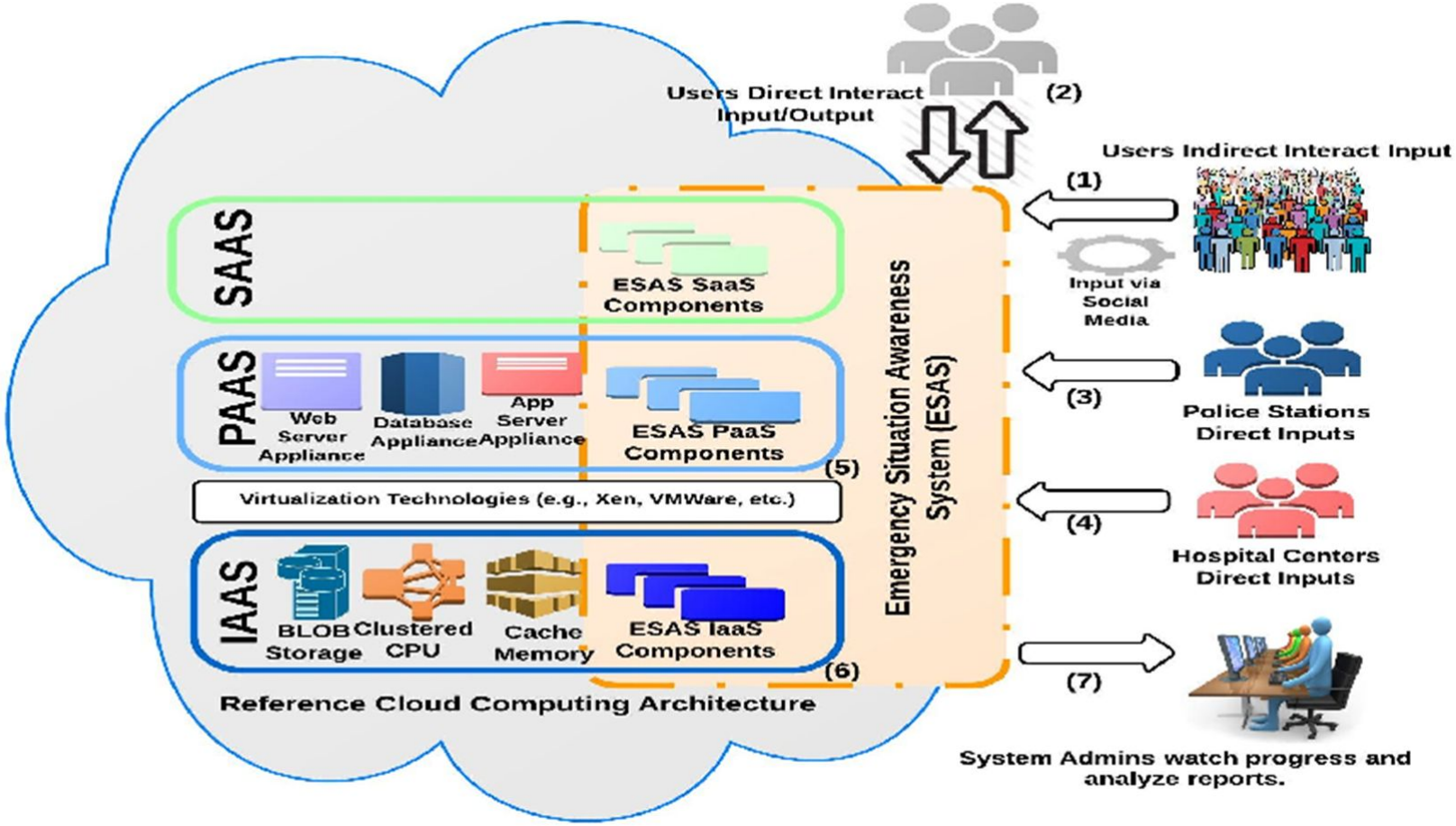
SASTRA
ENGINEERING · MANAGEMENT · LAW · SCIENCES · HUMANITIES · EDUCATION
DEEMED TO BE UNIVERSITY
(U/S 3 of the UGC Act,1956)
THINK MERIT | THINK TRANSPARENCY | THINK SASTRA

CSE407R01
CLOUD COMPUTING

What we are going to discuss??

