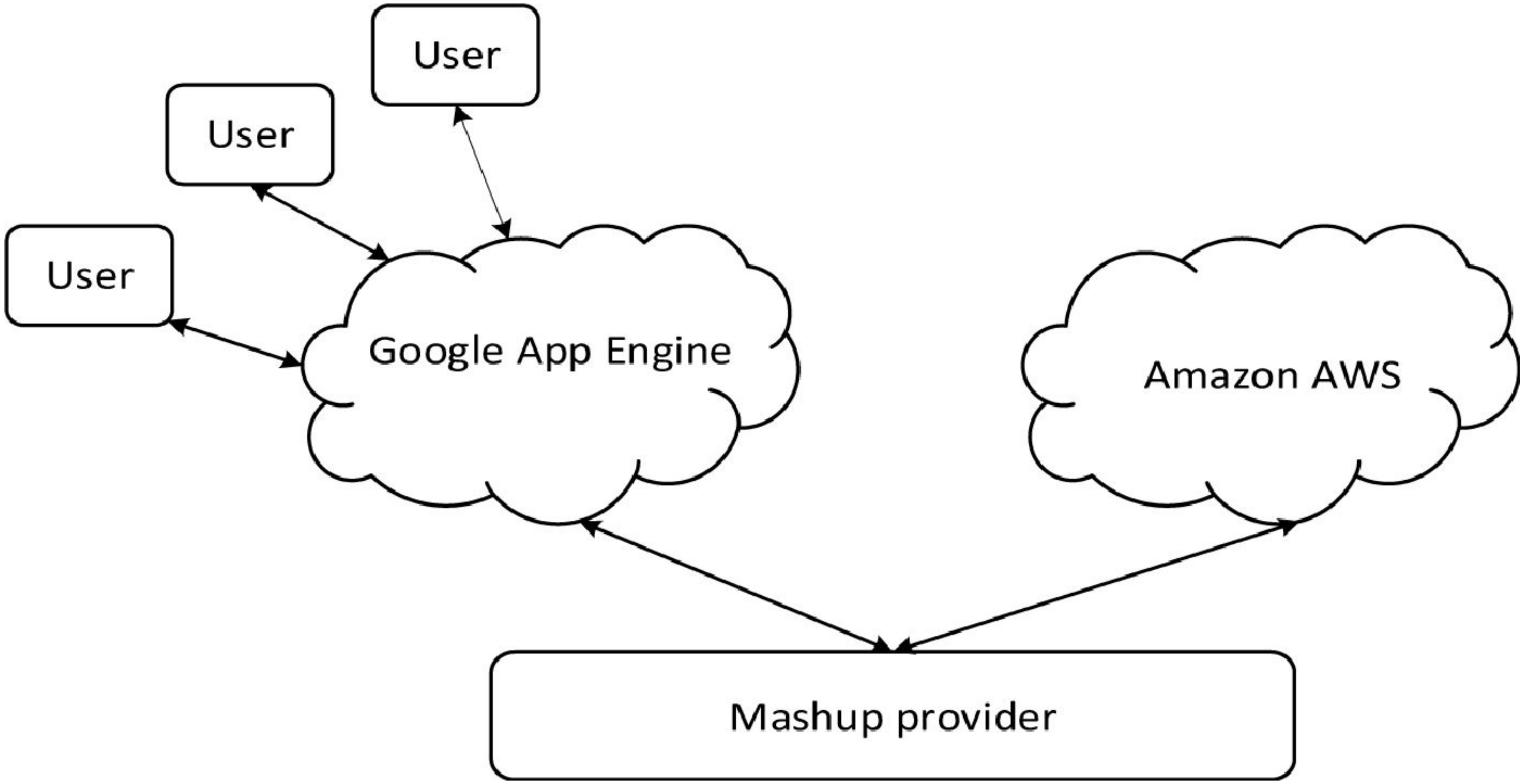
- To discover qualified services and compose them with guaranteed QoS, we propose an **INTEGRATED SKYLINE QUERY PROCESSING METHOD FOR BUILDING UP CLOUD MASHUP APPLICATIONS**

- The **Open Mashup Alliance (OMA)** is a nonprofit association that promotes the adoption of mashup solutions in the enterprise through the evolution of enterprise mashup standards like **Enterprise Mashup Markup Language (EMML).**

Cloud Mashup Architecture for Agility and Scalability

- The main characteristics of **the mashup are COMBINATION, VIRTUALIZATION, AND AGGREGATION.**
- The **AWS and GAE (*Google App Engine*)** clouds differ not only in their functionalities but also can **complement each other for better purposes**

The Idea of Cloud Mashup (The mashup between google app engine and amazon cloud platform)



Advantages of Cloud Mashup

Scalability on the EC2 virtual clusters	**Scaling the implementation** of MapReduce can be done in two dimensions: **problem size and virtual cluster size.**
Agility in using the AppEngine interfaces	Mashup platform is built by combined use of **google app engine and amazon web services** Essentially using the **web interfaces by google** and the **computing power** from amazon EC2

Multicloud Mashup Service Architecture

- Multicloud Mashup Service Architecture
  - Business (or enterprise) mashups
  - Consumer mashups
  - Data mashups

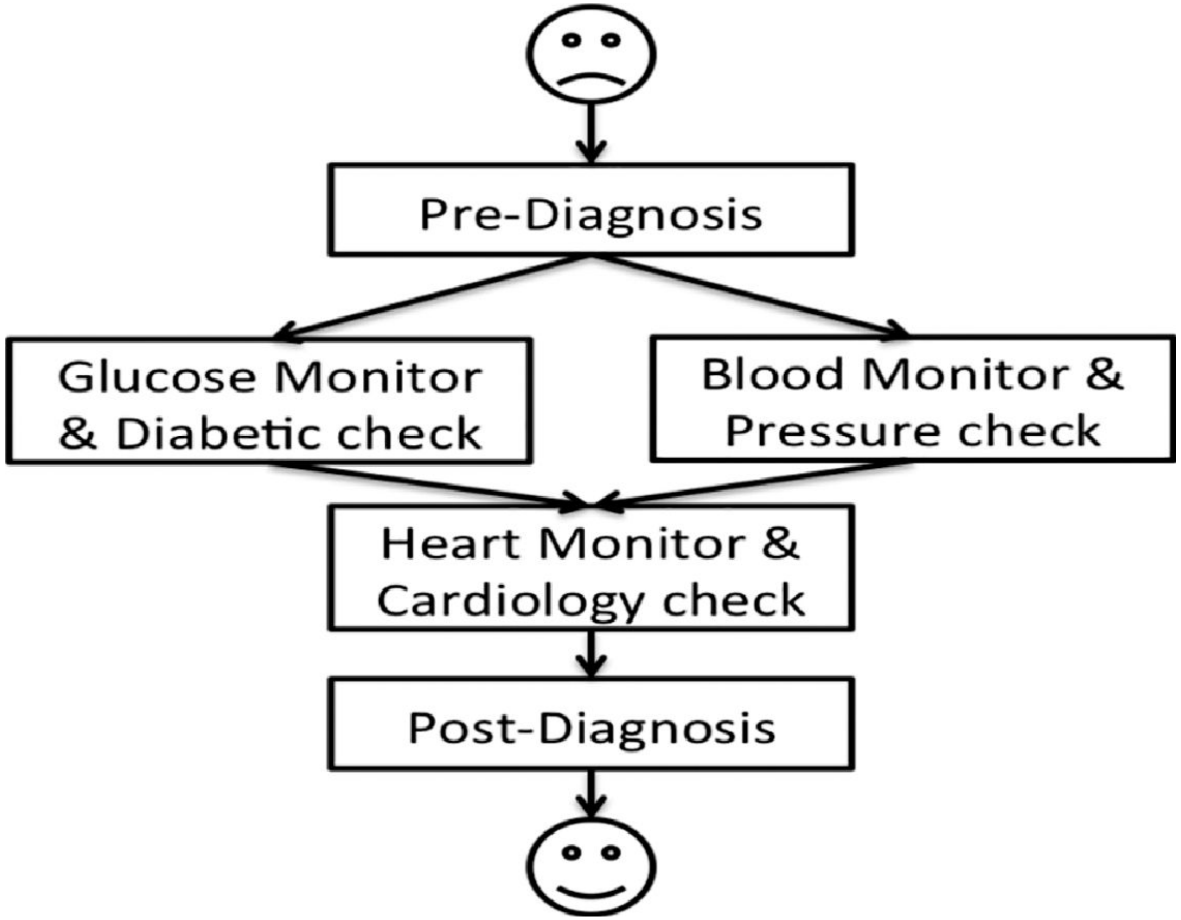
Scenario 1

- Google Maps is one of the most popular services that is used by a lot of other companies/services. Applications that provide information about road conditions, using maps to show social media friends' location, providing ratings of different cities, etc. are some services that combine Google's data and use it in their own application.

Scenario 2

- Amazon e-commerce is another one of the most popular services that some mashup websites use. Their API can be used in applications such as viewing product availability over different websites, cost comparison of a particular product over different websites, etc.

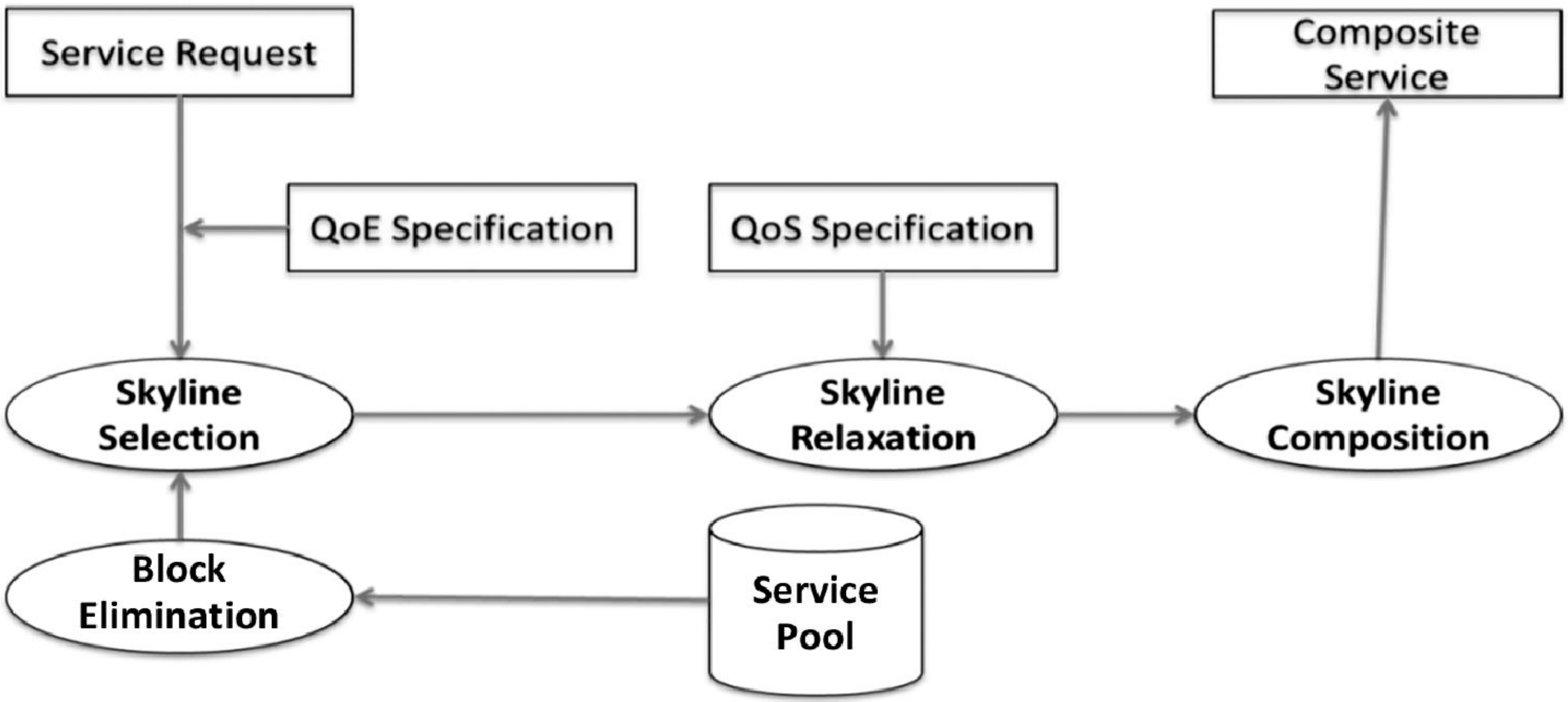
Mashup of Multiple Cloud Services in Healthcare Applications



Mashup Architecture Specification

- **PRESENTATION LAYER:** This is the user interface of mashups. The technologies used include *HTML/XHTML, CSS, JavaScript, Asynchronous JavaScript and XML (Ajax).*
- **WEB SERVICES:** A product's functionality can be accessed using API services. Popular tools used are XMLHttpRequest, XML-RPC, JSONRPC, SOAP, and REST.
- **DATA LAYER:** This layer handles the data such as sending, storing, and receiving. The tools are XML, JSON, and KML.

Skyline discovery and composition process for intercloud mashup services aided by similarity testing with **QoS and QoE assurance**.



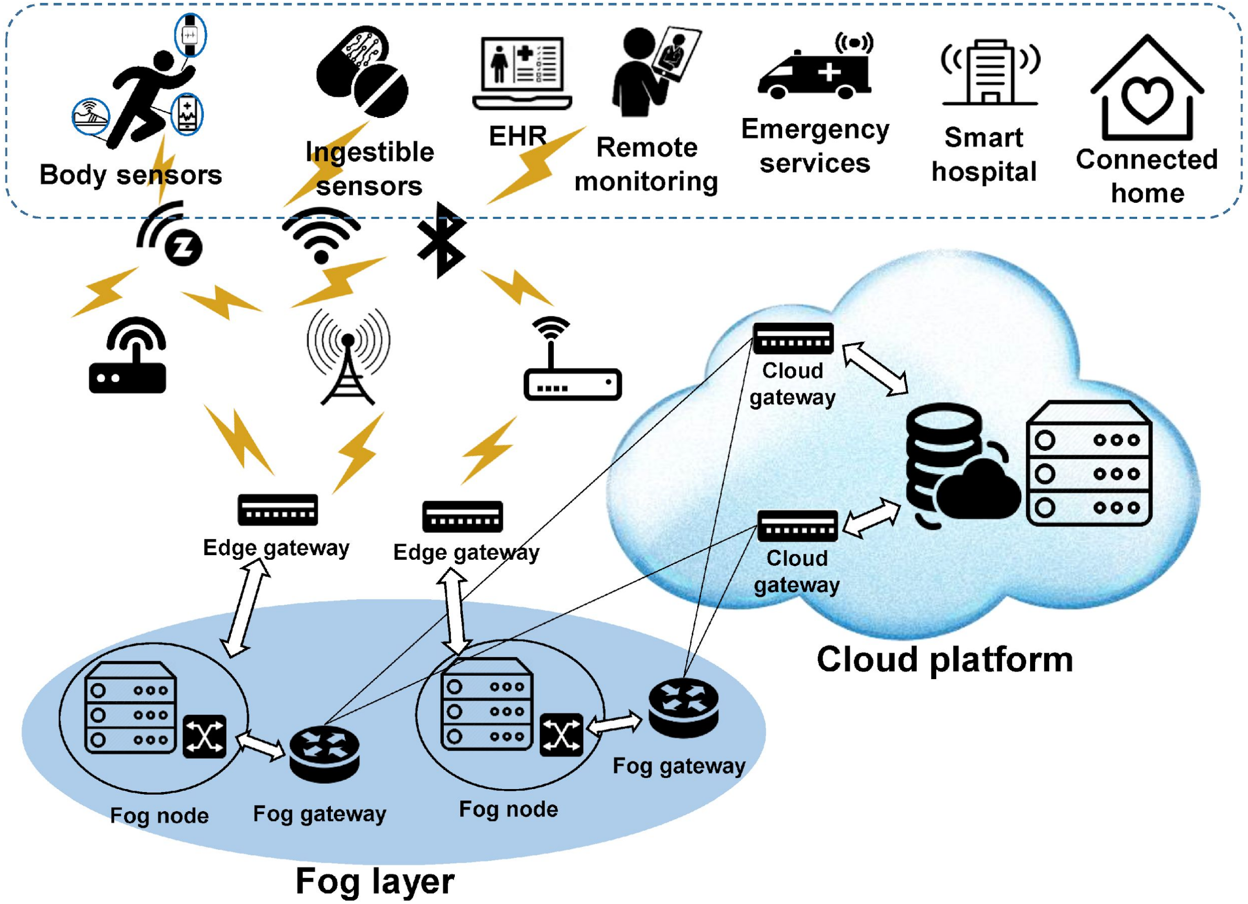
Quality of Mashup Service (QoMS)

- QoMS directly evaluates different **performance metric attributes of composite mashup services**.
- The duration of composite service is neither the optimal nor the actual duration.
- The first three attributes of a **composite service, waiting time, service time, and cost,** depend not only on those of its elementary tasks but also on the operations in between, while the last three, i.e., **reputation, reliability, and availability, are derived from its elementary attributes.**

Quality of Experience (QoE)

- How customers are satisfied with the solutions provided by the composite service is a **critical part of the evaluation of QoE.**
- The whole medical plan made by the planner is the solution of the composite service, and the quality of the medical plan depends on the solutions of each task ti, that is, the medical treatment applications, the cloud service providers, etc.
- People may argue that "reputation" can incorporate how users are satisfied, but it is the quality of service that matters most.

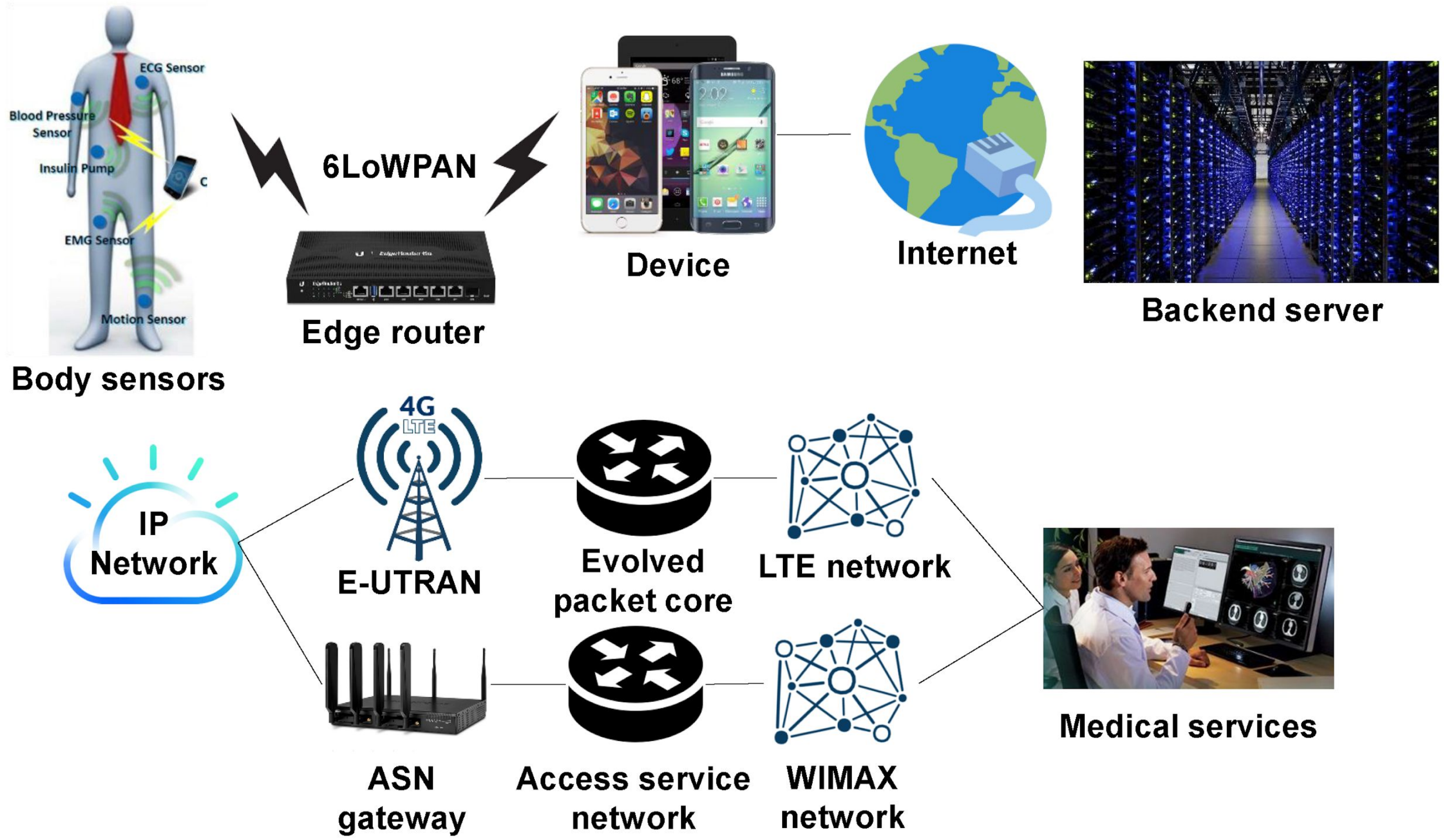
Sample diagram for building the architecture



SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

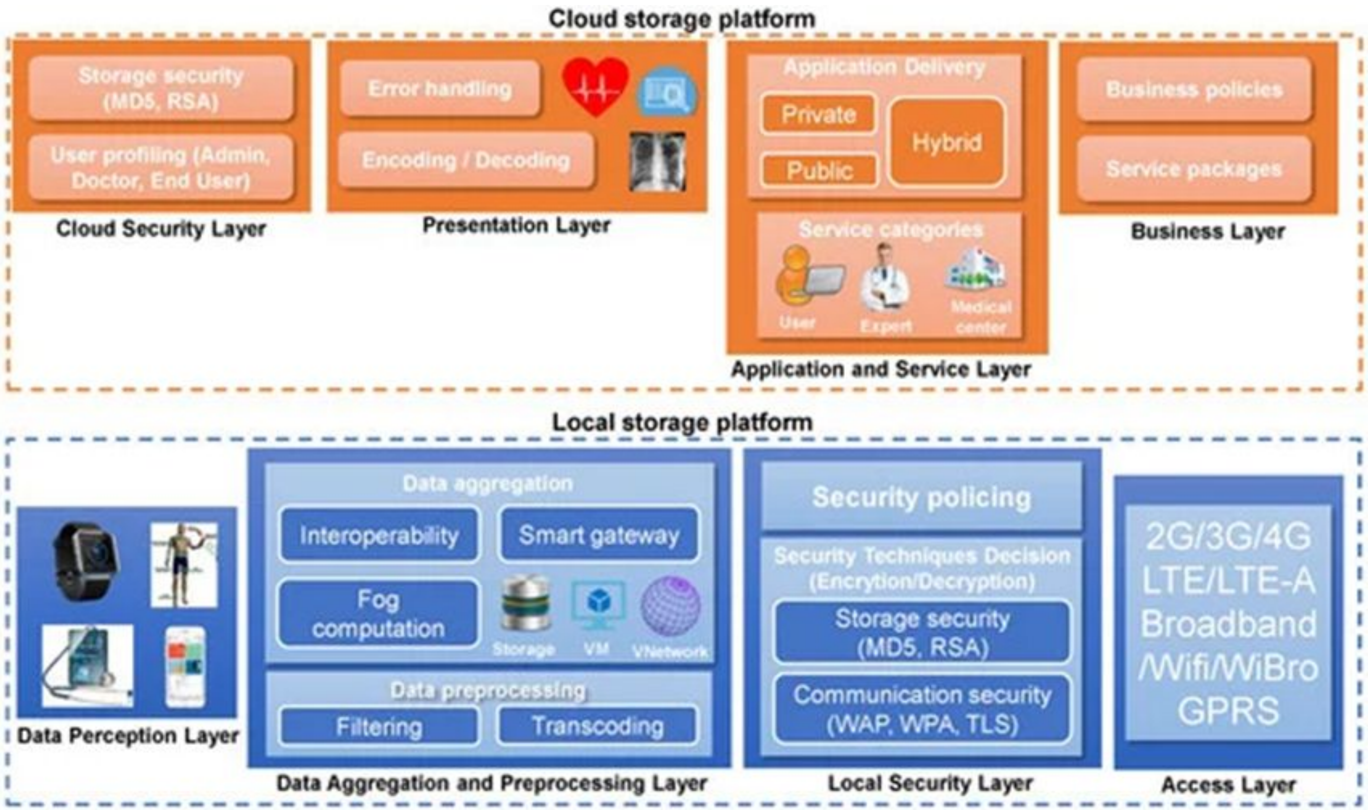
IoT framework for healthcare

Topology for remote patient monitoring using body sensors (top part) with two standard communication techniques (bottom part)

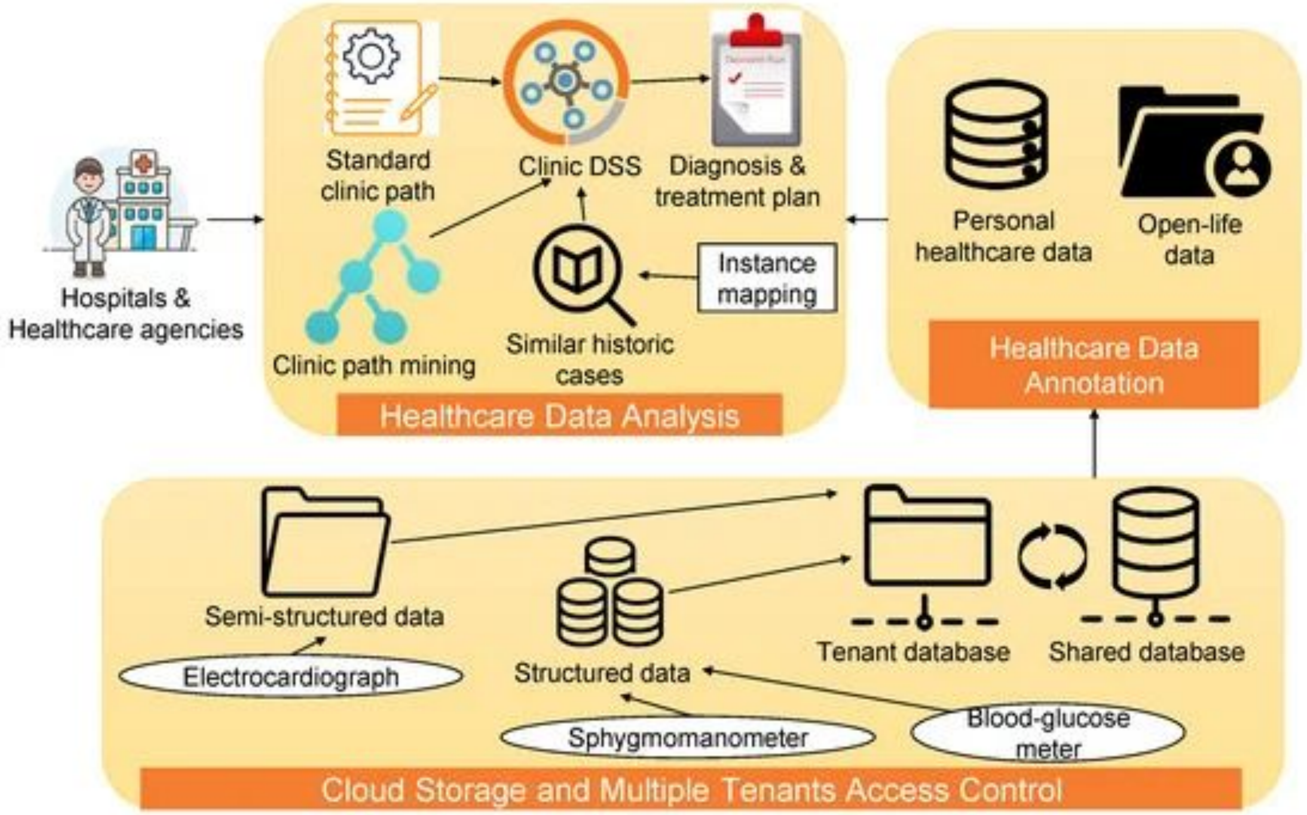


Common IoTHeF structure in a medical sensor network which includes smart healthcare gateways.

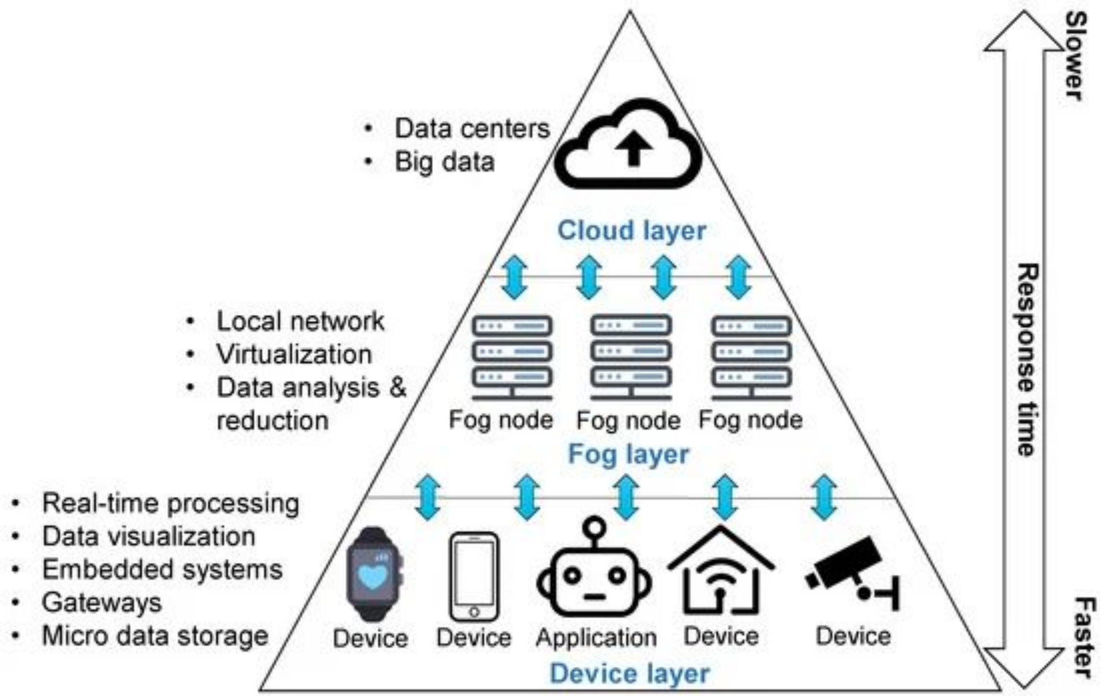
IoTHeF platform including local storage and cloud storage platforms



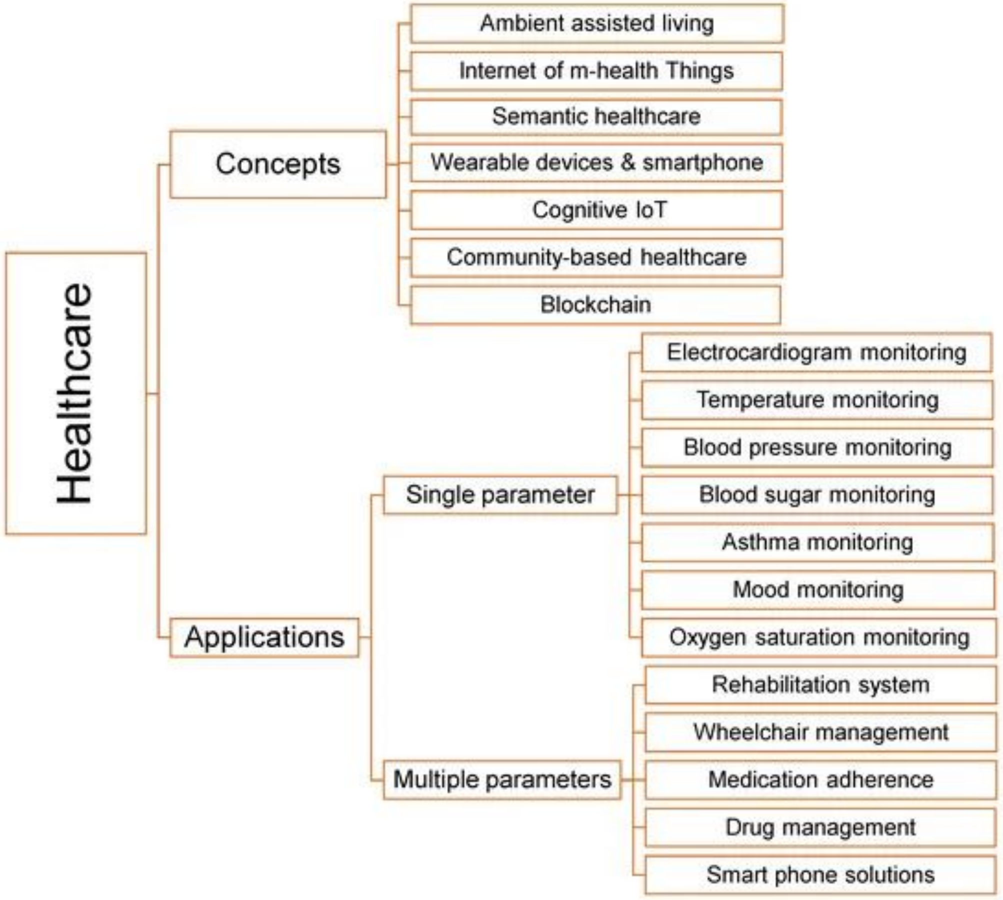
Functional platform of the cloud-computing-based m-Health monitoring system with three main layers' cloud storage and multiple tenants access control, healthcare data annotation, and healthcare data analysis



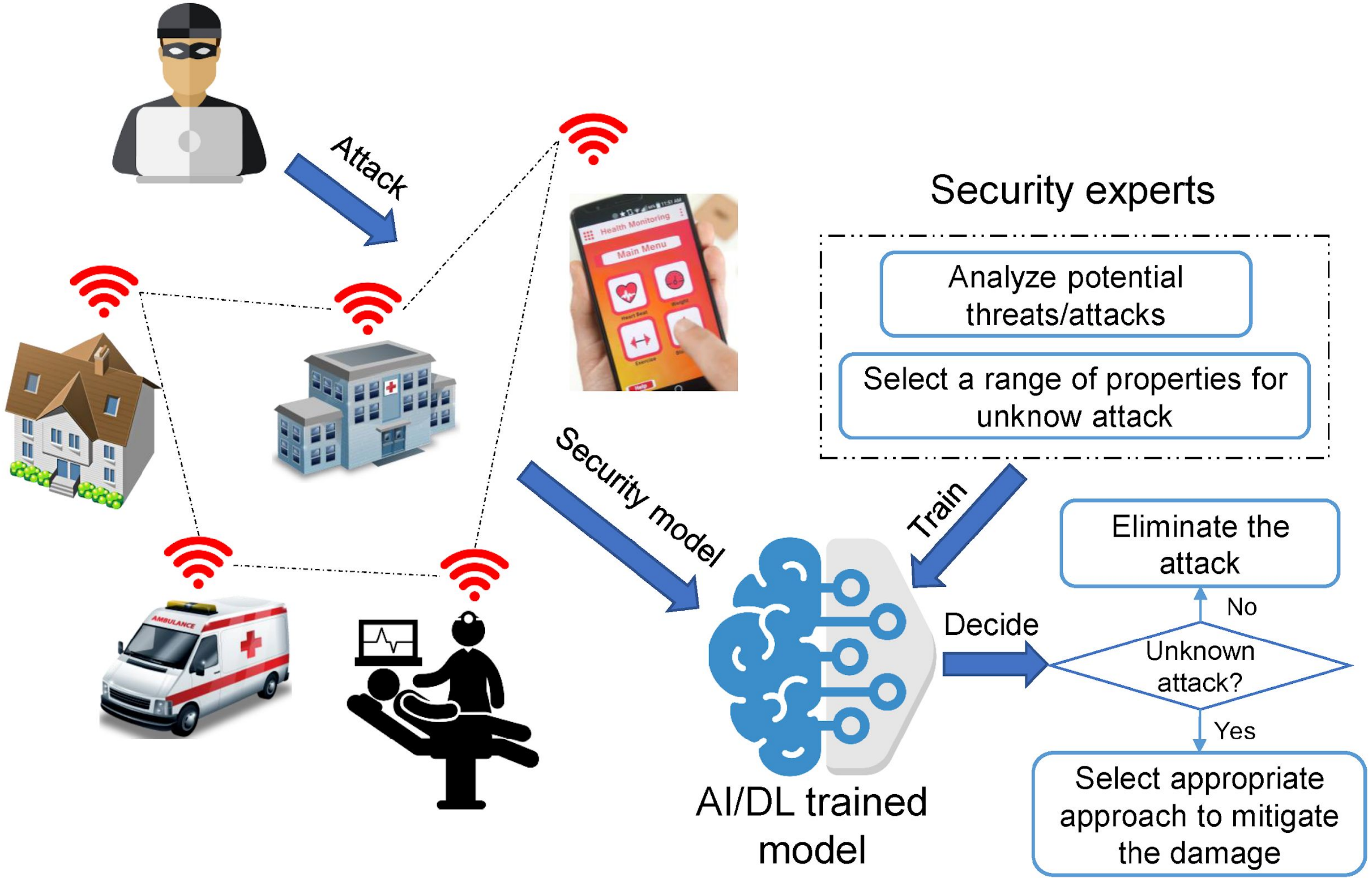
Fog computing architecture including device layer, fog layer, and cloud layer and their main attributes.



Classification of concepts, single parameter applications, and multiple parameters applications for IoT in healthcare



Smart security model for IoT in healthcare frameworks



SKYLINE METHOD

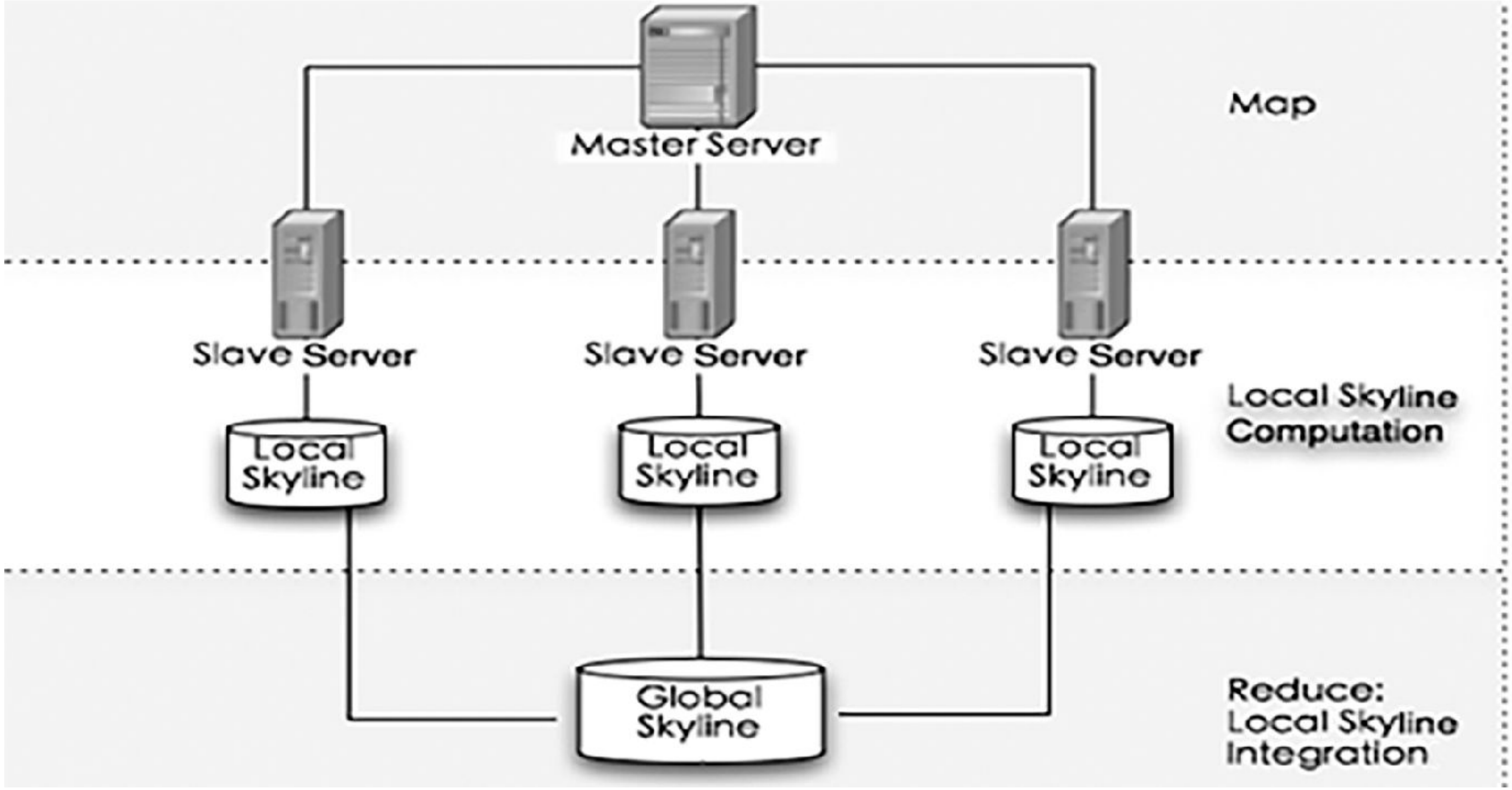
- The skyline method is especially attractive to discover qualified web services in a **multi-attribute decision-making process.**
- The **block-elimination-based partitioning method**, originally proposed in is modified to accelerate the MapReduce process with a relaxed skyline of representatives in subdivided skyline sectors.
- *skyline selection, similarity test and service composition*
- Is to optimize the mashup composition process with a MapReduce skyline selection guided by the **QoS and QoE constraints**

Skyline Discovery of Mashup Services

- Given a set **Q of data points** in d-dimensional QoS space, each dimension represents a performance attribute with values properly ordered.
- Suppose the lower-valued points are better than the higher-valued ones.
- A data point **Pj is dominated by Pi**, if Pi is better than or equal to Pj in all dimensions.
- Furthermore, **Pi must be better than Pj** in at least one dimension.
- All data points that are not dominated by any other point form a subset called the skyline.

- A skyline query selects the **BEST OR MOST INTERESTING POINTS** in all dimensions.
- MapReduce to upgrade **COMPUTING EFFICIENCY with scalable performance in large-scale skyline query processing**.
- Our approach is based on a novel **block-elimination method.**
- The elegance of the MR-block algorithm lies in its capability to **reduce the search space** from thousands of points to hundreds, leading to the Map phase processing on a much reduced dataset.
- Furthermore, we propose a variant of the MapReduce method by adding a process between **Map and Reduce**

MapReduce model for selecting skyline services to optimize QoS



- A Simple Solution is to **initialize skyline or result as empty**, then one by one add buildings to skyline.
- A building is added by first finding the overlapping strip(s).
- If there are **no overlapping strips**, the new building adds new strip(s).
- If overlapping strip is found, then height of the existing strip may increase.
- Time complexity of this solution is O(n2) We can find Skyline in Θ(nLogn) time using **Divide and Conquer**.

- The idea is similar to Merge Sort, divide **the given set of buildings in two subsets.**
- Recursively construct skyline for **two halves and finally merge the two skylines.**
- How to Merge two Skylines? The idea is similar to merge of merge sort, **start from first strips of two skylines, compare x coordinates.**
- Pick the strip with **smaller x coordinate and add it to result**.
- The height of added strip is considered as maximum of current heights from skyline1 and skyline2.

THE MAP PROCESS	• Service data **points are partitioned by the master server** (e.g., UDDI) into **multiple data blocks based on the QoS demand**. • The data blocks are dispatched to slave servers for parallel processing
LOCAL SKYLINE COMPUTATION	• In this process, each slave server generates the **local skylines from service data points on its own subdivided data blocks**.
THE REDUCE PROCESS	• In this process, local skylines generated by all the slave servers are **merged and integrated into a global skyline**, which applies to all services being evaluated

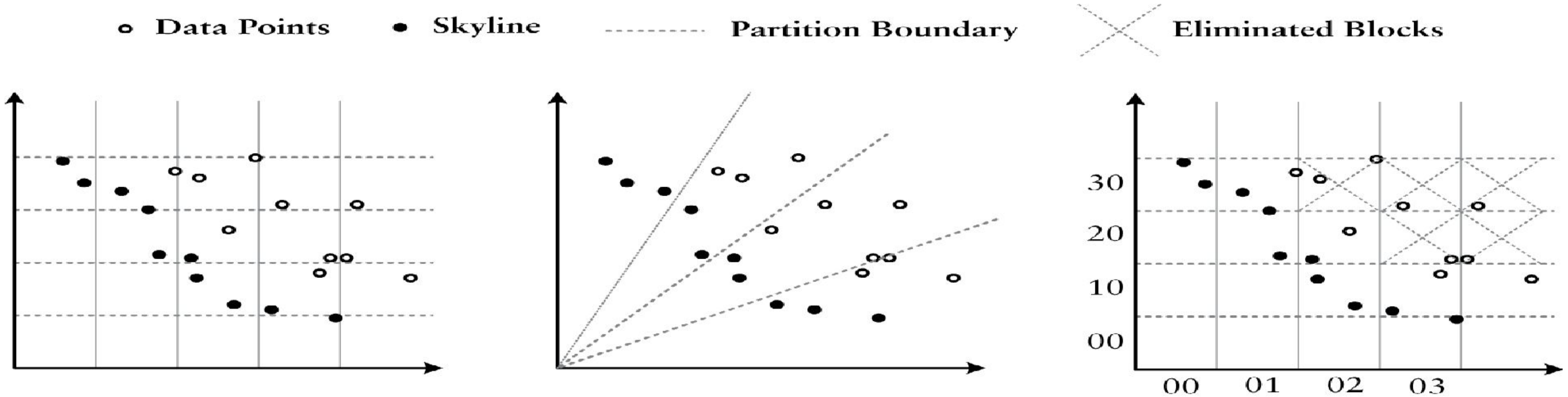
- The quality of the selected skyline service depends on the **efficiency of the local skyline computation** and the performance of the integration process.
- The efficiency and QoS of the MapReduce skyline process depends mainly on how to **explore the distributed parallelism to accelerate the Map stage**.
- The efficiency of the mapping depends on **data space partitioning**.
- The service data points are partitioned **into divided regions**.
- The goal is to **achieve load balancing**, to fit into the local memory and to **avoid repeated computations** when old services are dropped and new services are added dynamically

- MapReduce is effective to speed up the **skyline query processing process.**
- We need to compare **pairwise services in parallel**.
- With MapReduce, the new service is first **mapped into a group** and added into **the local skyline computation.**
- Then all local skylines are integrated **into the global skyline at the reduce stage.**

Example

- Consider two service **data points s1, s2, in the QoMS space Q**.
- The **service s1 dominates service s2**, if **s1 is better than or equal to s2 in all attribute dimensions of Q**.
- Furthermore, **s1 must be better than s2** in at least one attribute dimension.
- The subset S of services forms the skyline in space Q, if all service points on the skyline are better than or equal to other services along all attribute dimensions.
- In other words, all skyline services are not dominated by any other service in the space Q.