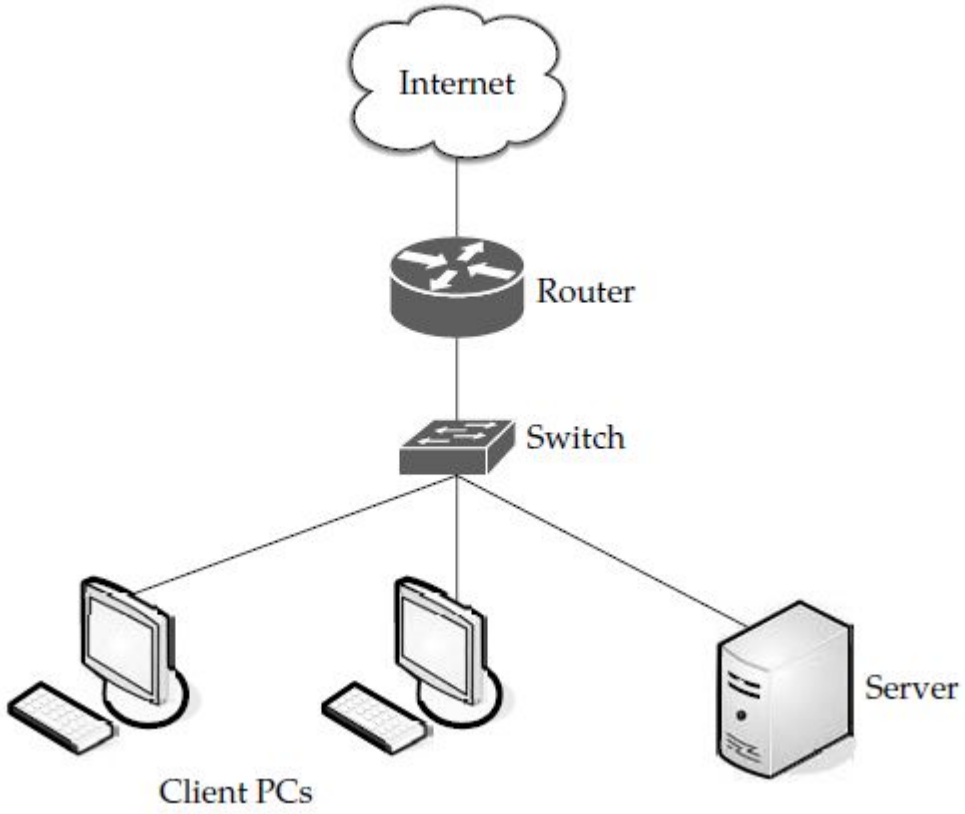
- To learn about the basic concepts of new evolving/emerging technologies in cloud computing
- To learn about the fundamentals of the cloud computing ecosystem and its characteristics
- To learn about the advantages and disadvantages of cloud computing
- To evaluate the cloud's business impact and economics
- 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