- MapReduce skyline methods, denoted as **MR-grid, MR-angular, and MRblock**, where MR stands for MapReduce in all figure labels and text body.
- Three MapReduce skyline algorithms are specified based on the three data partitioning schemes



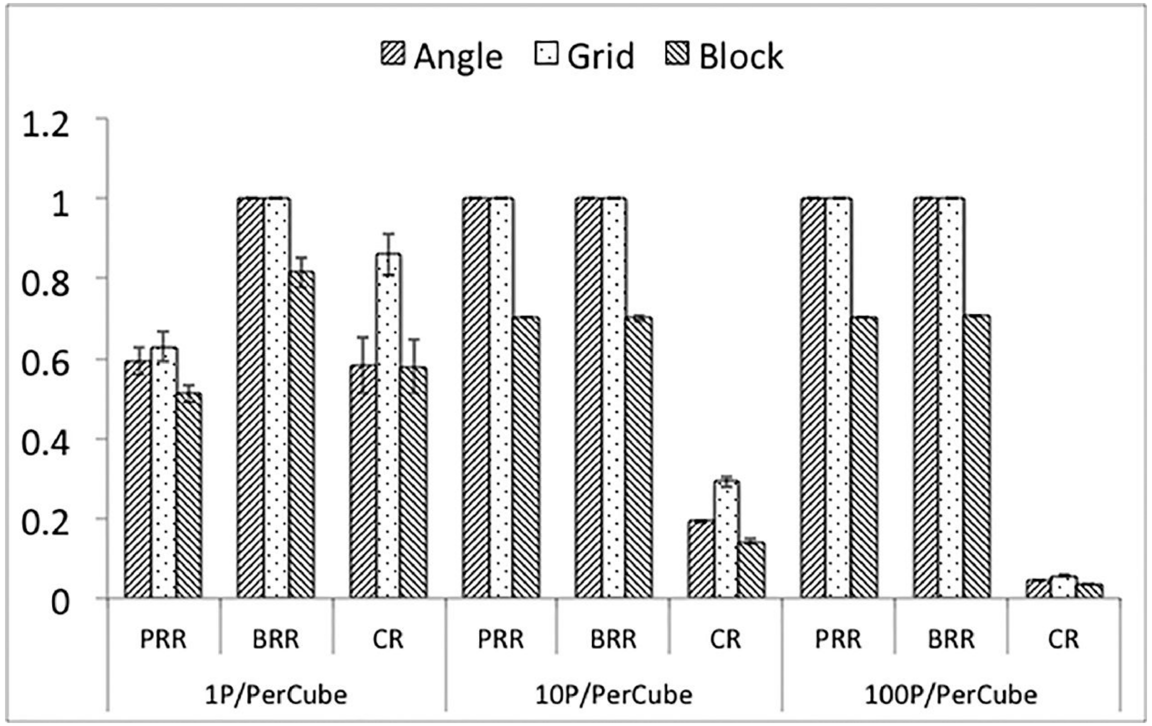
(a) Grid Partition (b) Angular Partition (c) Block Elimination Partition

- The **MR-grid algorithm** contains two stages:
- Partitioning the job, in which we divide **the data space into some separate subspaces and compute the local skyline of each subspace**
- **Merging the job**, in which we merge all local skylines to compute the global skyline.
- Empirically, the number of partitions is set as two times the nodes in the **MR-grid algorithm**.
- In the **MR-block method**, the data points in **the higher orthogonal blocks are dominated.**
- Therefore, we need not compute the local skyline in the blocks that are dominated by others.
- There is no dominance relationship **between any two angular sectors within the data space**

Dynamic Composition of Mashup Services

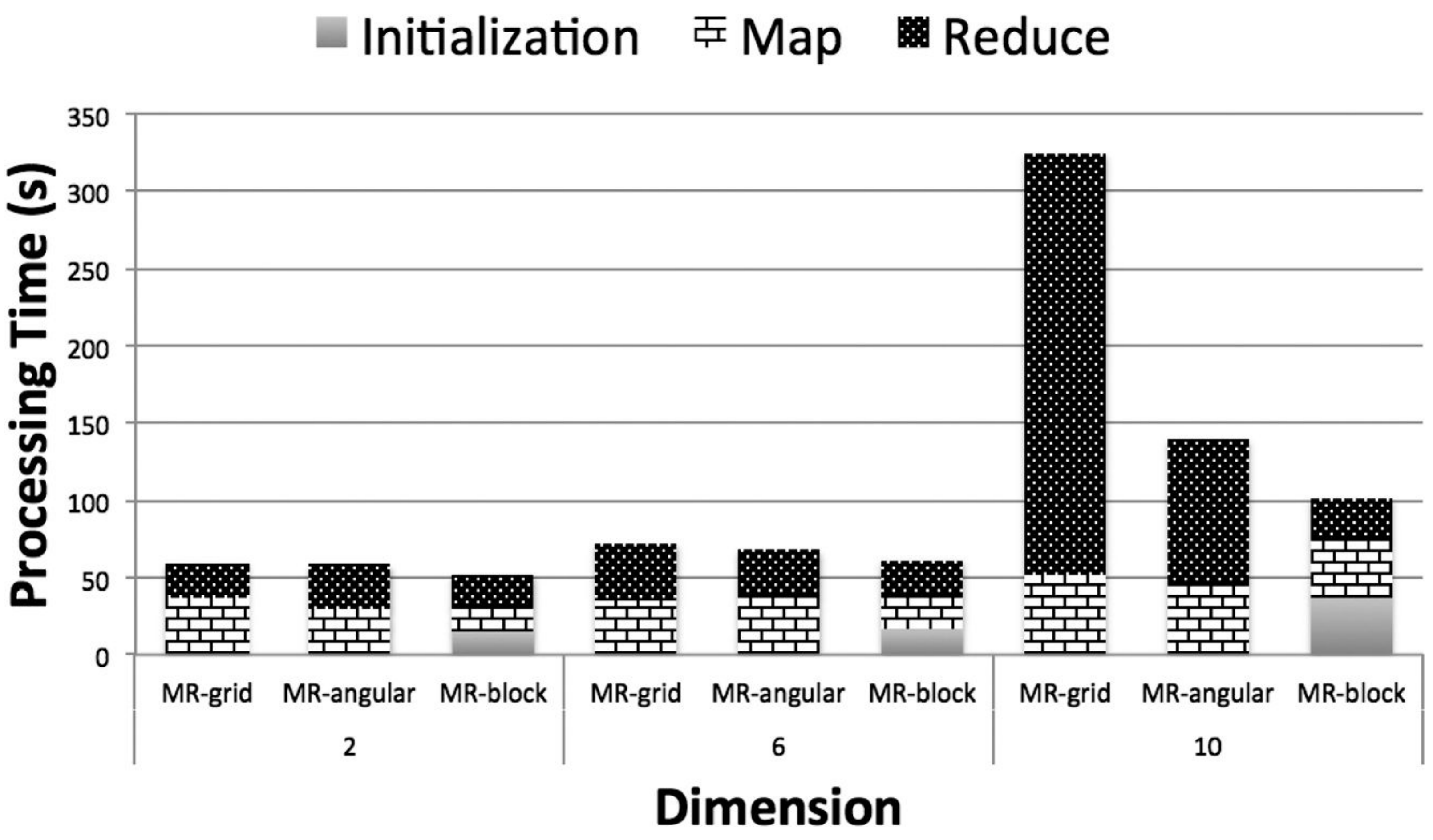
- The whole service space is first partitioned into **N disjoint** sections.
- Points within one partition are sent to one Map Task, and each **Map Task can process one or more partitions**.
- Map Task outputs the partition number as a key, and a list of **local skylines of that particular partition as a value**.
- The reduce phase, all the **local skylines are processed through a Reduce Task**

- Three MapReduce skyline methods for the composition of skyline-selected web services under QoMS assurance.
- Our experiments are carried out using both simulated data sets and real **Quality of Web Service (QWS)** data set-simulated data sets, including three categories: **random, anti-correlated, and correlated data**.



- The process of a skyline search includes two steps:
- **DIVIDE THE WHOLE SEARCH SPACE** into small spaces and search within **each small space**
- Send the skyline points of each small space to a node, which **computes the overall skyline points**.
- The **block-based elimination method reduces a few blocks** in step 1, and the other two methods (angle and grid) don't reduce any block.
- The **POINT REDUCTION RATE (PPR)** is measured by the aggregated local skyline points that are rolled over to step 2, over the total number of points.

- The **BLOCK REDUCTION RATE (BRR)** is defined as the blocks containing local skyline points that are rolled over to step 2, over the total number of blocks.
- We define a **SKYLINE RATIO (SR)** as the number of pairs of all of the blocks that need to be pairwise compared over the total number of pairs that have to be calculated in step 2.
- To evaluate the efficiency of various MapReduce skyline selection methods, we use the basic **metric of processing time, which consists of both reduce time and map t**


Initialization
Map
Reduce
Processing Time (s)
350
300
250
200
150
100
50
0
MR-grid
MR-angular
MR-block
MR-grid
MR-angular
MR-block
MR-grid
MR-angular
MR-block
2
6
10
Dimension

Cloud Security

Key areas in SC

- Cloud security - **Issues and challenges**
- The cloud **CIA security model**
- The cloud computing security architecture
- Cloud computing **model security threats**
- Privacy issues in the cloud and mitigation strategies
- Performance monitoring and management of cloud services
- Legal issues in cloud computing
- Risk management in cloud computing

Key areas in SC

- Business **continuity and disaster recovery**
- Threats in cloud
- **Cloud service level agreements** (SLA) practices
- Cloud vendors
- **Issues of quality of cloud services**
- **Migration of local** server into cloud
- Trust management in cloud

Cloud security and privacy

- Cloud security issues and challenges:
  - ✔Data breaches
  - ✔Data loss
  - ✔Network security
  - ✔Data locality
  - ✔Data access
  - ✔System vulnerabilities
  - ✔ Account hijacking
  - ✔ Malicious insider
  - ✔Advanced persistent threats among others

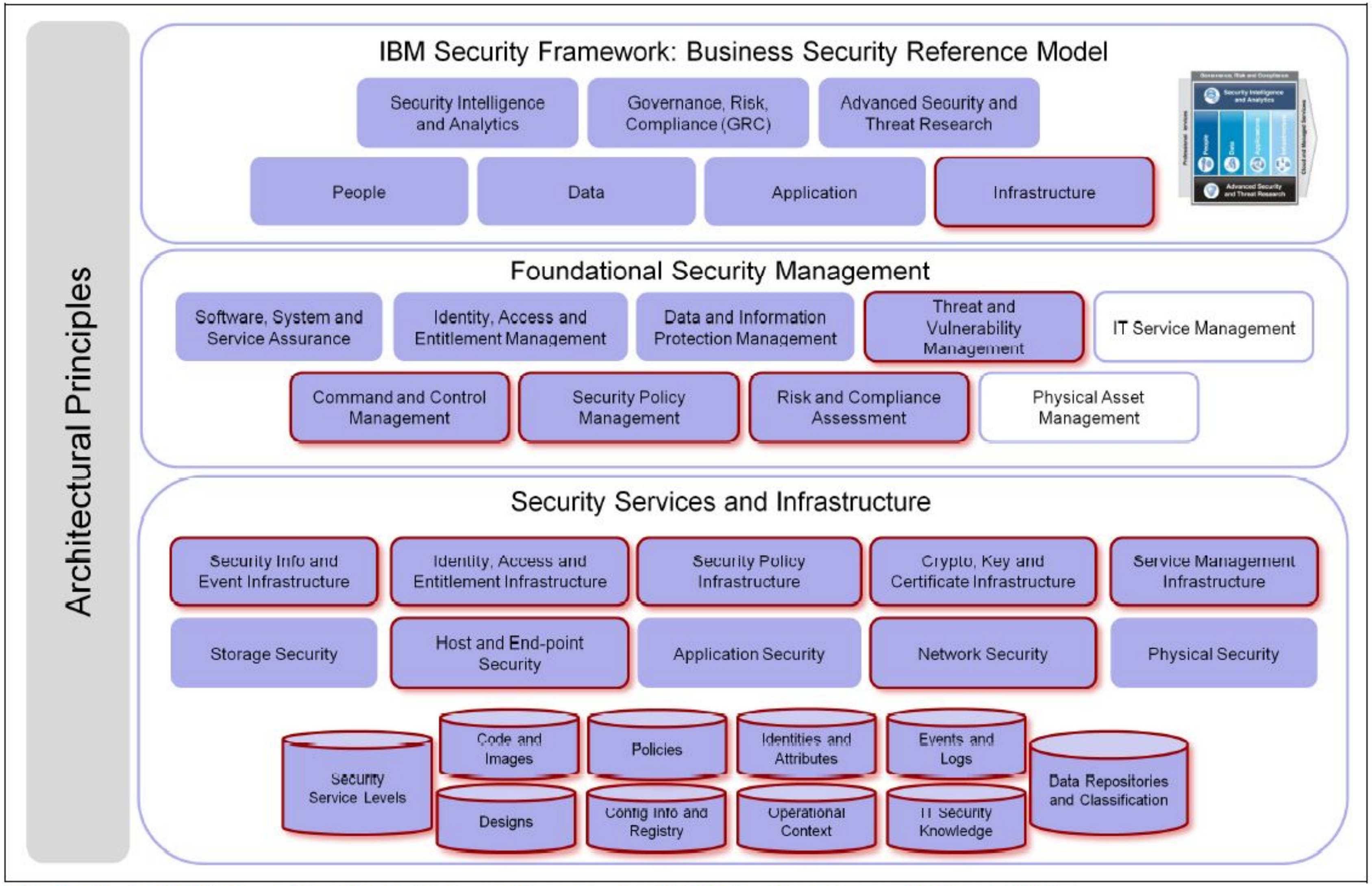


Figure 1-4 IBM Security Blueprint components for the Infrastructure solution pattern

Cloud security and privacy

- Data breaches:

✔ Unauthorized thirdparty maliciously gains access

✔ The attractive targets are the cloud data and file servers that hold massive volume of data.

✔ Impact of the attack depends on the type of the compromised data

✔ Financial data, personal data, trade secrets, health information and government critical data to intellectual properties of a person or an organization.

Cloud security and privacy

- Network security:

✔ Malicious-denial-of-service attack or unauthorized access leading to data leakage.

✔ Strong network traffic encryption

Cloud security and privacy

- Data locality:

✔ Cloud service consumers are not aware of **where their data is stored due to virtualization.**

✔ Legal implications of using, sharing and storing of data exist and vary from one country to another based on relevant laws and policies regarding intellectual property.

Cloud security and privacy

- Data access:

✔ **Gain authorized access** to their subscribed services anywhere and at any time.

✔ Strong authentication before access to the massive cloud resources is required

Cloud security and privacy

- System vulnerabilities:

✔ Program bugs in the operating system

✔ Prone to a denial-of-service attack, advanced persistent threat and malicious user's attack.

✔ When participating in a network, the attacker leverages on these vulnerabilities to distribute different kinds of malware.

Cloud security and privacy

- Account hijacking:

✔ **Stealing and using** of the account details of a legitimate user

✔ The credential hijackers could easily compromise the availability, integrity and confidentiality of the cloud services.

✔ A **multifactor authentication mechanism** required

Cloud security and privacy

- Malicious insiders:

✔ System administrators, former employees, business partners or a third-party contractors

Cloud security and privacy

- The advanced persistent threats:

✔ This is a stealthy computer network attack in which multiple assault code are injected into a vulnerable system at entry points

✔ Remain undetected over a long period of time

Cloud security and privacy

- Permanent data loss:

✔ Natural disaster

✔ Total hardware failure

✔ Unintentional cancellation by clients or support staff at the service providers' end

✔ Adequate mitigation plans against data loss regardless of any form it takes is required

✔ Regular backup routines to remote locales

Cloud security and privacy

- Shared technology, shared dangers:

✔ Vulnerability and misconfigured components or weak isolation properties

✔ Proper data management and client implementation is required

Cloud security and privacy

- Compromised credentials and broken authentication:

✔ **Email spam, DoS assaults** to gain unauthorized access to critical data, control and management functionalities of the cloud services.

✔ Attackers **can inject malicious software to attack the cloud services**

✔ Modify data and service management/control parameters or sniff data in transit.

Cloud security and privacy

- Hacked interfaces and Application Programming Interface (API):

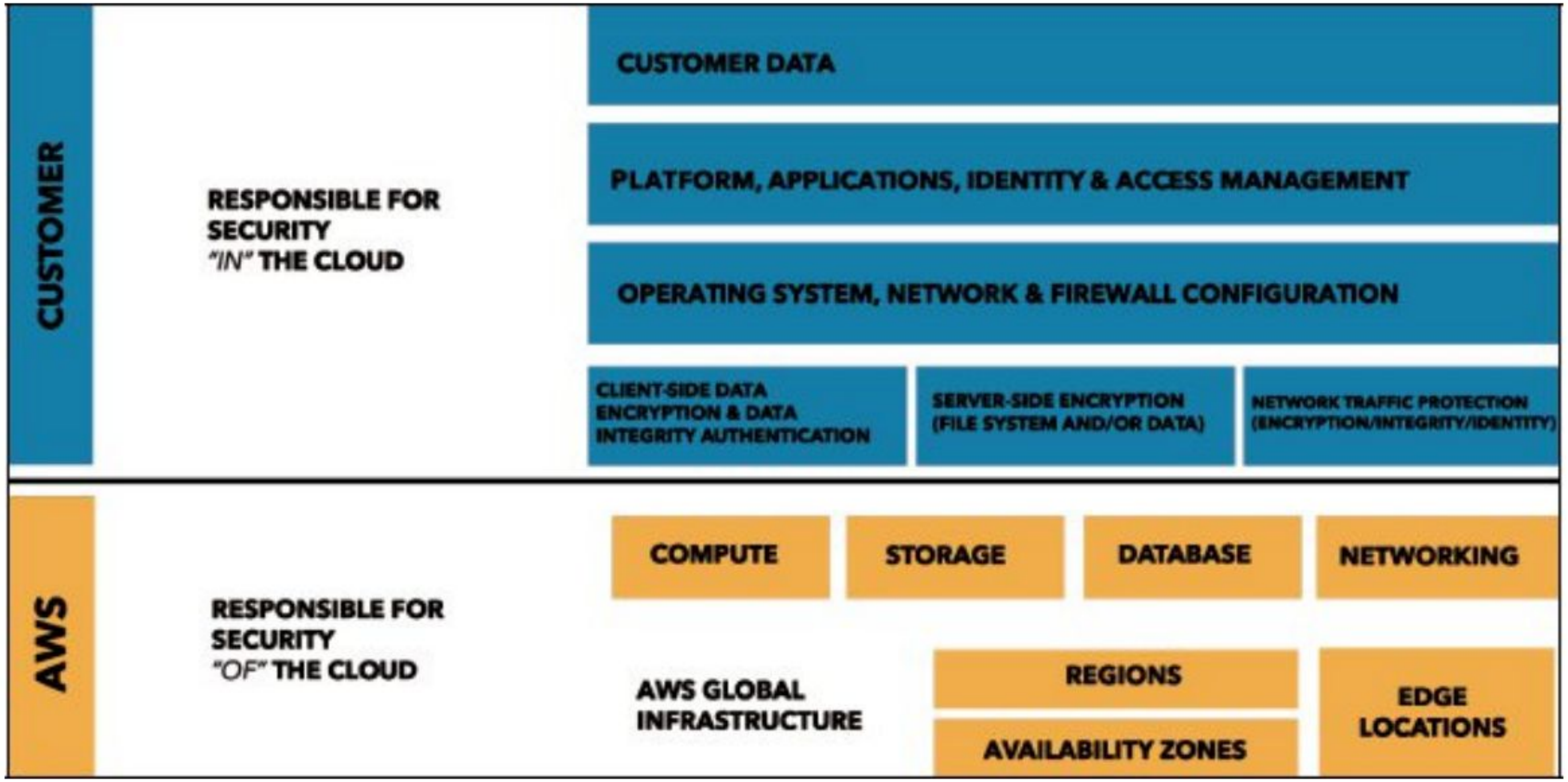
✔ APIs and user interfaces are the fundamental backbones of cloud system connections

✔ **Cloud APIs' Internet Protocol (IP) addresses** expose the association between clients and the cloud

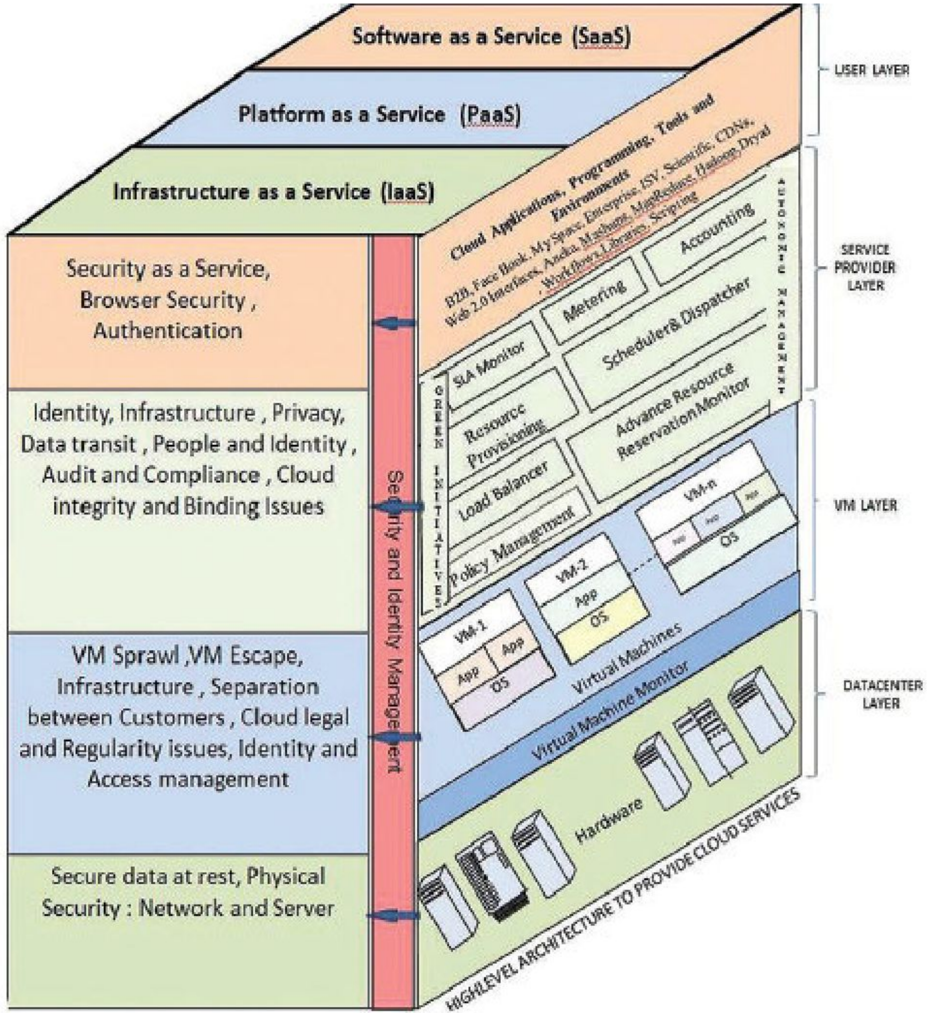
✔ Securing APIs from corruption or human mistakes is pertinent to cloud security.

✔ A special security requirement for the APIs needs to be designed to allow for access via encrypted keys

Shared responsibility model



A prototype security architecture of cloud computing



- Facebook, B2B, CDN, Aneka, Enterprise, Mashups, Web 2.0 Interfaces, Scientific and MapReduce
- **Browser Security, Authentication and Security-as-a-Service**.
- SLA Monitor, Scheduler & Dispatcher, Metering, Load Balancer, Accounting, Policy Management and Advance Resource Reservation Monitor.
- Data transmission, privacy, People and Identity, Infrastructure management, Audit and Compliance, Cloud integrity and Biding Issues
- VM Escape, VM Sprawl, Infrastructure, Identity and Access management

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Key concepts in Security

- **Browser security**
- **AUTHENTICATION**
- **Privacy**
- **Securing data in transmission**
- **User identity**
- **Audit and compliance**

- **BROWSER SECURITY : Transport layer security (TLS), same origin policy (SOP)**
- **AUTHENTICATION:** Trusted Platform Module (TPM), **Identity and access management (IAM)** involves the **Authentication, Authorization and Auditing (AAA**), virtual private networks (VPNs), intrusion prevention systems (IPSs), intrusion detection systems (IDSs)
- **PRIVACY :**identity management, data protection, secure operations
- **SECURING DATA IN TRANSMISSION :**Encryption techniques, secure socket layer protocols
- **USER IDENTITY :** single sign-on logons
- **AUDIT AND COMPLIANCE :** cloud service providers (CSPs)

Security issues in virtualization

- **Virtual machine escape :** VM escape is a security situation which occurs when a **total system failure** is experienced due to **improperly configured virtual machines.**
- The other potential risk associated with virtualization is **Rogue Hypervisors**

VM security recommendations (best practices security techniques)

- Using Encrypted Communications is to provide secure communications via cryptography techniques like **Secure Shell (SSH), Transport Layer Security (TLS), Secure HTTP (HTTPS) and encrypted Virtual Private Networks (VPNs**)
- *Disabling Background Tasks*
- *Updating and Patching*
- *Implementing File Integrity Checks*
- *Securing VM Remote Access*
- *Separation between users*

Cloud legal issues

- ***Datacenter (infrastructure) security issues***
- ***Securing data-storage***
- ***Network and server***
- ***Server-side protection***
- ***Securing the hybrid cloud***

Privacy issues in the cloud and mitigation strategies

- **P**ersonally **I**dentifiable **I**nformation (PII)
- Sensitive information
- Usage data
- Unique device identities

Category	Pattern title	Solutions in AWS	Solutions in Azure
Compliance and Regulatory	**Data Citizenship**	Use AWS location tags to designate the location for data processing	Azure information protection and location tag. Azure frontdoor service
	Cryptographic Erasure	Use AWS KMS	Use Azure Key Vault
	Shared Responsibility Model	AWS provides different services to ensure protection of data and system. It is upto client to use it or not. However, AWS is responsible for only the vailability and basic security of cloud platform.	Azure provides different security tools to ensure protection of data and system. It is upto client to use it or not. However, Azure is responsible for only the vailability and basic security of cloud platform
	Compliant Data Transfer	AWS locaton tags	Azure location tag
	Data Retention	The data retention policies can be defined and executed by AWS. For example Lambda	Azure provides option to define data retention policy in Database system
	Data Lifecycle	AWS data lifecycle manager	Azure blob storage lifecycle
	Intentional Data Remanence	database (e.g. DynamoDB)	database (e.g. Azure backup)
Identification, Authentication and Authorisation	**Multi-Factor Authentication**	AWS Cognito	Azure active directory : multi-factor
	Federation (Single Sign-On)	AWS SSO (Single Sign-On)	Azure AD Seamless Single Sign-On
	Access Token	AWS security token service	Azure active directory : Token service
	Mutual Authentication	Use AWS TLS/SSL certificate, Certificate feature of API Gateway (AWS client VPN)	Azure App service
	Secure User Onboarding	AWS customer on boarding	Azure security center
	Identity and Access Manager	AWS IAM and Cognito	Azure IAM
	Per-request Authentication	AWS Signing and Authenticating REST Requests	Azure API management & REST API authentication
	Access Control Clearance	AWS cloud watch and AWS Cognito/IAM	Azure access control service
Secure Development, Operation and Administration	**Bastion Server**	AWS bastion host	Azure Bastion host
	Automated Threat Detection	AWS GuardDuty	Azure advanced threat protection
	Durable Availability	AWS cloud watch, AWS WAF	Azure web access firewall & firewall application gateway
	Economic Durability	AWS cloud watch	Azure Monitor
	Vulnerability Management	AWS vulnerability scanning	Vulnerability scan in Azure security center
Privacy and Confidentiality	**End-to-End Security**	AWS KMS, Certificate manager	Azure Key Vault
	Computation on Encrypted Data	N/A	N/A
	Data Anonymisation	Algorithms can be defined and ran by AWS module (e.g. lambda)	Azure provides Dynamic Data Masking on SQL database
	Processing Purpose Control	N/A	N/A
Secure Architecture	**Virtual Network**	AWS Virtual Private Cloud	Azure Virtual Network
	Web Application Firewall	AWS WAF	Azure application firewall gateway
	Secure Element	AWS IoT Device Management	Azure IoT Hub & IoT Suit
	Secure Cold Storage	AWS Glacier	Azure Coldblob storage
	Certificate and Key Manager	AWS Certificate and Key manager (AWS KMS)	Azure Key Vault
	Hardware Security Module	AWS CloudHSM	Azure Dedicated HSM
	Secure Auditing	AWS Auditing Security Checklist	Azure Monitor, Stream, Network Watcher

Security Services

- **Amazon CloudWatch**
- Monitor your Amazon Web Services (AWS) resources and the applications you run on AWS in real time.

✔ **Collect and track metrics**, which are variables you can measure for your resources and applications.

✔ The CloudWatch home page automatically **displays metrics about every AWS service you use.**

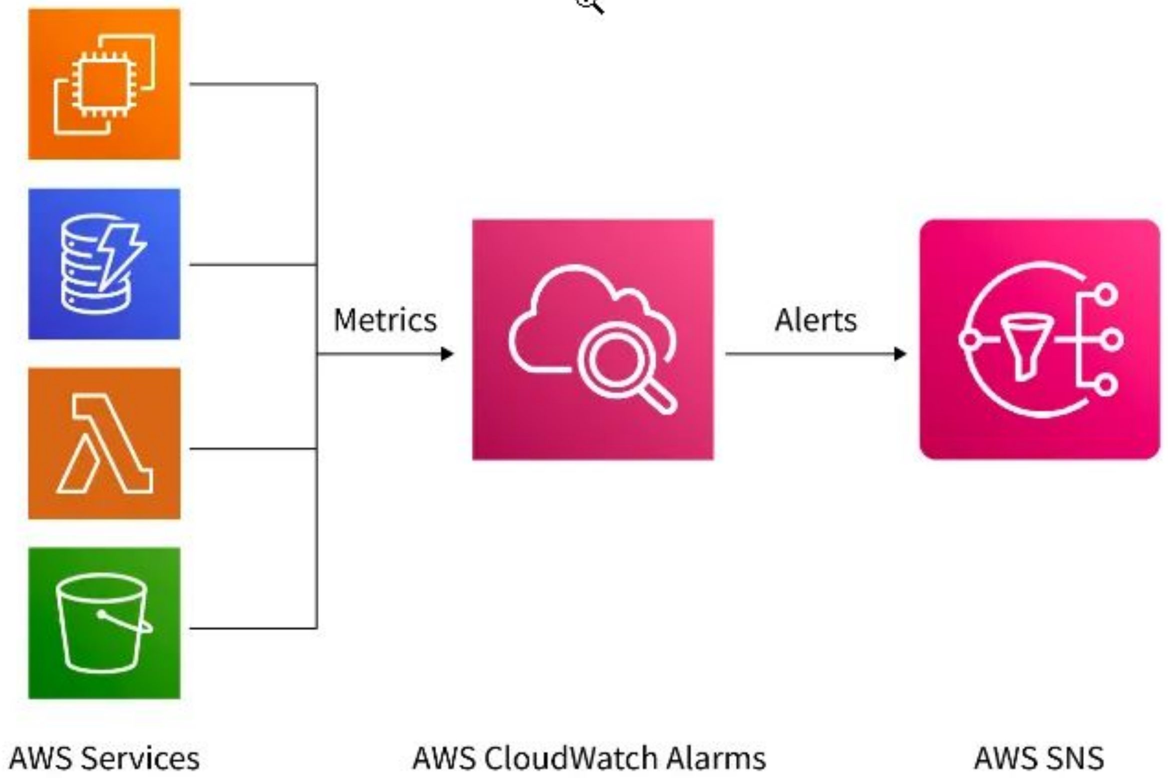
✔ You can additionally create **custom dashboards** to display metrics about your custom applications, and display custom collections of metrics that you choose.

✔ You can **create alarms** that watch metrics and send **notifications or automatically** make changes to the resources you are monitoring when a threshold is breached.

AMAZON CLOUDWATCH

- Private Cloud Monitoring Systems
- Cloud Management System
- Runtime Model for Cloud Monitoring
- Flexible Automated Cloud Monitoring Slices

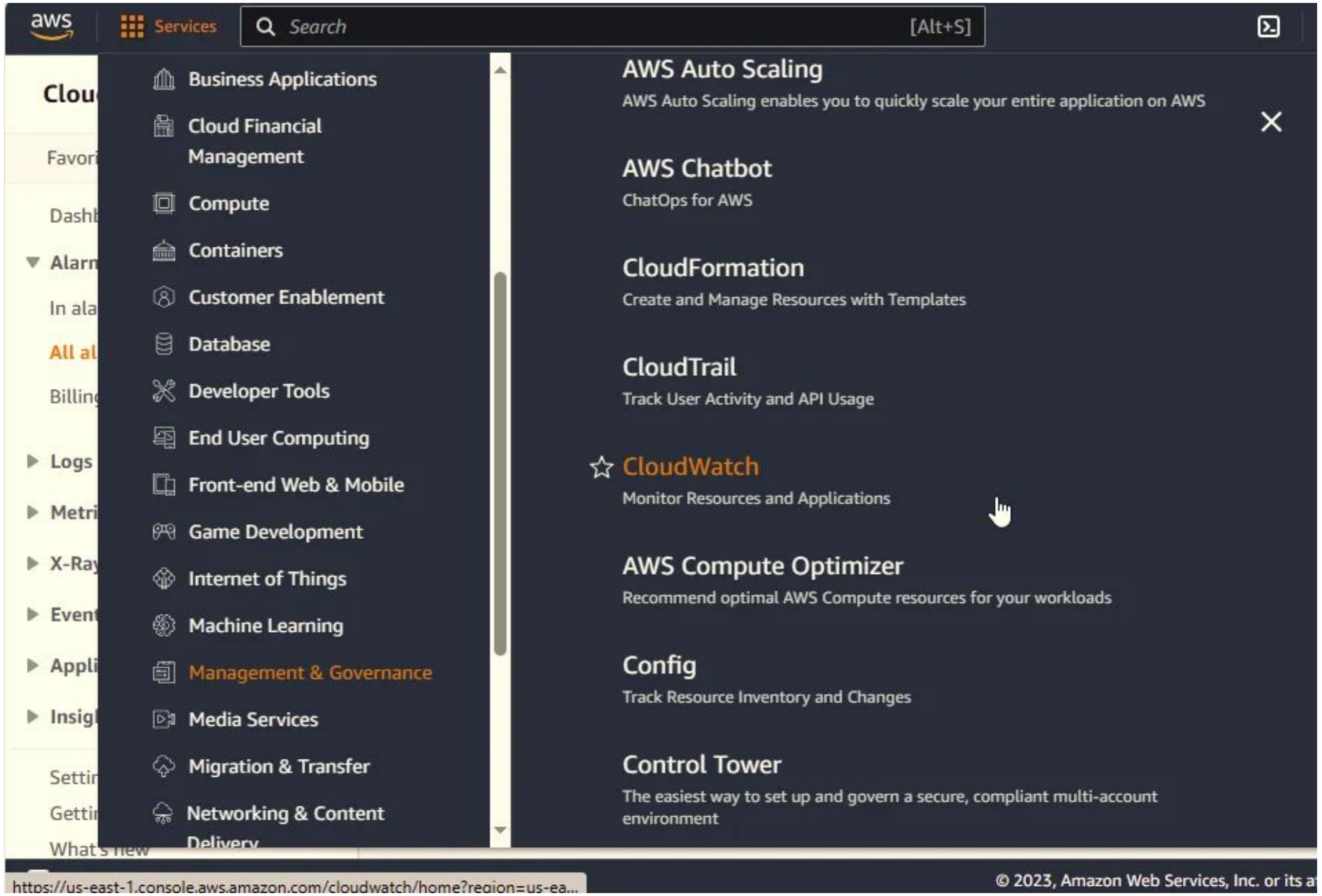
Amazon CloudWatch



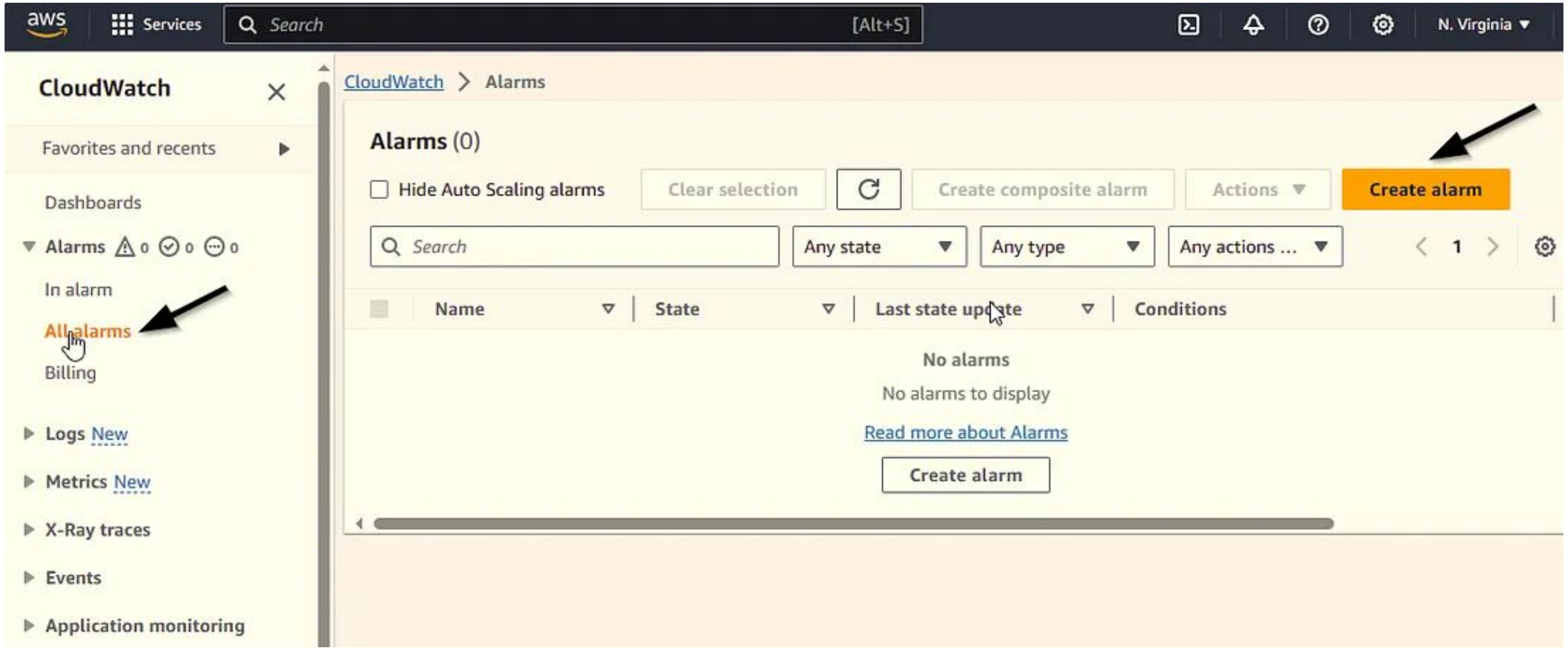
- **Nagios:** is a free and open-source tool to monitor computer systems, networks, and infrastructure. Nagios also offers alerting services for servers, switches, and user applications.
- **New relic**: is an application performance monitoring (APM) solution that uses agents placed in a VM, say in a Cloud or local server, to monitor how that application is behaving

CloudWatch Alarms

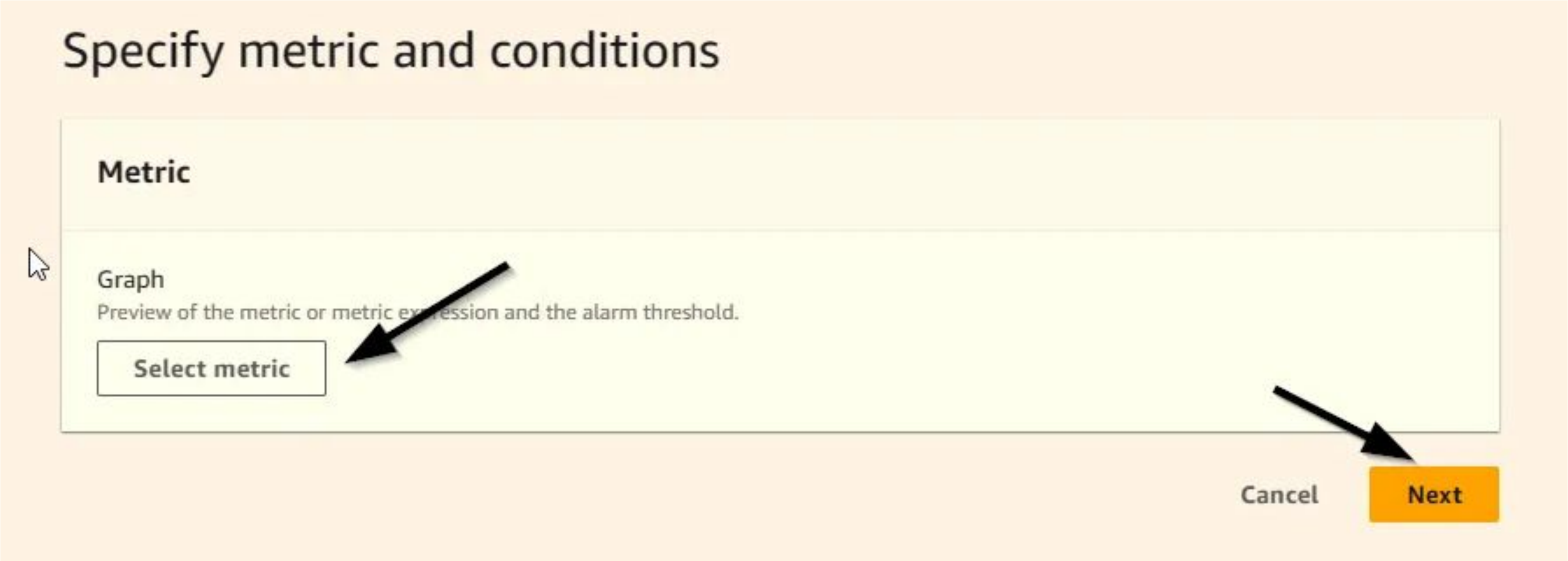
- **Access the AWS Management Console:**
- Open your web browser and navigate to AWS Management Console.
- **Log in to your AWS account**.
- **Go to CloudWatch:**
- In the AWS Management Console, locate and click on "Services" in the top left corner.
- Under "**Management & Governance**," select "CloudWatch."


aws
Services
Search
[Alt+S]
Business Applications
Cloud Financial Management
Compute
Containers
Customer Enablement
Database
Developer Tools
End User Computing
Front-end Web & Mobile
Game Development
Internet of Things
Machine Learning
Management & Governance
Media Services
Migration & Transfer
Networking & Content Delivery
AWS Auto Scaling
AWS Auto Scaling enables you to quickly scale your entire application on AWS
AWS Chatbot
ChatOps for AWS
CloudFormation
Create and Manage Resources with Templates
CloudTrail
Track User Activity and API Usage
CloudWatch
Monitor Resources and Applications
AWS Compute Optimizer
Recommend optimal AWS Compute resources for your workloads
Config
Track Resource Inventory and Changes
Control Tower
The easiest way to set up and govern a secure, compliant multi-account environment
https://us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-ea...
© 2023, Amazon Web Services, Inc. or its a

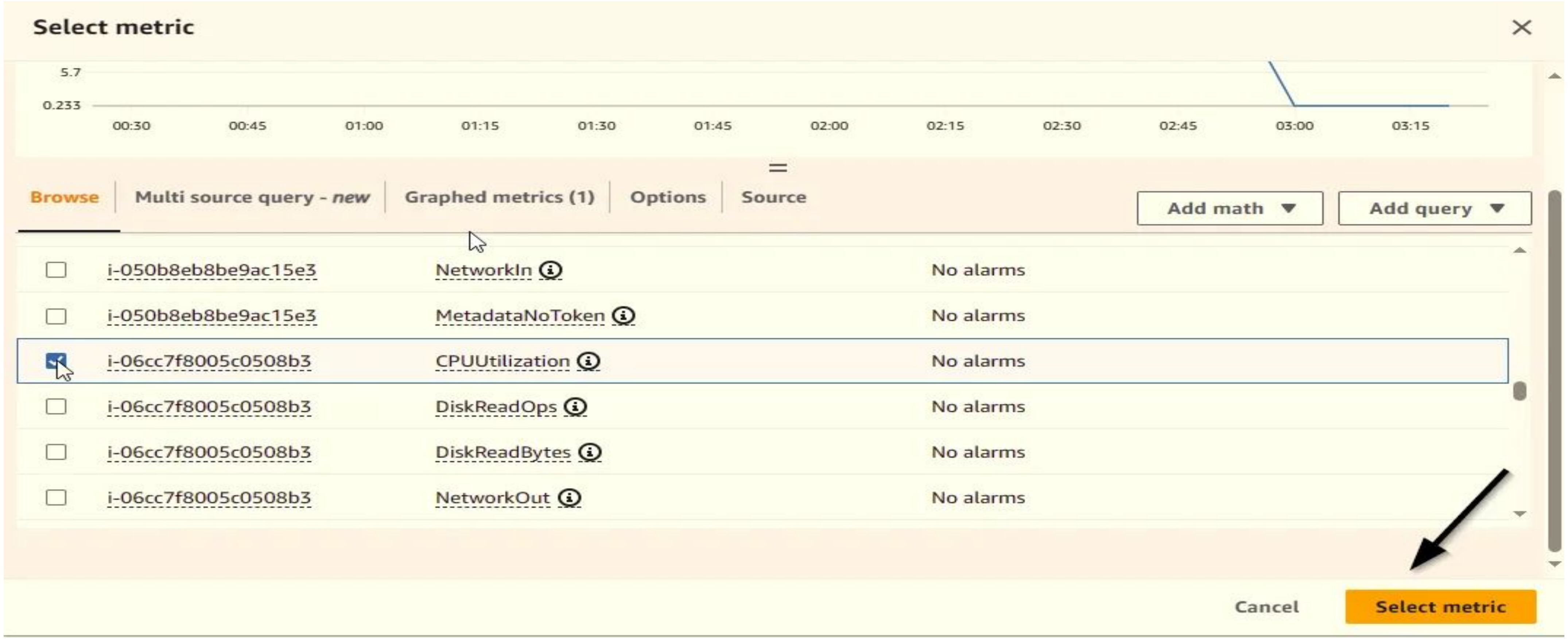
- Navigate to Alarms:
- In the CloudWatch dashboard, find the left-hand navigation pane.
- Click on “Alarms” under the “All arms” section.
- On the “All alarms” page, click the “Create Alarm” button.



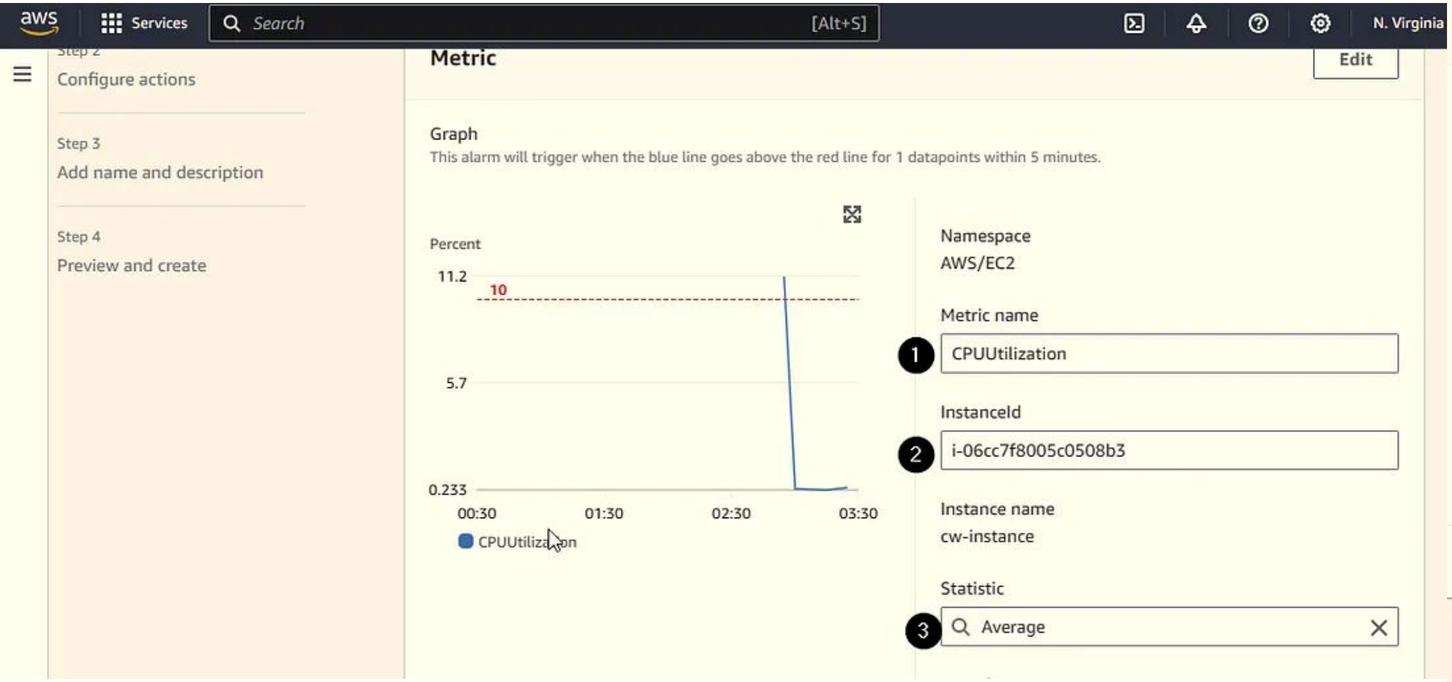
- Select Metric:
- Choose the metric you want to monitor by selecting a data source (e.g., EC2, RDS) or a custom metric

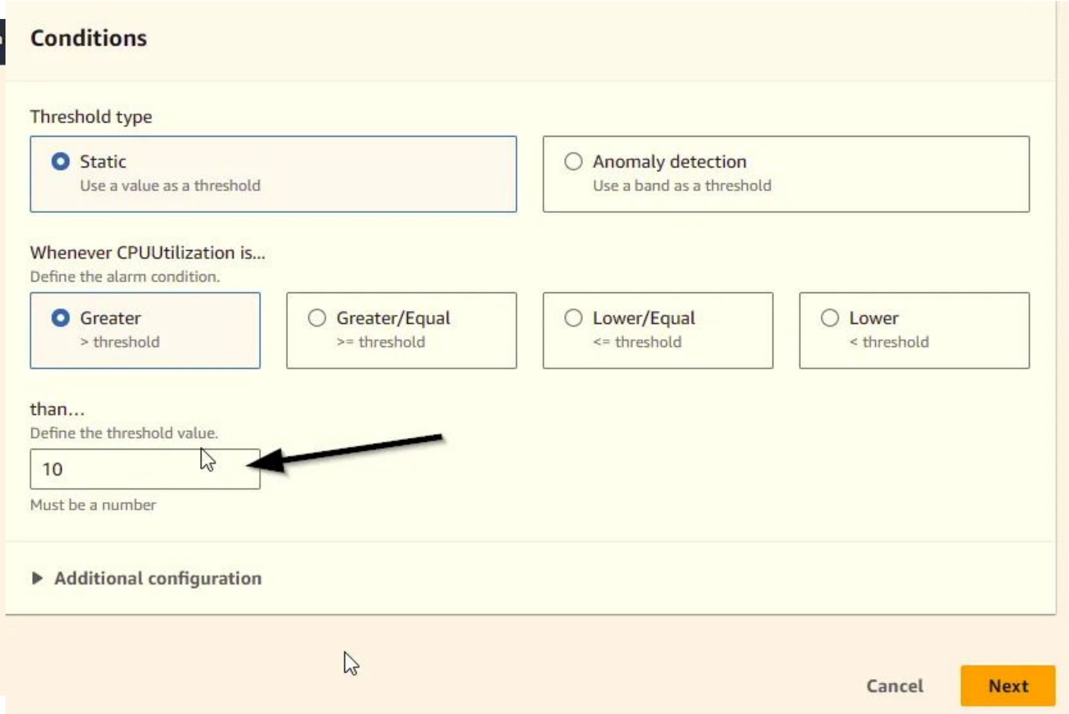


- Here we can select our ec2 preferred conditions
- Here we selecting “cpu utilization”
- And click “select metric”

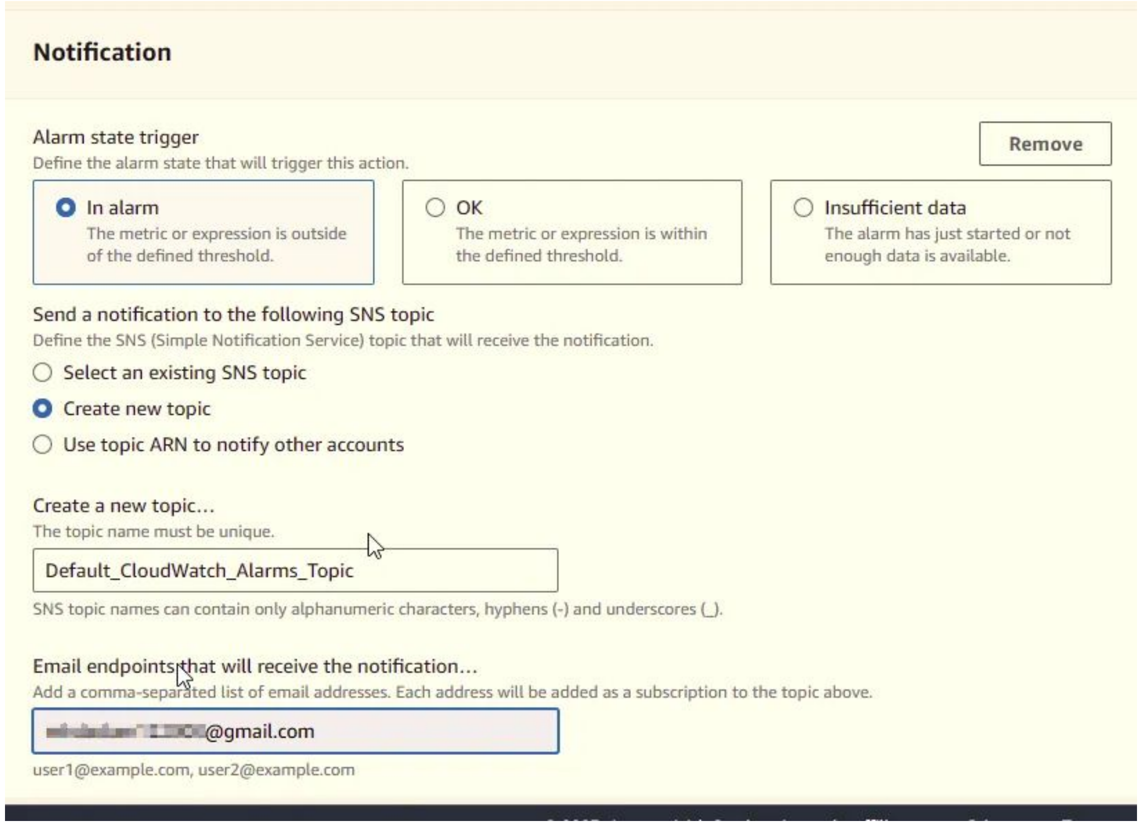


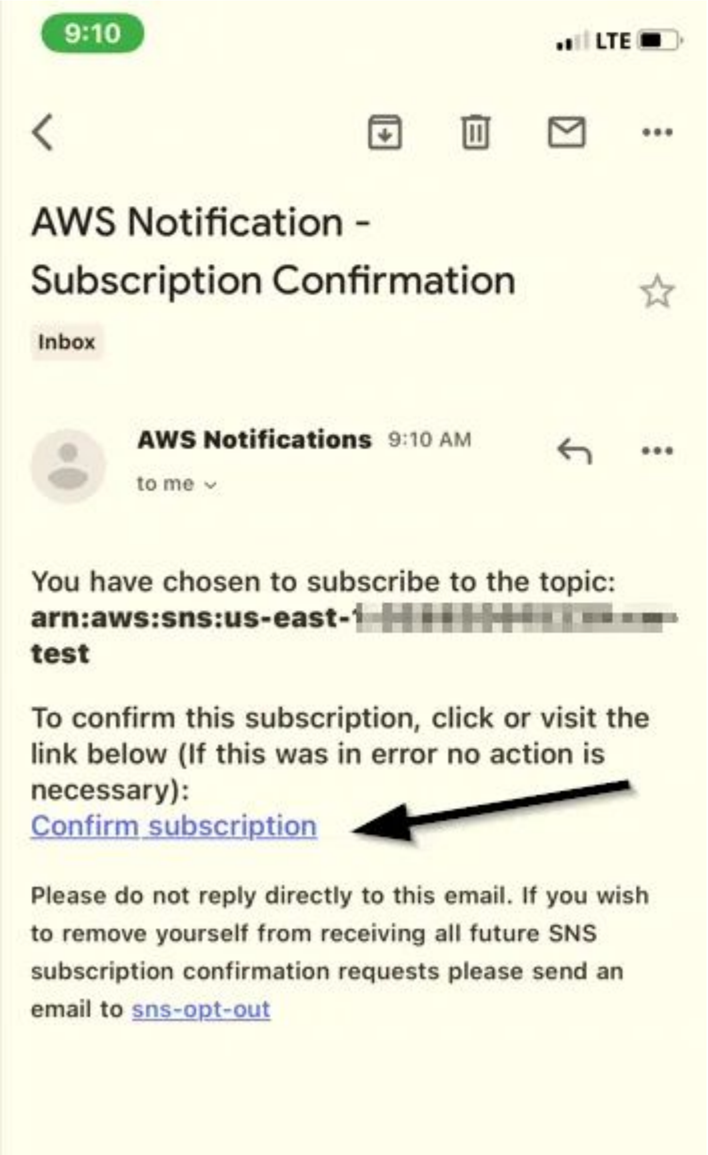
- Define Conditions:
- Set conditions for the alarm. Specify thresholds, comparison operators, and evaluation periods.
- we can set conditions what is our requirments

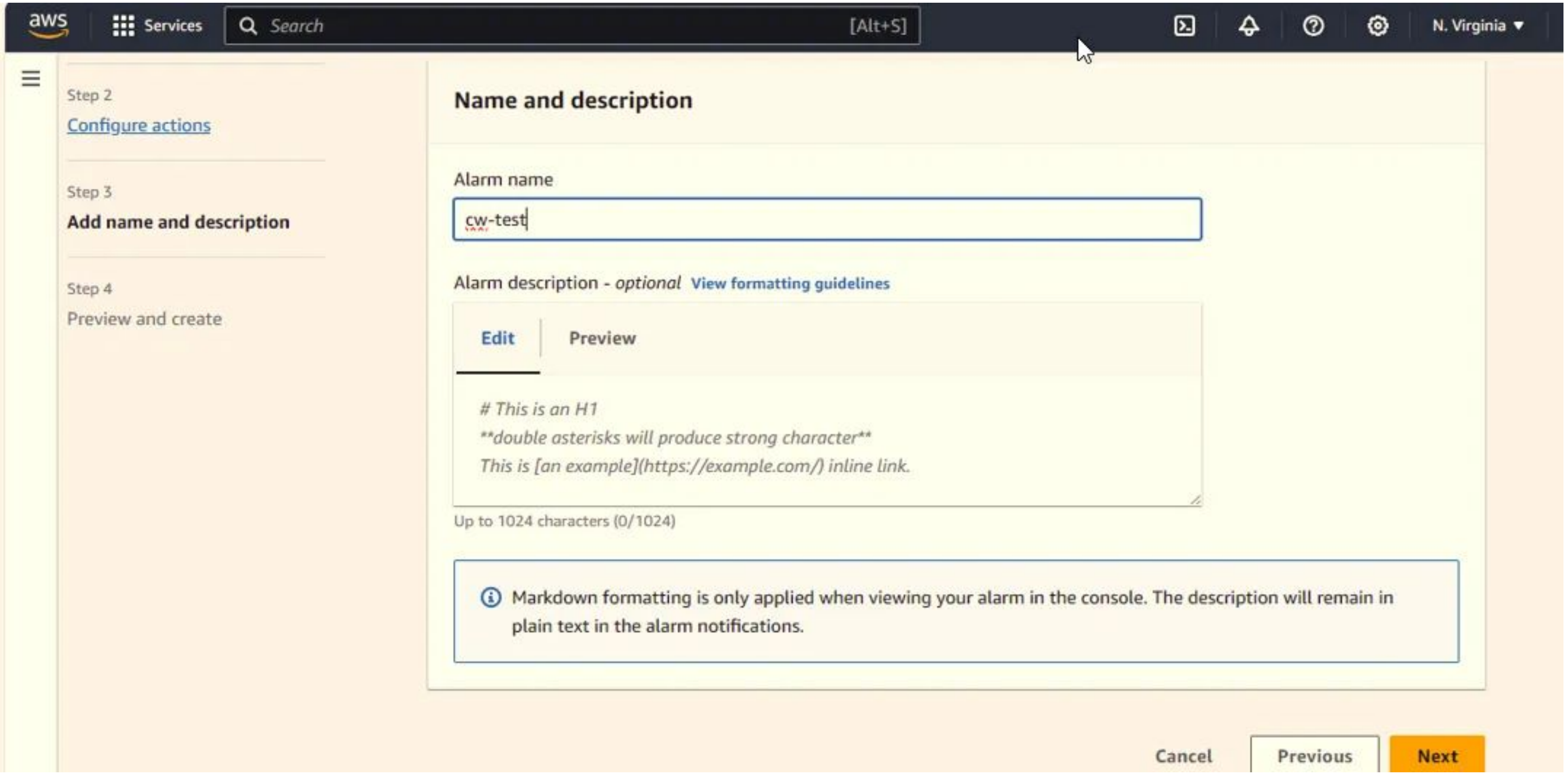


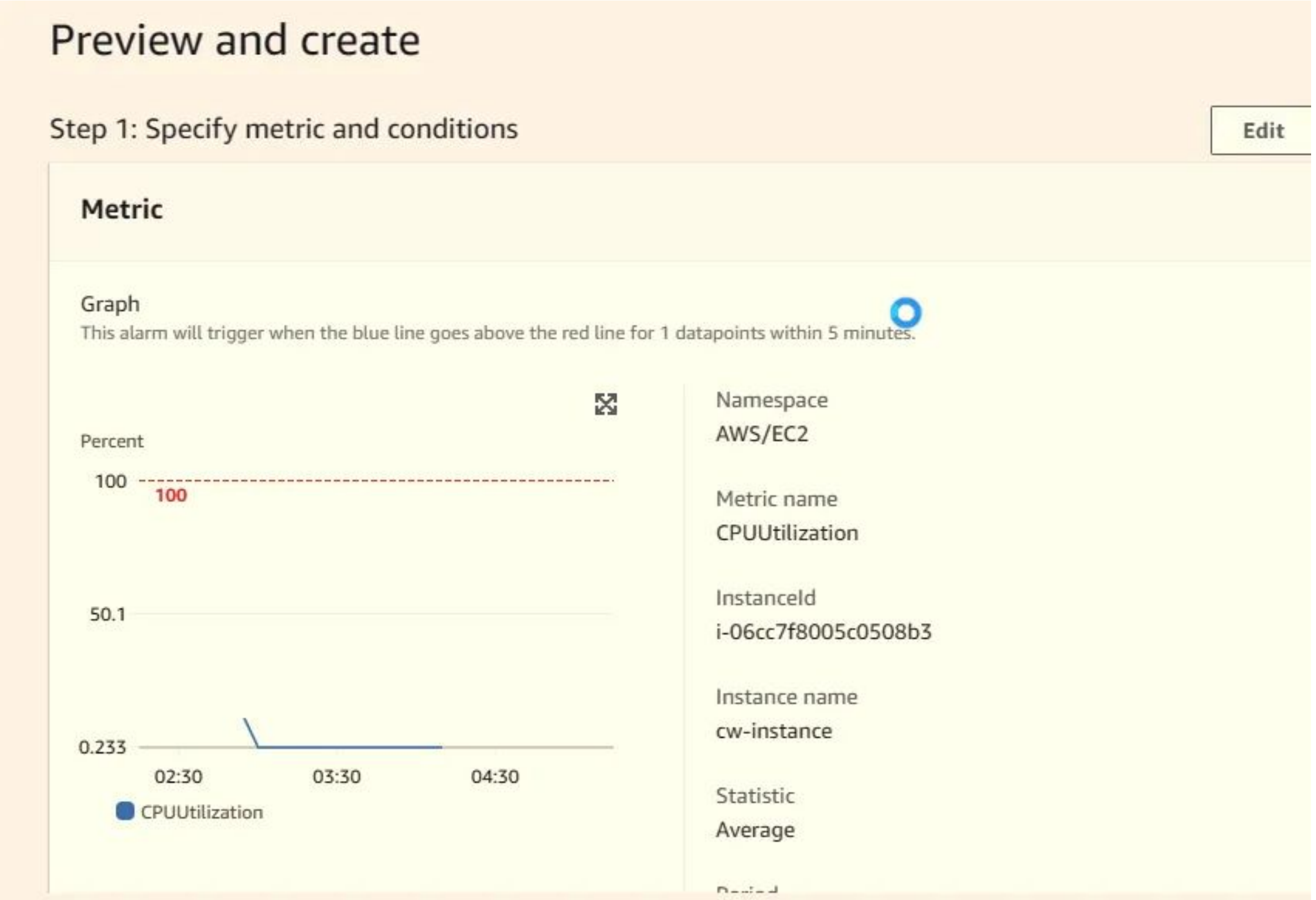


- Set Actions on configure SNS topic
- Choose “Create a new SNS topic.”
- Enter a name for the new SNS topic.
- Enter E-mail as a end point
- click create topic




9:10
LTE
AWS Notification - Subscription Confirmation
Inbox
AWS Notifications 9:10 AM
to me
You have chosen to subscribe to the topic:
arn:aws:sns:us-east-
test
To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription
Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to sns-opt-out


aws
Services
Search
[Alt+S]
N. Virginia
Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create
Name and description
Alarm name
cw-test
Alarm description - optional
View formatting guidelines
Edit
Preview
This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.
Up to 1024 characters (0/1024)
Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.
Cancel
Previous
Next


Preview and create
Step 1: Specify metric and conditions
Edit
Metric
Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.
Percent
100
100
50.1
0.233
02:30
03:30
04:30
CPUUtilization
Namespace
AWS/EC2
Metric name
CPUUtilization
InstanceId
i-06cc7f8005c0508b3
Instance name
cw-instance
Statistic
Average

Preview and create

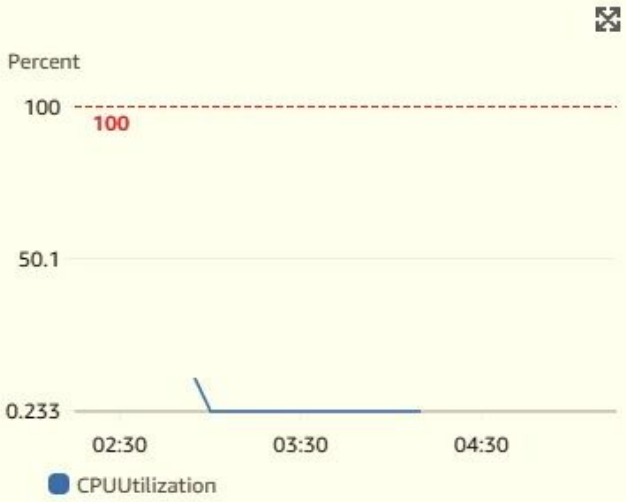
Step 1: Specify metric and conditions

Edit

Metric

Graph

This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.



Namespace
AWS/EC2

Metric name
CPUUtilization

InstanceId
i-06cc7f8005c0508b3

Instance name
cw-instance

Statistic
Average

Period

Conditions

Threshold type
Static

Whenever **CPUUtilization** is
Greater (>)

than...
100

▶ **Additional configuration**


Step 2: Configure actions
Edit
Actions
Notification
When In alarm, send a notification to "cw-test"
Step 3: Add name and description
Edit
Name and description
Name
cw-2
Description
-
Cancel
Previous
Create alarm

- Private Cloud Monitoring Systems (PCMONS): PCMONS is an **open source monitoring solution** developed for private clouds.
- It uses an integration layer to **grant homogeneous access to managers** (administrators, services providers, cloud service brokers, and so on) and users (cloud service consumers) that manipulate resources in a cloud.
- **Cloud Management System (CMS):** CMS leverages on RESTful Web Services to provide monitoring services.
- The REST serves as a technology for **designing monitoring elements**

- **Runtime Model for Cloud Monitoring (RMCM):** RMCM is designed to monitor resources through abstract models which allow **possible homogeneous handling of heterogeneous resources**.
- **Flexible Automated Cloud Monitoring Slices Flex-ACMS** is a composite and comprehensive cloud monitoring solution resulting from a rich integrated set of monitoring solutions

Security Services

- Amazon CloudTrail:

✔ Enable operational and **risk auditing, governance, and compliance of your AWS account.**

✔ Actions taken by a user, role, or an AWS service are recorded as events in CloudTrail.

✔ Events include actions taken in the ***AWS Management Console, AWS Command Line Interface, and AWS SDKs and APIs.***

✔ CloudTrail is enabled on your AWS account when you create it.

✔ When activity occurs in your AWS account, that activity is recorded in a **CLOUDTRAIL EVENT**.

✔ You can easily view recent events in the **CloudTrail console by going to Event history**

Security Services

- Azure security center:

✔ Security health monitoring for both **cloud and on-premises workloads**

✔ Security threat blocking through **access and app controls**

✔ **Adjustable security policies** for maintaining regulatory and standards compliance

✔ **Security vulnerability discovery tools** and patches and advanced threat detection through security alerts and analytics

✔ Service to obtain an **analysis of all data** and compute resources that you have deployed on your account.

✔ Scan each of your **data containers to determine their encryption and access status**.

Security Services

- Azure Threat Analytics:

✔ **Detect abnormal behavior**, malicious attacks, and other security issues in your environment.

✔ Enable application whitelisting, by which you declare which applications are allowed to access your resources.

Role-based Access Control

- ✔ Azure - **Role-based access control (RBAC)** :
  - *Allows you to control how different parties use resources under your account.*
  - *Each new user must have a role*
  - *General role -“Contributor”*
  - *Specific role - “Data Lake Analytics Developer” or “SQL DB Contributor.”*
- ✔ Amazon - **Identity and Access Management (IAM) service**
  - You can create a variety of different IAM roles and assign those roles to users, applications, and services.
  - An IAM role defines who the user is and what that user is authorized to do.
  - Monitor the use by holders of different roles.

Role-based Access Control

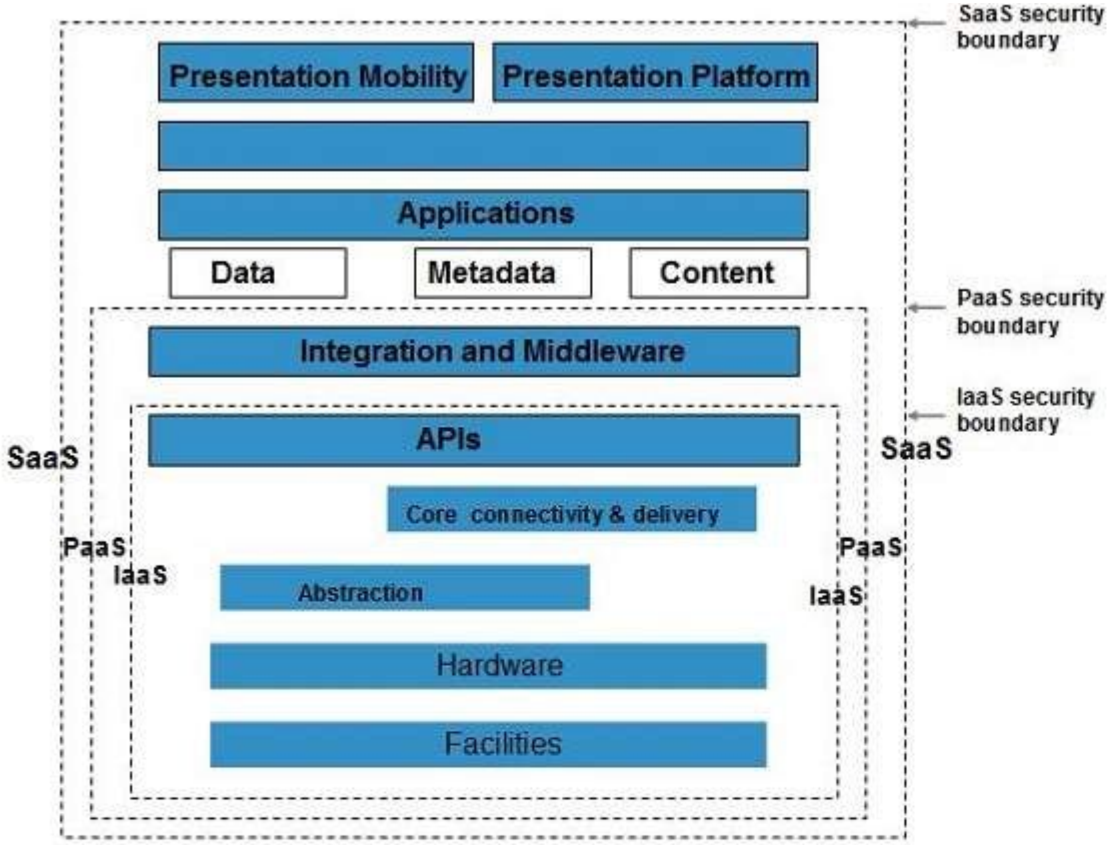
Sharing Secrets among Containers in a Cluster

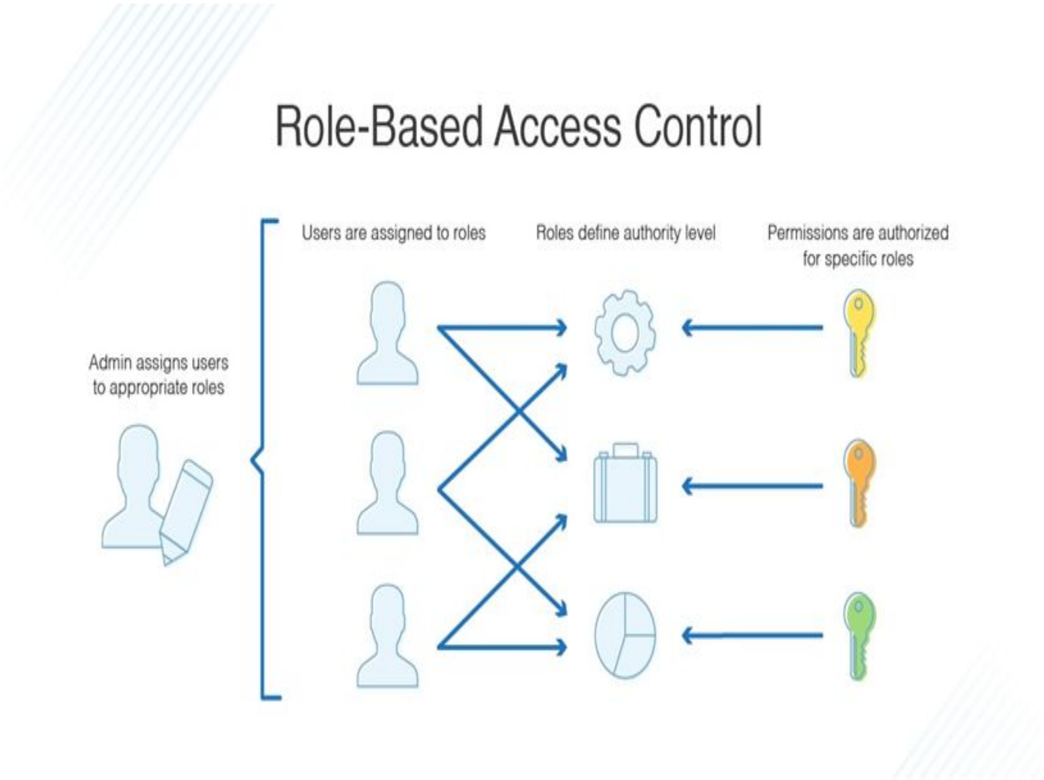
- ✔ Container instances access various secrets, such as the **API keys, identities, and passwords of the services that they invoke.**
- ✔ While you can pass the keys to individual instances from the command line, that approach does not work for instances that are managed dynamically.
- ✔ Leaving the keys in the **container Dockerfile is not secure**, because they are then embedded in the Docker image.
- ✔ Amazon's IAM role system solves this problem for the Amazon container service, ECS.
- ✔ RBAC solves the similar problem for Azure.
- ✔ Docker Swarm services to manage a collection of containers

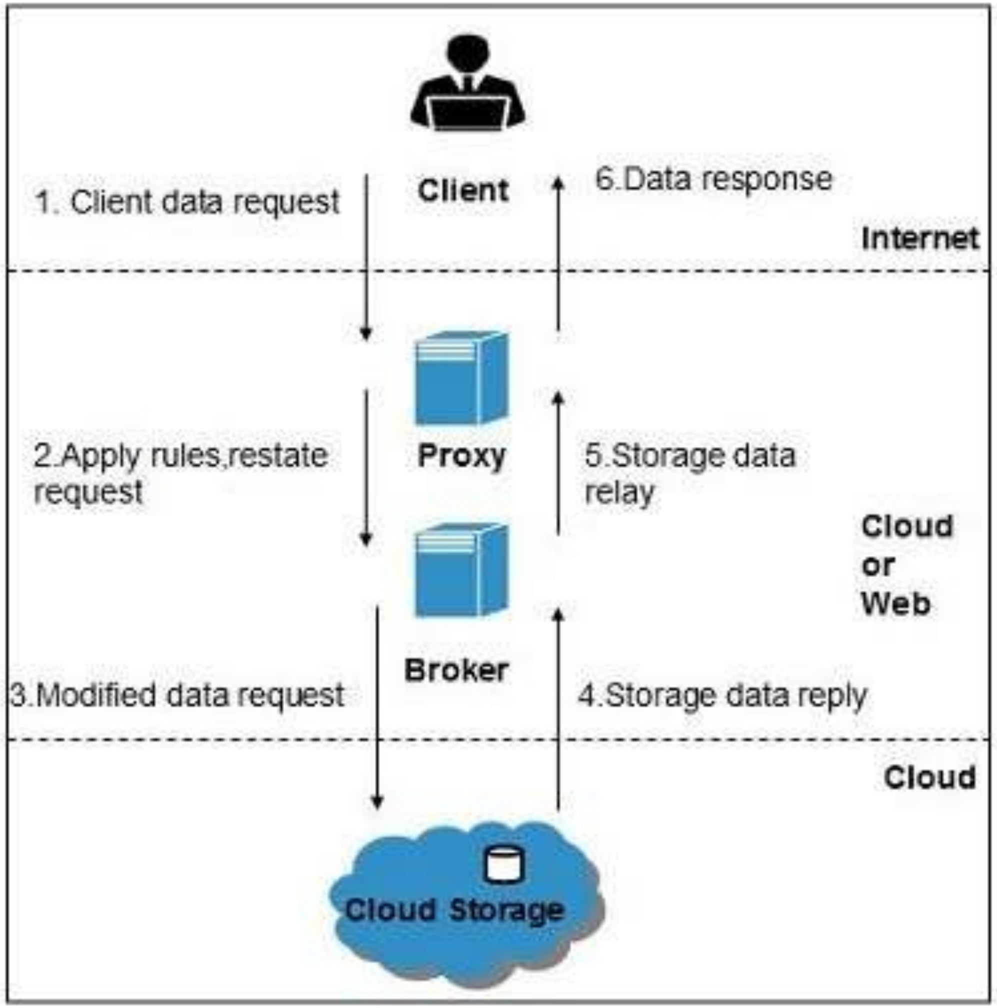
Roles	Responsibilities
Role assignment	• Each transaction or operation can only be carried out if the user has assumed the **APPROPRIATE ROLE**. • An operation is defined as any action taken with respect to a system or network object that is **protected by RBAC**. • Roles may be assigned by a separate party or selected by the user attempting to perform the action.
Role authorization	• The purpose of role authorization is to **ensure that users can only assume a role** for which they have been given the appropriate authorization. • When a user assumes a role, they must do so with authorization from an administrator
Transaction authorization	An operation can only be completed if the user attempting to complete the transaction possesses the appropriate role.

Secure Data in the Cloud

- **Secure Data in Transit**
- **Control Who Can Access Your Data**
- **Encrypt Your Data**
- **Complexities of Sensitive Data**


SaaS security boundary
Presentation Mobility
Presentation Platform
Applications
Data
Metadata
Content
PaaS security boundary
Integration and Middleware
IaaS security boundary
APIs
SaaS
SaaS
Core connectivity & delivery
PaaS
PaaS
IaaS
IaaS
Abstraction
Hardware
Facilities


Role-Based Access Control
Users are assigned to roles
Roles define authority level
Permissions are authorized for specific roles
Admin assigns users to appropriate roles


Client
1. Client data request
6.Data response
Internet
Proxy
2.Apply rules,restate request
5.Storage data relay
Cloud or Web
Broker
3.Modified data request
4.Storage data reply
Cloud
Cloud Storage

Secure Data in Transit

- A **potential weak link** is in the Internet between you and the data center.
- If **moving data with Globus Transfer**, you can request that data be encrypted prior to transfer.
- When using the Python SDK, you need only to set **encrypt_data=true**.
- In addition, Globus endpoints can be configured to force **encryption for all transfers** involving that endpoint, whether as source or destination.

Techniques	Description	Algorithms	IN AWS
Symmetric encryption	Encrypts and decrypts data using the same key	**Data Encryption Standard (DES), Triple Data Encryption Standard (3DES), Advanced Encryption Standard (AES)**	AWS cryptography services
Asymmetric encryption	Uses two different keys, a public key for encryption and a private key for decryption.	**Digital Signature Algorithm (DSA), RSA and Diffie-Helman Algorithm.**	
Hash functions	Hashing is using a special cryptographic function to transform one set of data into another of fixed length by using a mathematical process.	**Message Digest Algorithm (MD5) Secure Hashing Algorithm (SHA-1, SHA-2, SHA-3)**	
Key management	Securely stores and manages encryption keys to ensure the security of encrypted data. Key management define as managing cryptographic keys within a cryptosystem	Distribution of public keys. Use of public-key encryption to **distribute secrets.** **Key Authority, Certification**	AWS Key Management Service (AWS KMS)

SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

Cloud Security Cheat Sheet (AWS | Google Cloud | Azure)

Element	aws	Google Cloud	Azure
Infrastructure Security	Shield Security Hub WAF Certificate Manager	Cloud Armor Security Command Center Cloud Armor Certificate Manager	DDoS Protection Security Center WAF Key Vault
Identity Security	Cloud Trail IAM Directory Service Firewall Manager Resource Access Manager	Cloud Audit Logs IAM Managed Service Active Directory Firewall Rules Resource Manager	Azure Audit Logs Active Directory Active Directory Domain Services Firewall Manager Resource Manager
Data Security	Macie CloudHSM KMS Secrets Manager Config	Data Loss Prevention CloudHSM KMS Secret Manager Security Command Center	Information Protection Managed HSM Key Vault Key Vault Microsoft Defender
Business Security	Fraud Detector Rekognition Cognito	reCAPTCHA Enterprise Vision AI Identity Platform	Microsoft Dynamics Fraud Computer Vision Active Directory B2C

Control Who Can Access Your Data

- When you **upload data to your storage accounts**, you are responsible for managing who and what can access those data
- Role-based access control can allow you to **restrict access** from your team or your services to the data storage system
- Azure blob and table storage, the storage account has two associated keys, named key1 and key2.
- **KEY1 AS THE MASTER KEY**

- You can give **KEY2 TO COLLABORATORS**, who then have full access to the storage account.
- You can regenerate either key if you want to terminate access.
- You can use a **SHARED ACCESS SIGNATURE (SAS).**
- This is a powerful mechanism for granting limited access to objects in your storage account to others

Encrypt Your Data

- Server-side encryption :

✔ Cloud vendor automatically encrypt data on arrival in the cloud and then decrypt that data automatically each time that you access them

✔ For example, **Amazon S3 allows you to request**, when uploading data to S3, that server-side encryption be performed

✔ Amazon then performs that encryption

- # Upload the file 'test.jpg' into the newly
- created bucket
- **s3.Object('datacont', 'test.jpg').put(**
- **Body=open('/home/mydata/test.jpg', 'rb'),**
- **ServerSideEncryption='AES256')**

Encrypt Your Data

- Client-side encryption:

✔ Ensure that the cloud provider never has access to your **unencrypted data**.

✔ Amazon and Azure both provide tools that you can use to encrypt data before they are sent over the wire.

Encrypt Your Data

- Complexities of Sensitive Data:

✔ Personal health data or other sensitive information

✔ U.S., work with personal health information (PHI) must comply with the provisions of the Health Insurance Portability and Accountability Act (HIPAA)

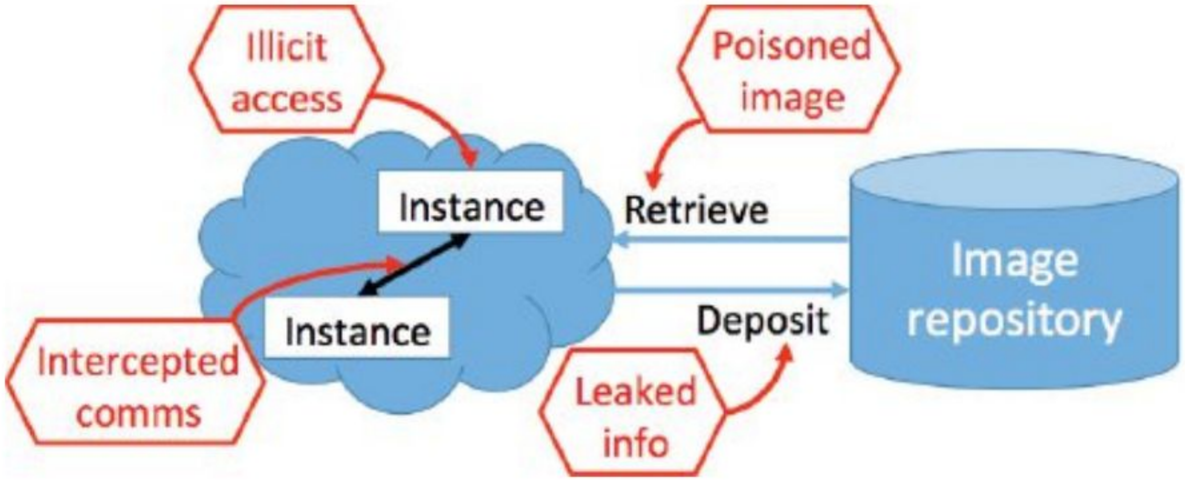
✔ Mandates administrative, physical, and technical safeguards for electronic PHI.

✔ The major commercial cloud vendors can all satisfy HIPAA physical security standards

✔ You must ensure that your entire end-to-end computing infrastructure is compliant

Secure Your VMs and Containers

- Poisoned VM or Container Image
- Illicit Access to Running VMs
- Intercepted Communications
- Information Leakage via VM Image



Poisoned VM or Container Image

- Run a VM image or launch a container that **you did not create yourself**
- Downloaded VM image may not be up to date with **security patches and thus is vulnerable to attacks**.
- Ensure that it is up to date with any patches, and run it within a secure environment.
- Cloud vendor supplies a collection of **trusted images that you can deploy**
- Cloud vendor also provides free malware tools that you may install once your image is running

Illicit Access to Running VMs

- Limit who **can access the instance**
- Ensure that **the credentials** that allow access to the instance are not compromised
- Ensure that the software running on an instance is up to date with all security patches

Intercepted Communications

- Virtual private network (VPN)
- **A VPN is a layer on top of an existing network** defined by point-to-point encrypted tunnels or a set of routes through a software defined network that carry encrypted packets.
- A VPN carries its own ip addresses and subnets that are not recognized as being part of the internet.
- You can set up a vpn in a number of ways
- Each public cloud allows you to use their cloud portal to create a VPN that solves your specific problem.
- Cloud vendor provides extensive tutorials to guide you through the process.

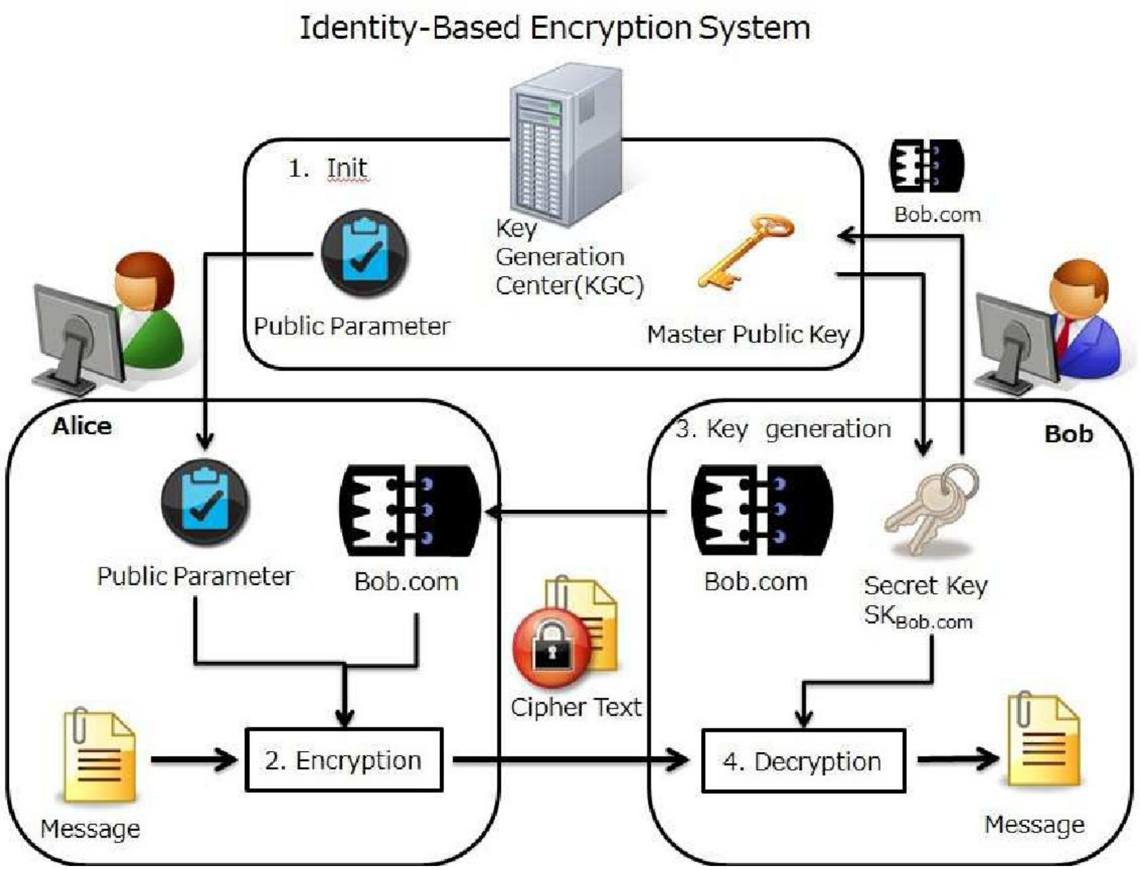
Information Leakage via VM Image

- When pushing code to GitHub, you need to make sure that the **images that you share do not contain credentials** or other confidential information.
- It is even more important if you modify a public image and then push that image to an image repository for others to use
- Amazon warns users whenever you **clone an image from any repository**, to look for any **AUTHORIZE_KEYS** files in user home directories and delete them.

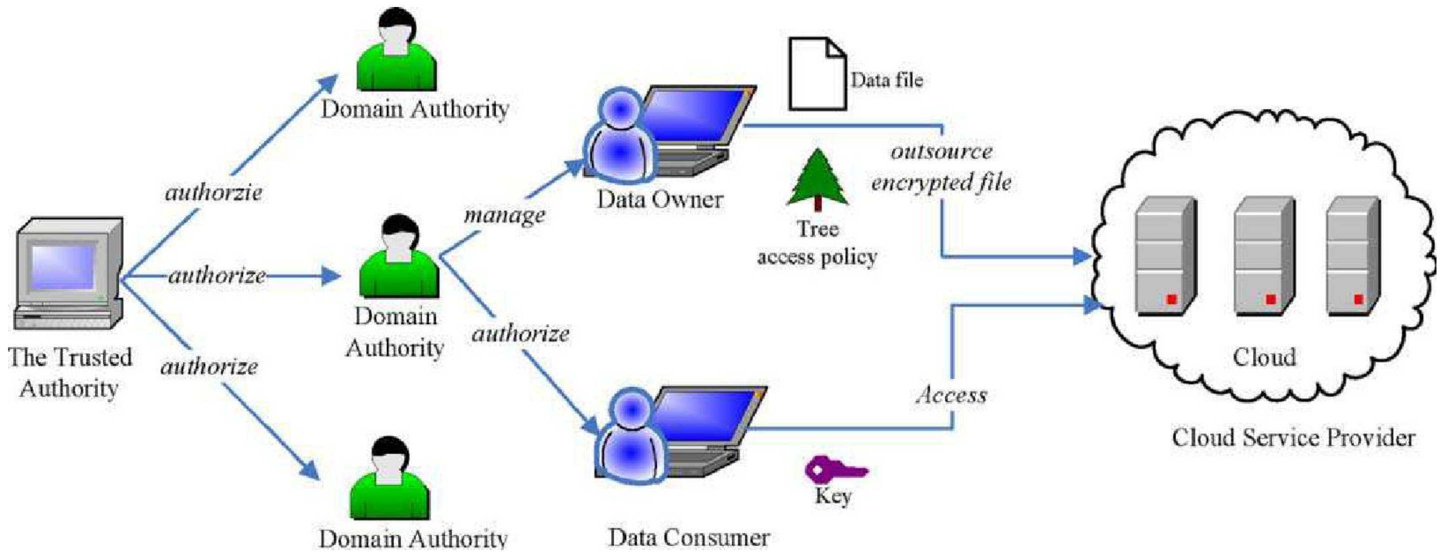
Secure Access to Cloud Software Services

- You can **create access control lists** that can add some protection if you have a way to authenticate your users.
- Another solution to the authentication problem is to use a **THIRD-PARTY AUTHENTICATION SYSTEM**.
- We have all seen online services that allow us to login using our Facebook or Google identity and password.
- The Azure app service provides a simple tool that you can use to enable Facebook, Google, or Microsoft as the authentication provider for your service

For scenario based Question



For scenario based Question



Measures	Description	Metrics
Communication	This measures the quality of the efficiency of the connection and data transfer between internal service instances, different cloud services, or between the external consumer and the cloud.	Packet loss frequency
		Connection error rate
		MPI transfer bit/byte speed
		MPI transfer delay
Computation	This represents the computing task and/or data processing in the cloud.	CPU load (%)
		Benchmark OP (FLOP) Rate
		Instance efficiency (% CPU peak)

Measures	Description	Metrics
Memory	This defines the efficiency in the rate of use of temporarily stored information contained in slow-accessed hard disk drive.	Mean hit time (s)
		Memory bit/byte speed (MB/s, GB/s)
		Random memory update rate
		Response time (ms)
Time	This specifies the time taken to run a project to completion without violating quality requirement constraints.	Computation time
		Communication time

Measures	Description	Metrics
Data security	This attribute refers to a consolidated set of technologies, policies, controls, and systems employed to protect data, applications components and other infrastructure of cloud systems.	Is SSL applicable
		Communication latency over SSL
		Auditability
Authentication	This attribute is responsible for the verification and management of valid identities of both users and devices on the cloud.	Meaning
		Sensitivity
		Effectiveness
		Confidentiality

Measures	Description	Metrics
Availability	This phenomenal attribute depicts the ability to access cloud services, data and tools anytime and anywhere.	Flexibility
		Accuracy
		Response time
Scalability	This depicts the ability to enhance the resizable computing power of the service provider´s system by adding more workload without affecting the system´s performance.	Average of assigned resources among the requested resources
Reliability	This is a critical measure of the system´s capability to continuously provision a service with an acceptable degree of efficiency without malfunctioning.	Service constancy
		Accuracy of service
		Fault tolerance
		Maturity
		Recoverability

Applications and Case studies

- **Cloud computing in education**
- **Cloud computing in healthcare**
- Cloud computing in politics
- **Cloud computing in business**
- **Cloud computing in agriculture**
- *Cloud computing adoption in Sub-Saharan Africa*
- *Cloud computing adoption in India*

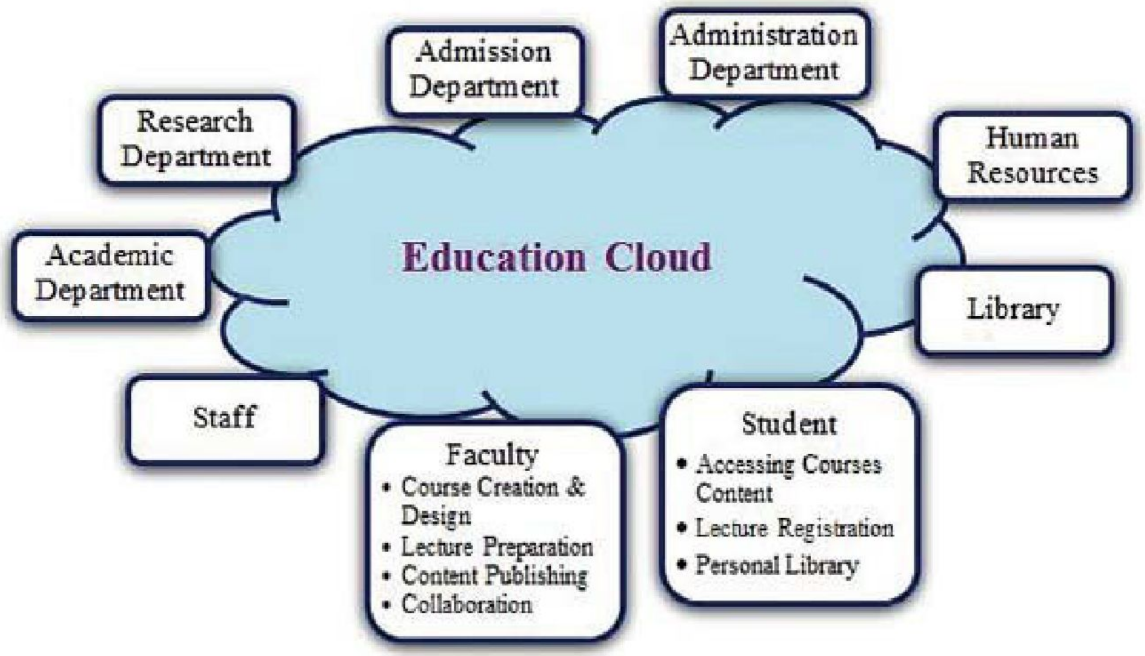
Outline will be shared as points . For Scenario based questions you want to expand the topic

For instance if you are using Google Cloud Platform (need to explain how it works with diagram and it should be mapped with the given scenario)

https://www.semanticscholar.org/paper/Cloud-Computing%3A-Demand-and-Supply-in-Banking%2C-and-Thakur-Kumar/770a81b54e82d33fbeed4ff01e4b3bbee784ea3b

Cloud computing in education

- **Information Communications Technology (ICT)** to service their academic and commercial requirements.
- Massive Open Online Courses (MOOCs)



Software-as-a-Service (SaaS)	• **YouTube is an extremely popular video resource** SaaS that many students and teachers worldwide use to watch and even download videos related to their various subjects. • Students can **watch subject-related videos to understand what is taught in class and teachers** can also watch the videos to get presentation ideas to drive home the concept of the lesson that seem difficult to teach/present to students. • Canvas, CampusAnyware, College Office, and CloudEMS.
Platform-as-a-Service (PaaS)	• An educator can design **A CUSTOMIZED VIRTUAL LAB** for the students using a paas. • They will have to possess technical know-how or hire an it team for application development
Infrastructure-as-a-Service (IaaS):	• The computer lab **configured for teaching networking** is not appropriate for teaching programming. • By adopting the cloud computing iaas, it is not necessary for the two departments to develop their own IT infrastructure to cater to their students • **Cisco Metapod, Google Compute Engine (GCE), and Microsoft Azure.**

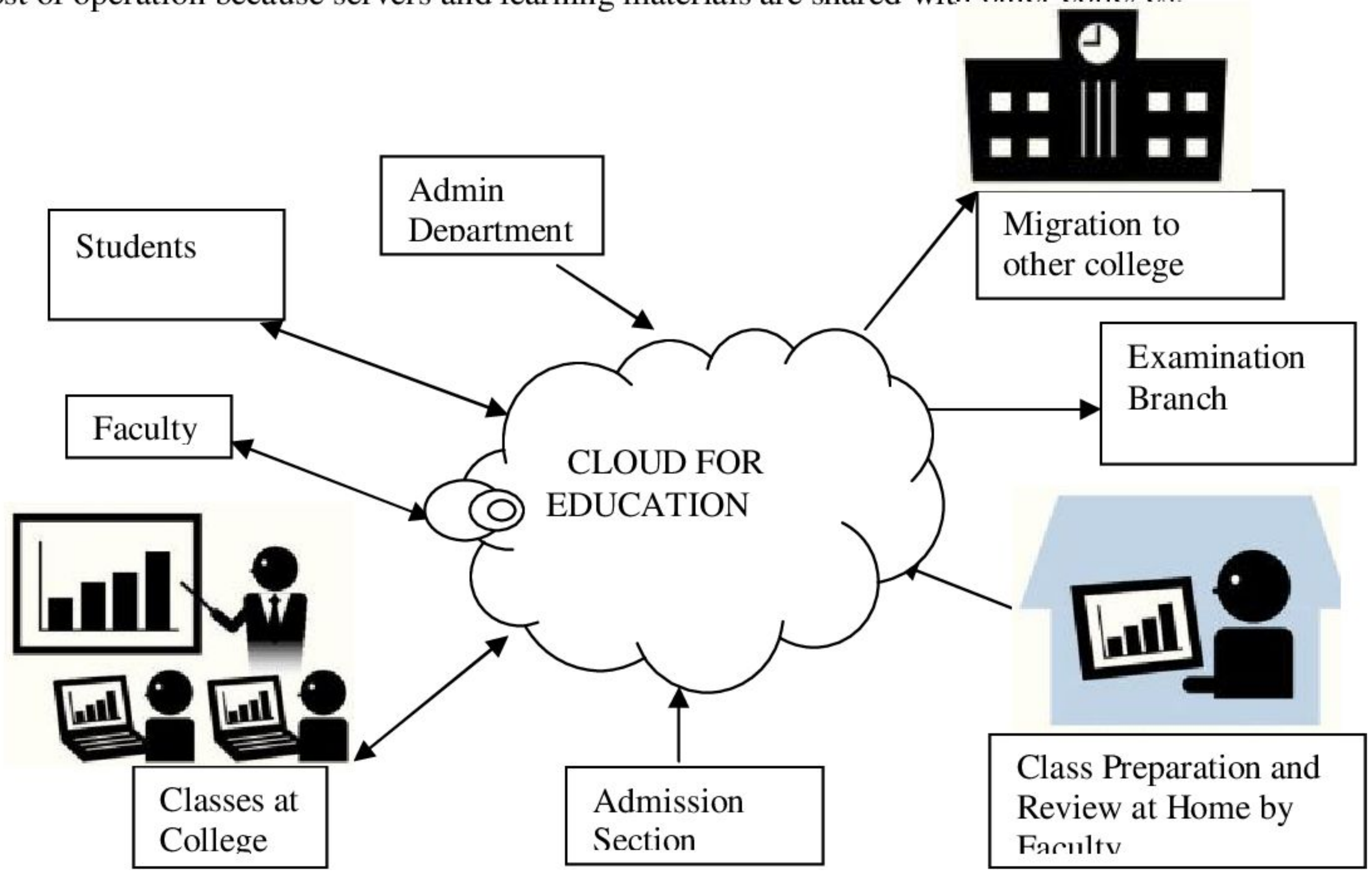
Benefits of cloud computing in higher education sector

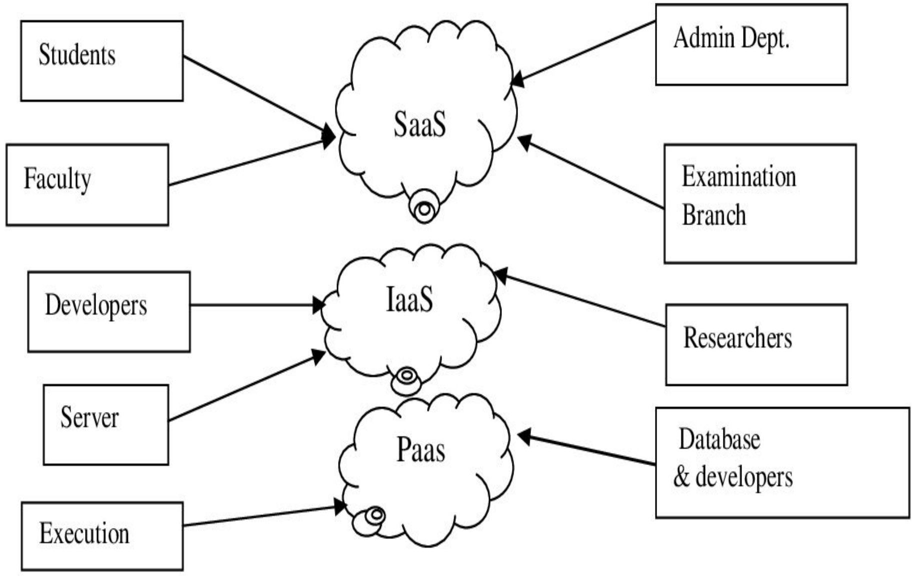
- Enhanced productivity and more efficient educational practices.
- Reduced expenditures.
- Increase in collaborative work
- Backup of information
- Support in financial and HR management
- Fostering of university accreditation

Challenges of cloud computing in education

- Security and privacy
- Real benefits
- Lack of adequate network responsiveness
- Data security
- Unsolicited advertising

cost of operation because servers and learning materials are shared with other colleges.





Cloud computing in healthcare

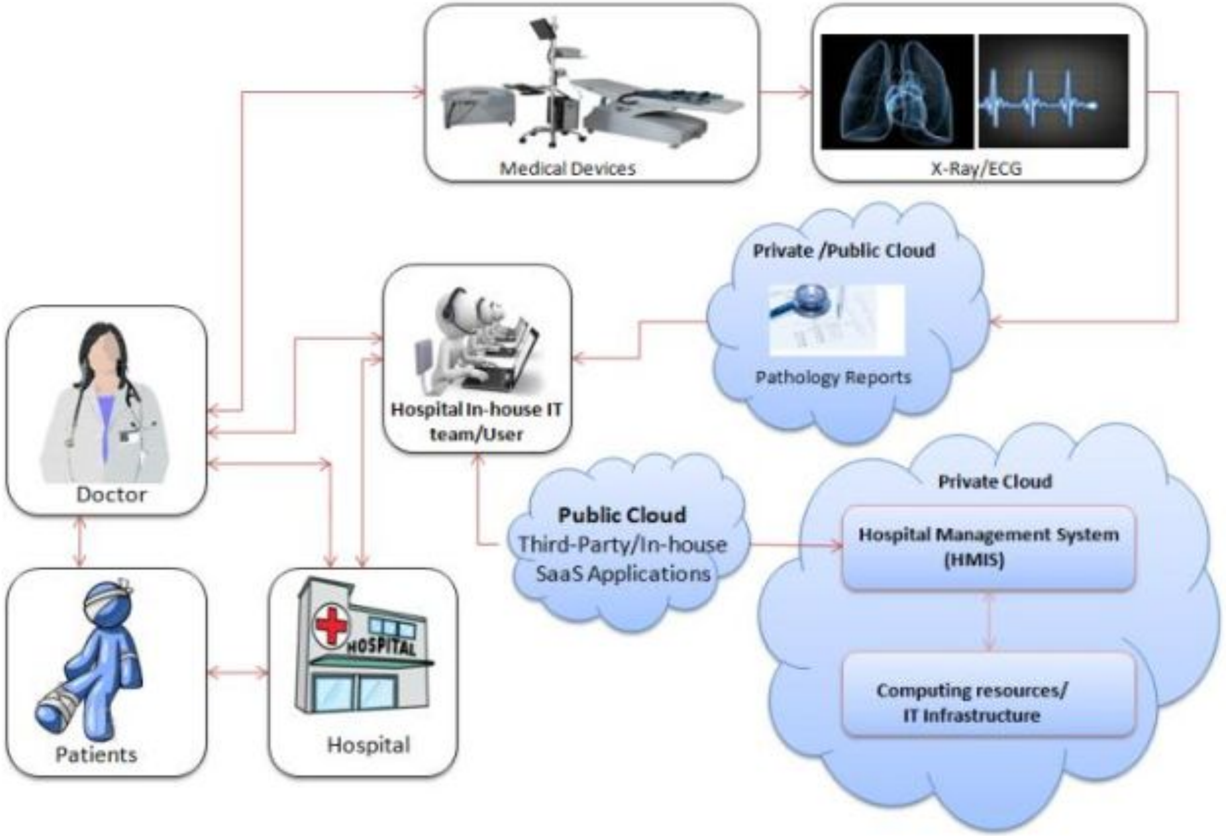


Fig. 3.1 Healthcare Ecosystem Cloud Adoption

Benefits of cloud computing in healthcare

- Mobility of records
- Speed
- Security and privacy
- Reduction of costs

Challenges of cloud computing in healthcare

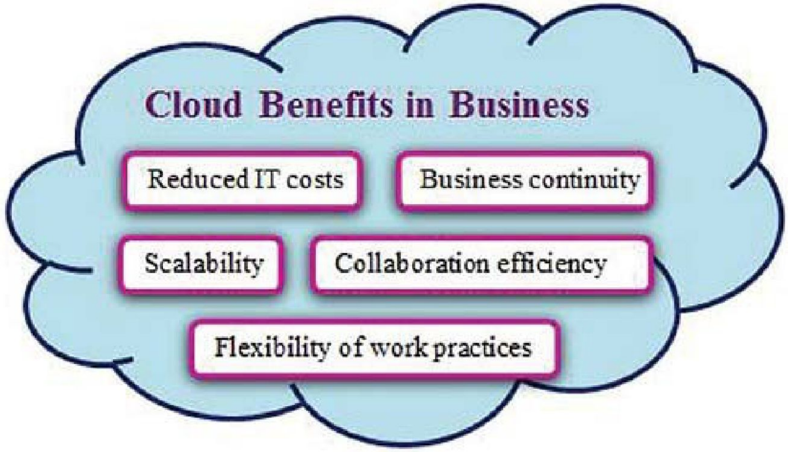
- Security
- Vendor stability and compatibility
- IT skills

Cloud computing in politics

- Barack Obama's 2012 presidential campaign ran on the AWS Elections as a service cloud platform which was specifically designed to provide elections management support for his political activities while he and his associates focused on politics without worrying about the underlying technology.
- Capital expenditure
- Political periods
- Diverse levels of access

Cloud computing in business

- Reduced IT costs
- Scalability
- Business continuity
- Collaboration efficiency
- Flexibility of work practices



Cloud for Business

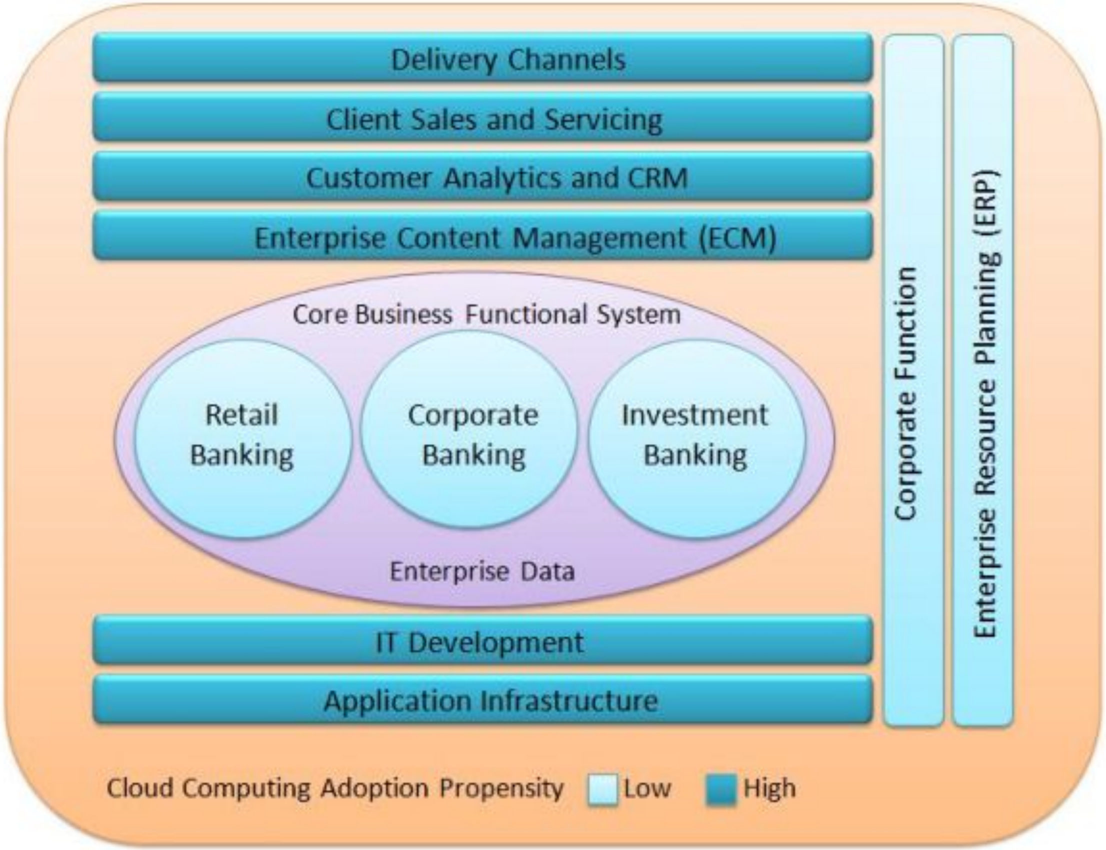
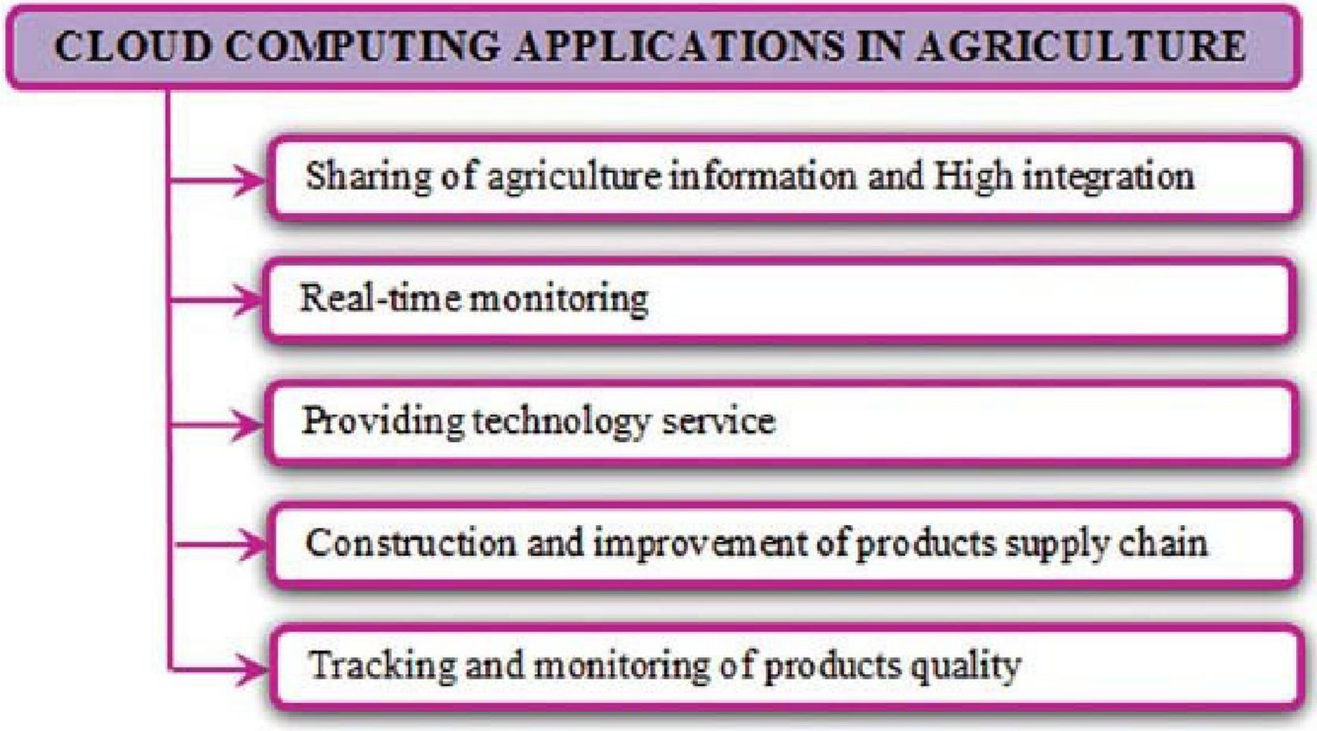


Fig. 2.1 Area Based Suited for Cloud Adoption

Cloud computing in agriculture



- Cloud computing adoption in Sub-Saharan Africa
- Cloud computing adoption in India
    - *Surge in demand for cross-domain skilled resources*
    - *Companies' reliance on SaaS*
    - *The software industry utilizing PaaS*
    - *More IaaS public cloud offerings*

Cloud for Agriculture System

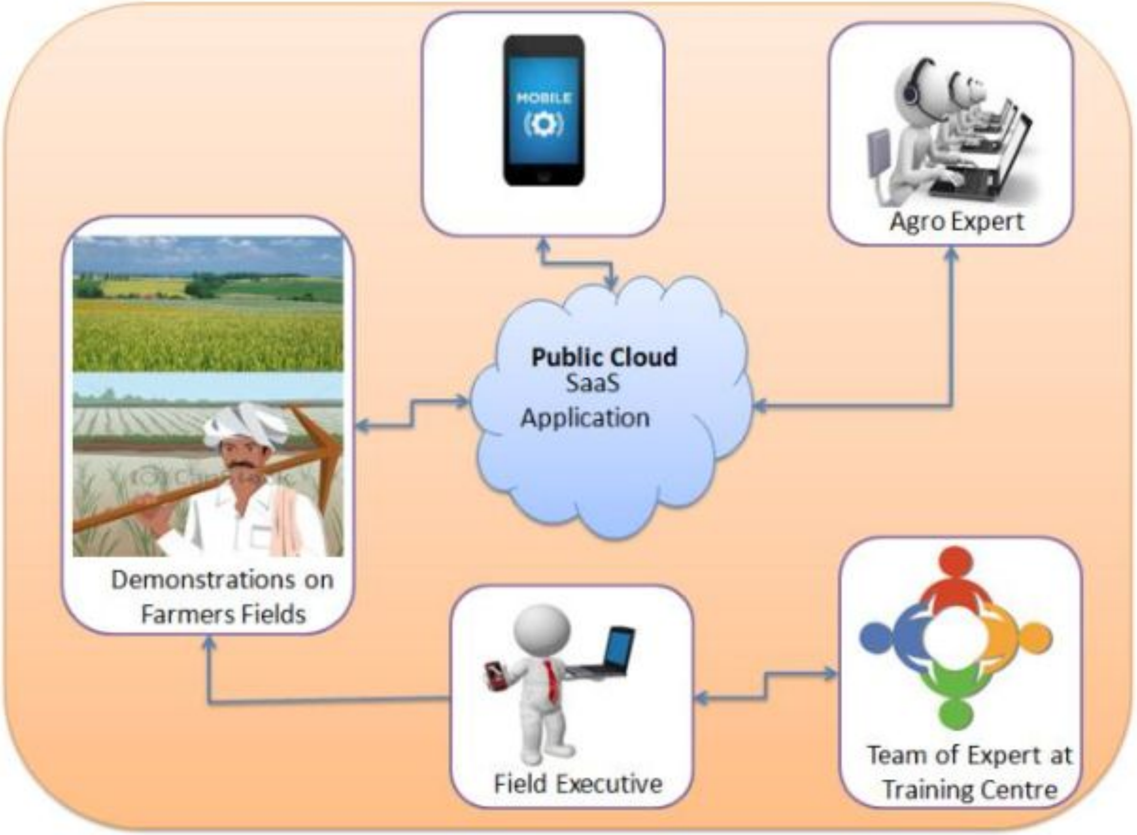
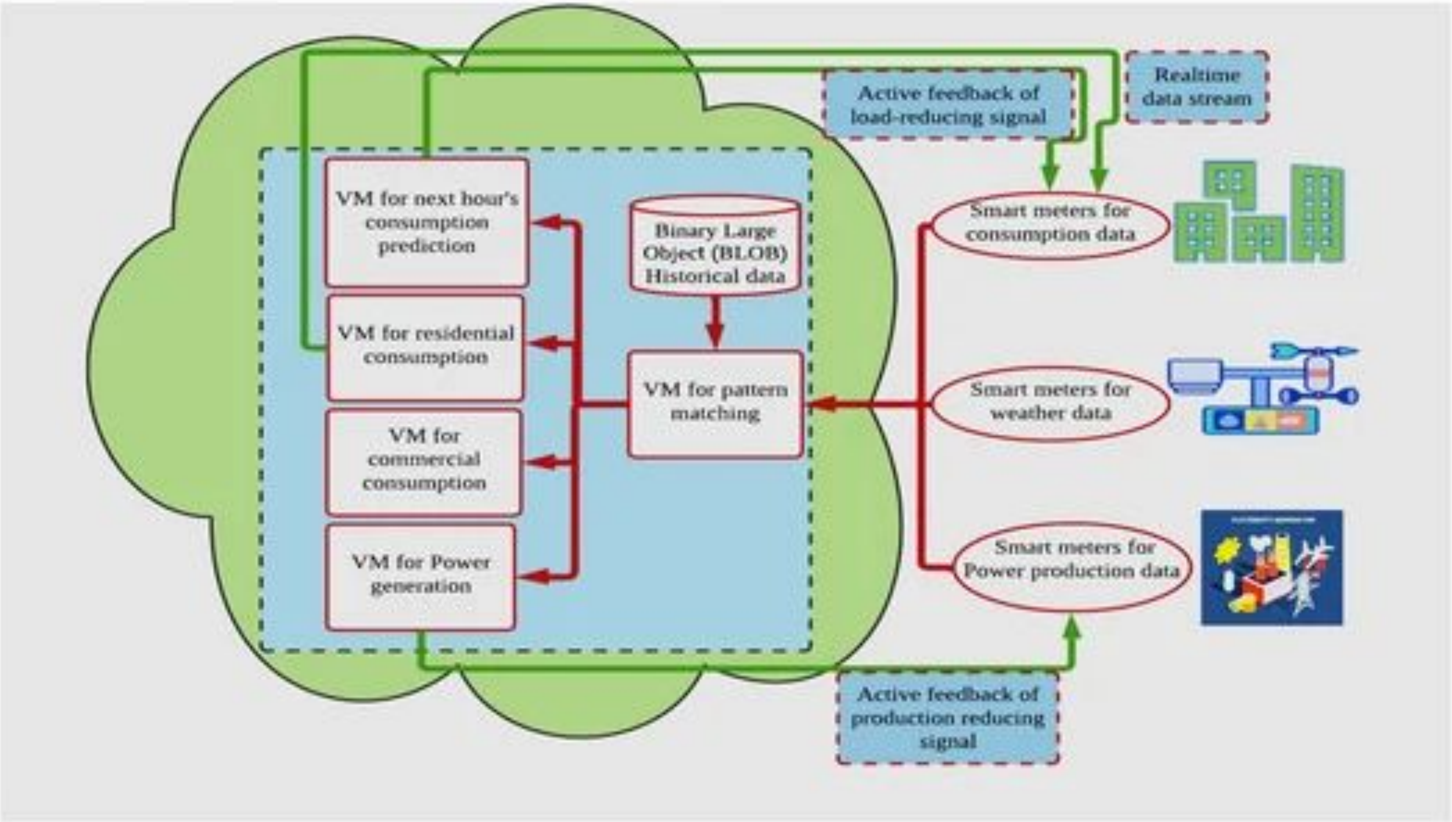
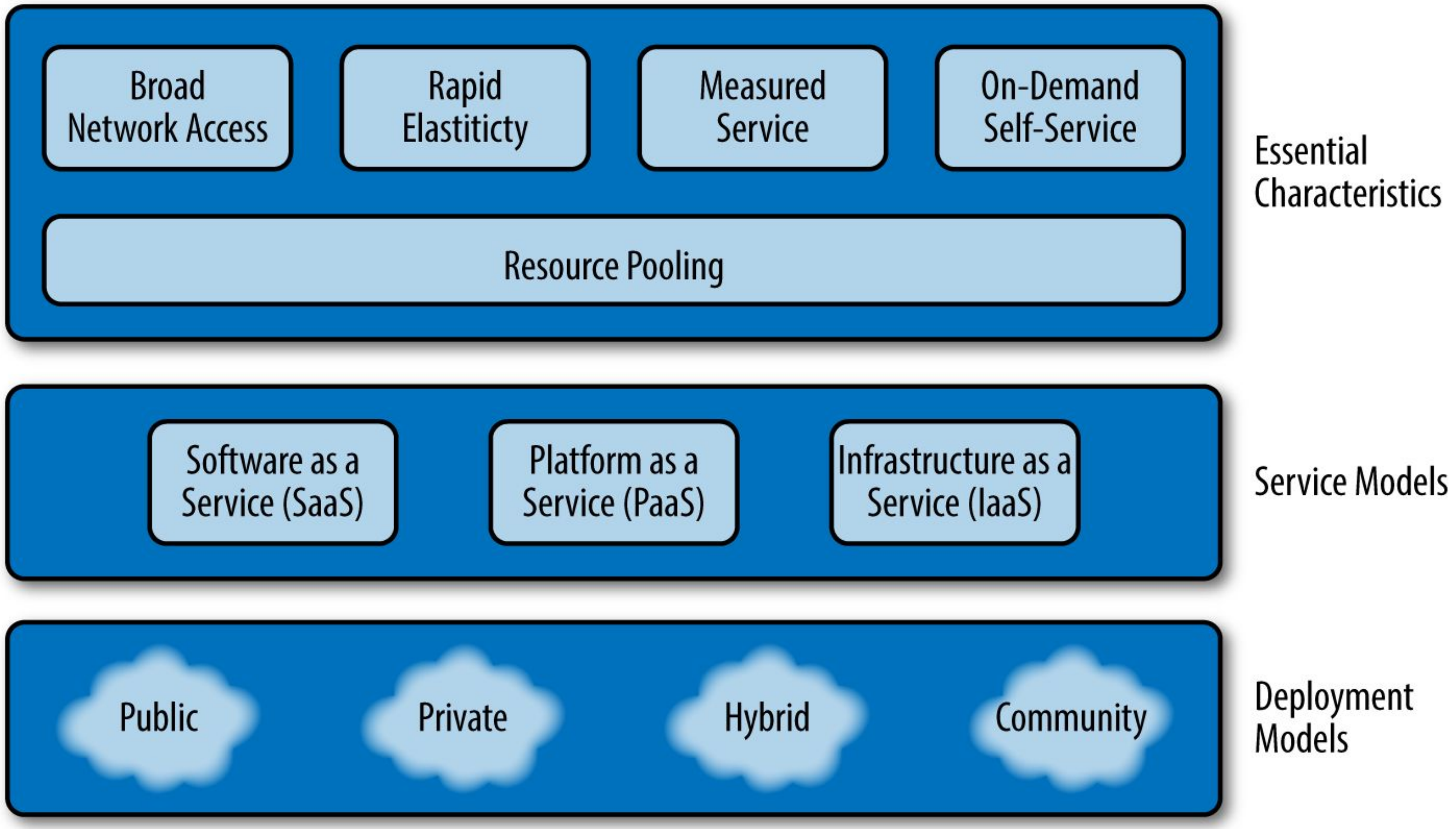


Fig. 4.1 Agriculture Ecosystem Cloud Adoption

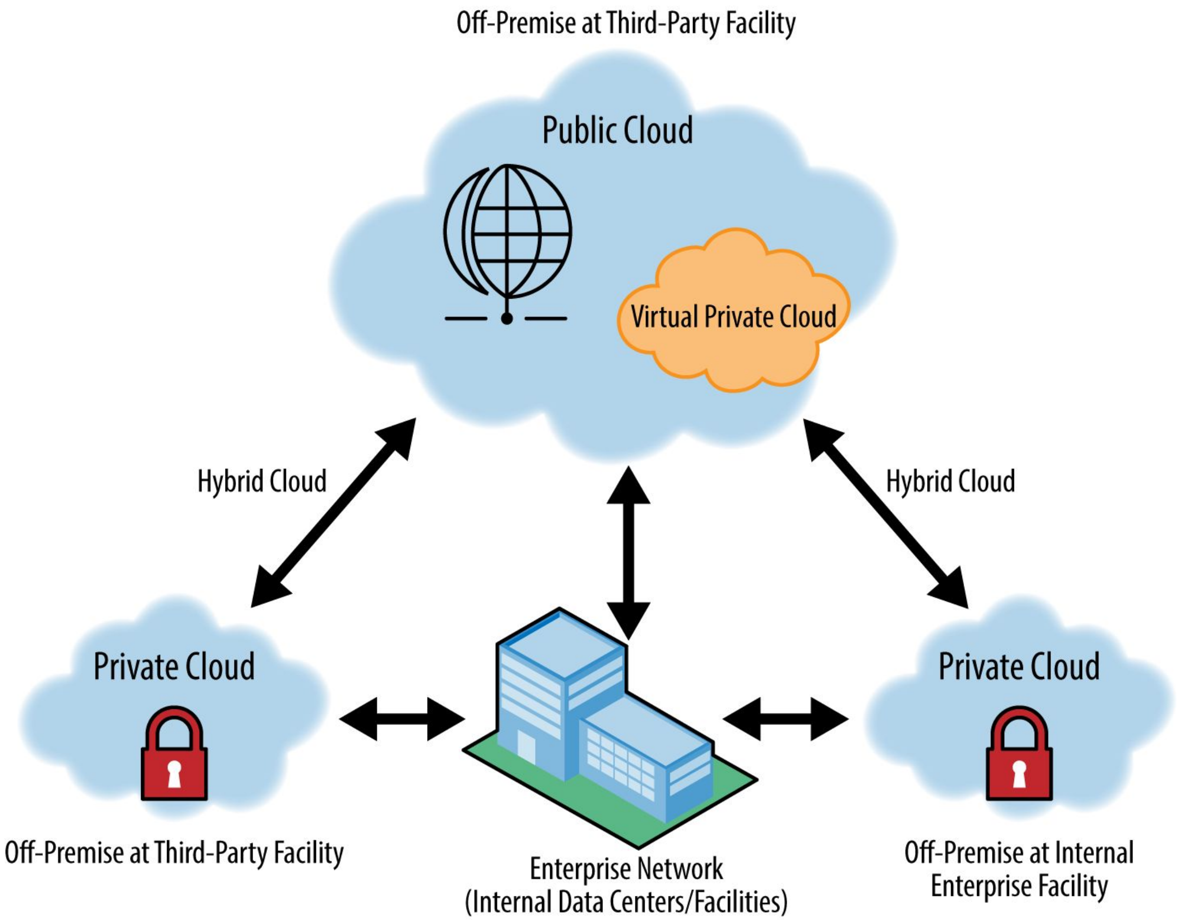
Cloud for Power Grid System



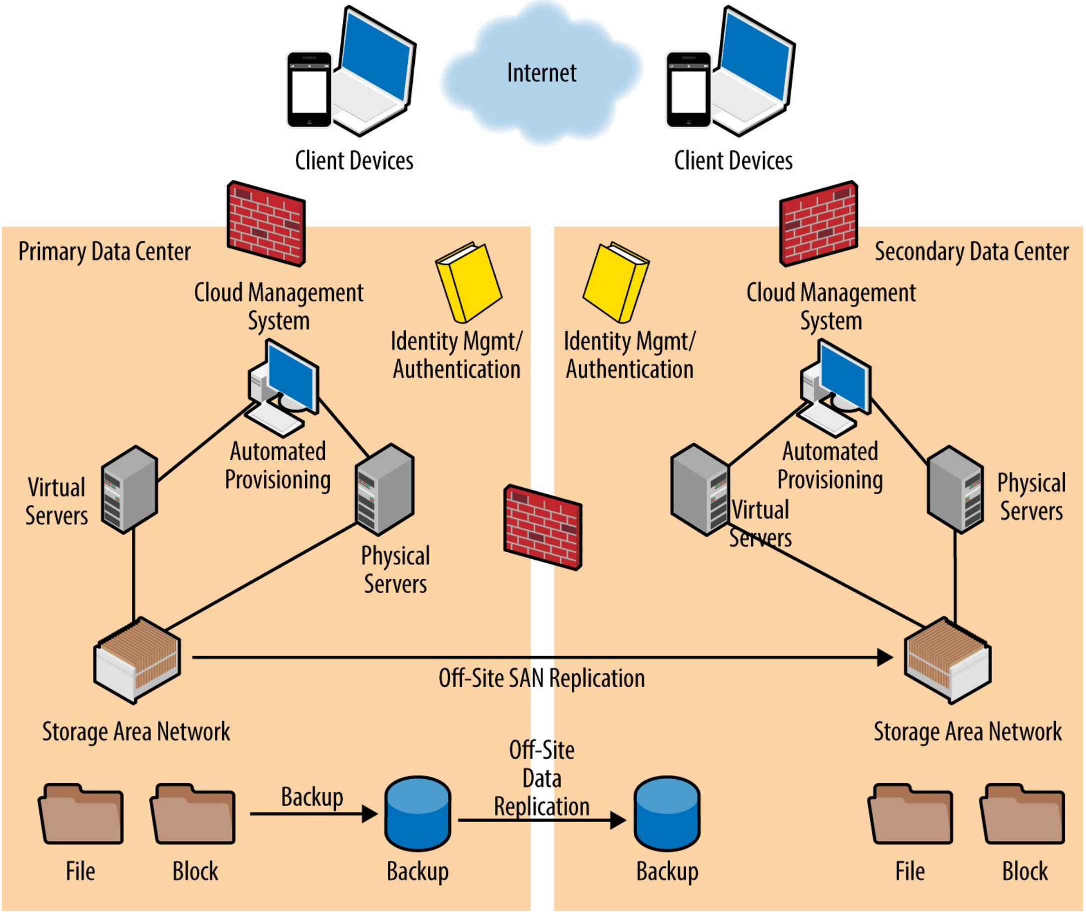
General Diagram required for scenario Questions



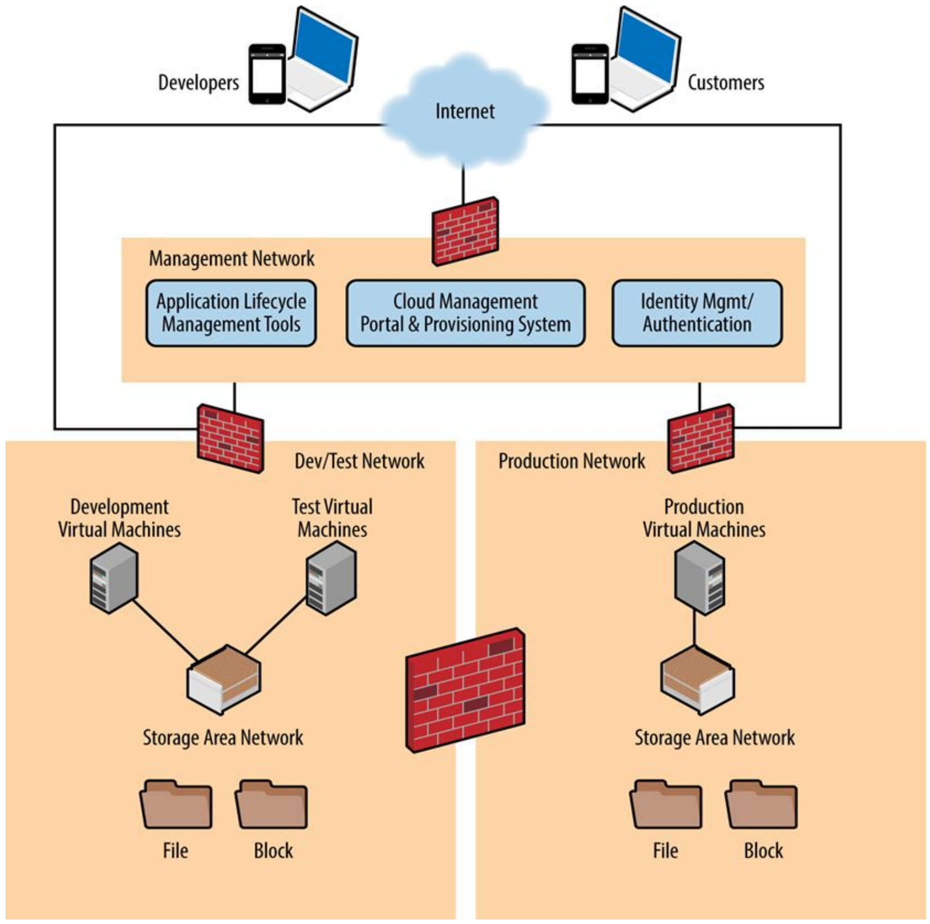
General Diagram required for scenario Questions



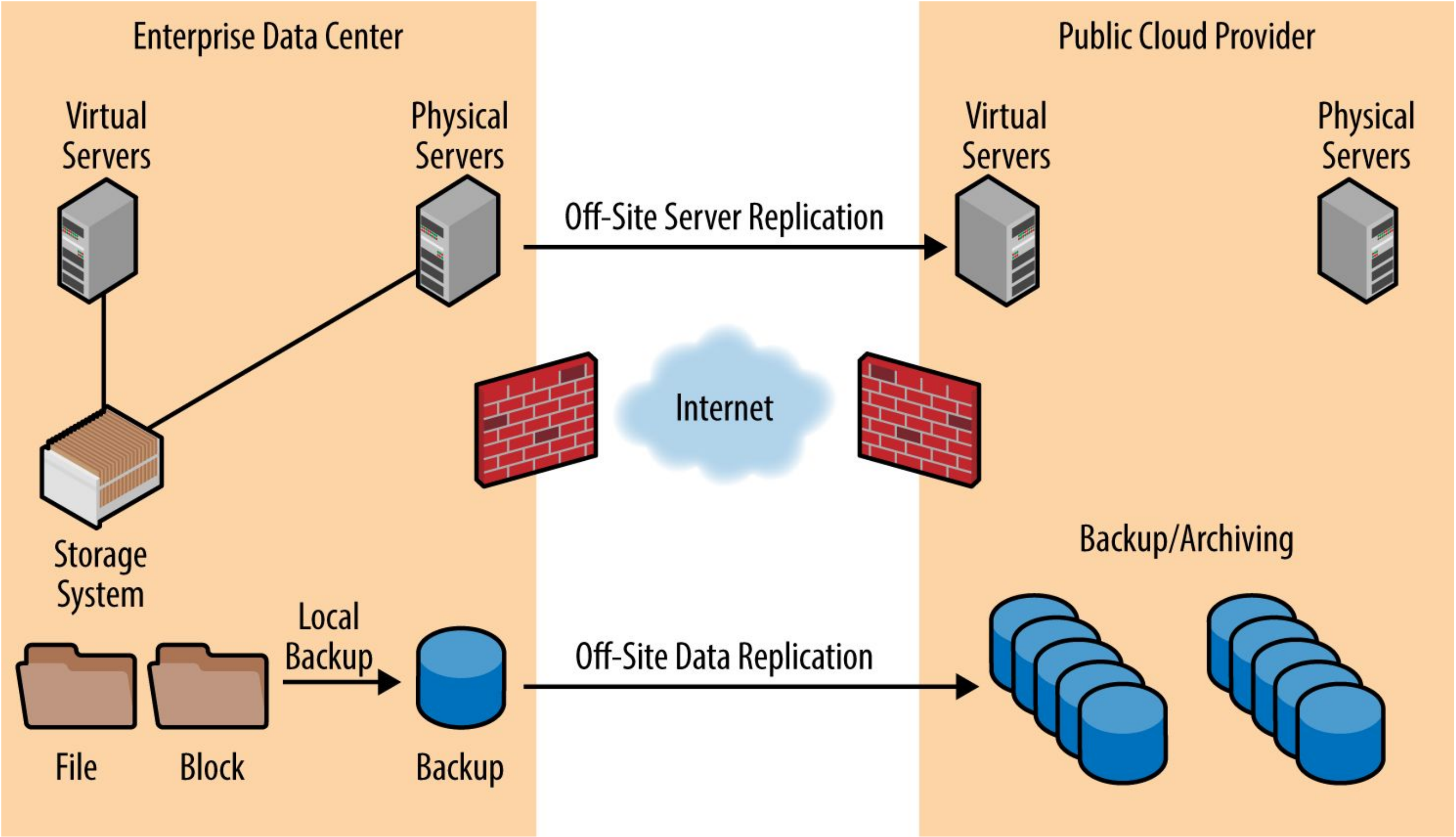
General Diagram required for scenario Questions



General Diagram required for scenario Questions



General Diagram required for scenario Questions



Cloud Comparison Cheat Sheet

blog.bytebytego.com

aws	Azure	Google Cloud	ORACLE CLOUD
Elastic Compute Cloud (EC2)	Virtual Machine	Compute Engine	Virtual Machine
Elastic Kubernetes Service (EKS)	Azure Kubernetes Service (AKS)	Google Kubernetes Engine (GKE)	Oracle Container Engine
Lambda	Azure Functions	Cloud Functions	OCI Functions
Simple Storage Service (S3)	Blob Storage	Cloud Storage	Object Storage
Elastic Block Store	Managed Disk	Persistent Disk	Persistent Volume
Elastic File System	File Storage	File Store	File Storage
Virtual Private Cloud	Virtual Network	Virtual Private Cloud	Virtual Cloud Network
Route 53	DNS	Cloud DNS	DNS
Elastic Load Balancing	Load Balancer	Cloud Load Balancing	Load Balancer
Web Application Firewall	Web Application Firewall	Cloud Armor	Web Application Firewall
RDS	SQL Database	Cloud SQL	ATP
DynamoDB	Cosmos DB	Firebase Realtime Database	NoSQL Database
Redshift	Synapse Analytics	BigQuery	Autonomous Data Warehouse
Elastic MapReduce	HDInsight	Dataproc	Big Data
Kinesis	Streaming Analytics	Dataflow	Streaming
SageMaker	Machine Learning	Vertex AI	Data Science
Glue	Data Factory	Data Fusion	Data Integration
EventBridge	Event Grid	Eventarc	Events
Simple Queuing Service	Storage Queues	Pub/Sub	Streaming
Simple Notification Service	Service Bus	Firebase Cloud Messaging	Notifications
CloudWatch	Monitor	Cloud Monitoring	Monitoring
CloudFormation	Resource Manager	Deployment Manager	Resource Manager
IAM	Active Directory	Cloud Identity	IAM
KMS	Key Vault	Cloud KMS	Vault