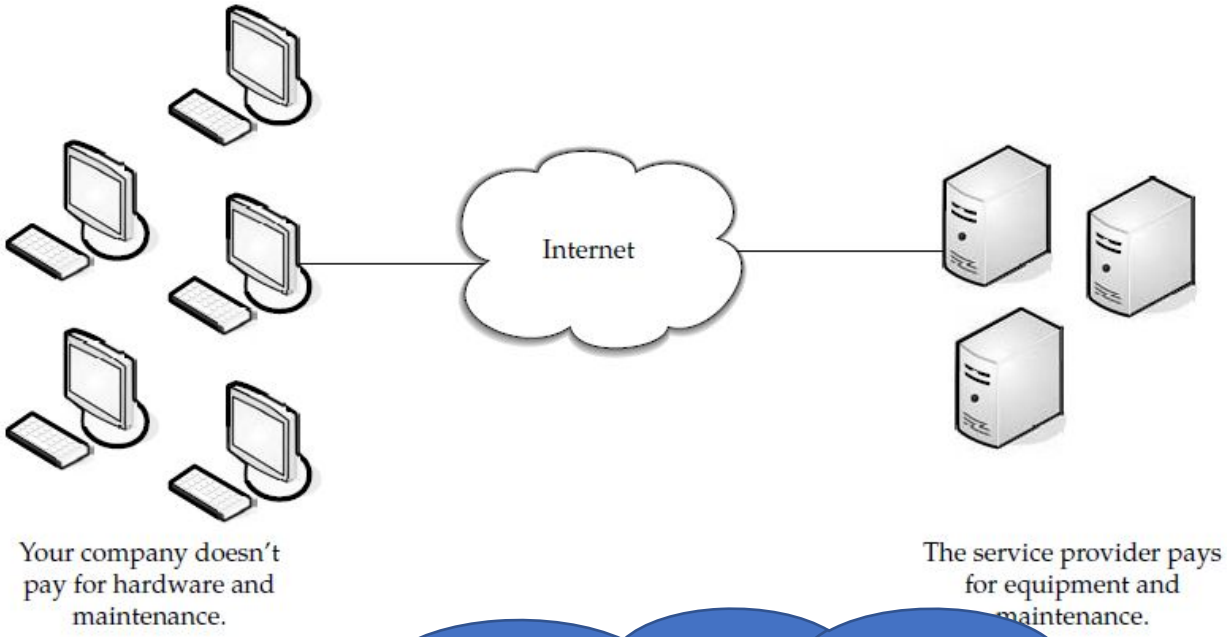


Cloud Computing: Cloud Components

- 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



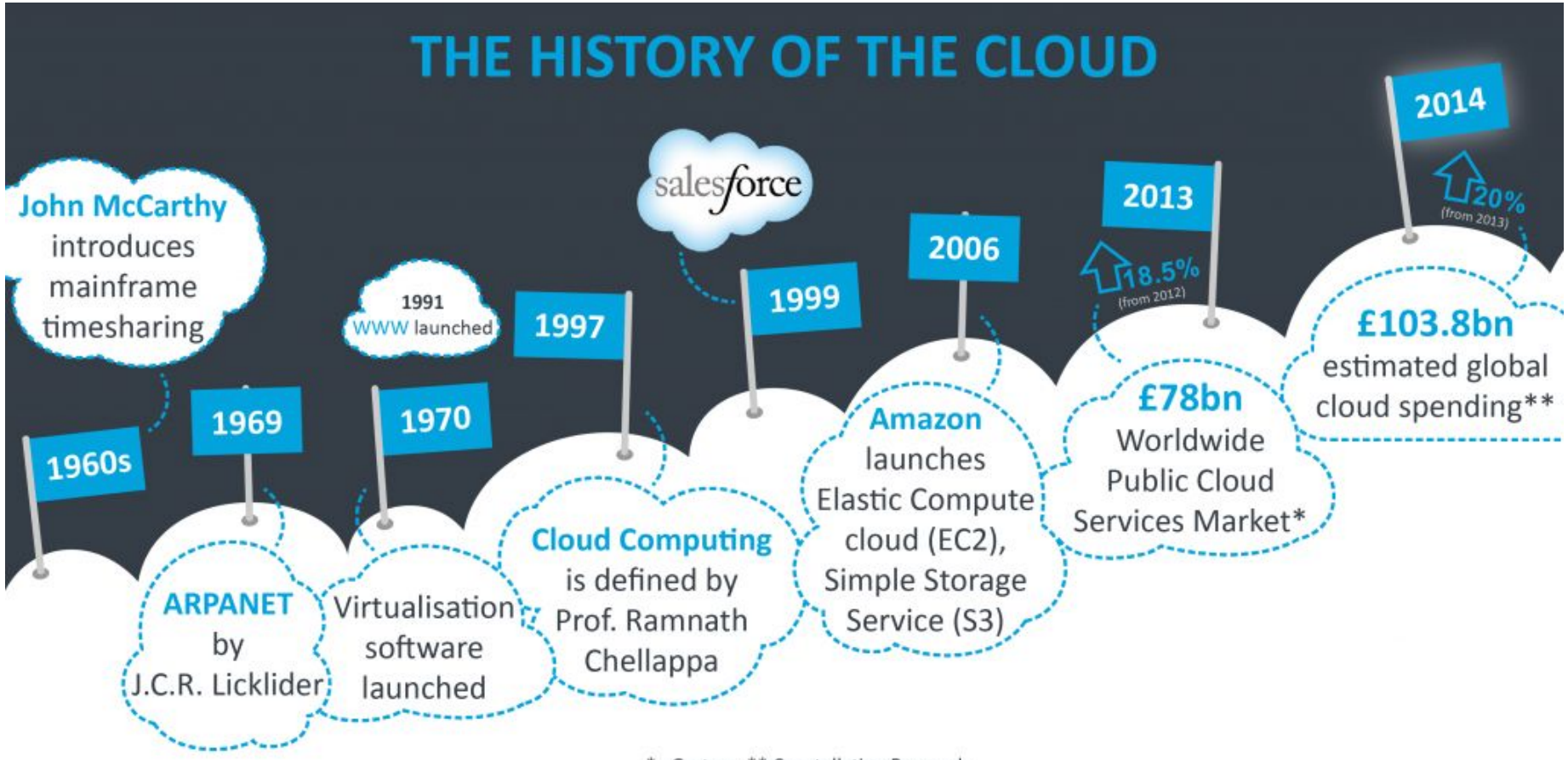
64% of companies have reduced waste and lowered energy consumptions after shifting to the cloud increased data efficiency and utilization

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



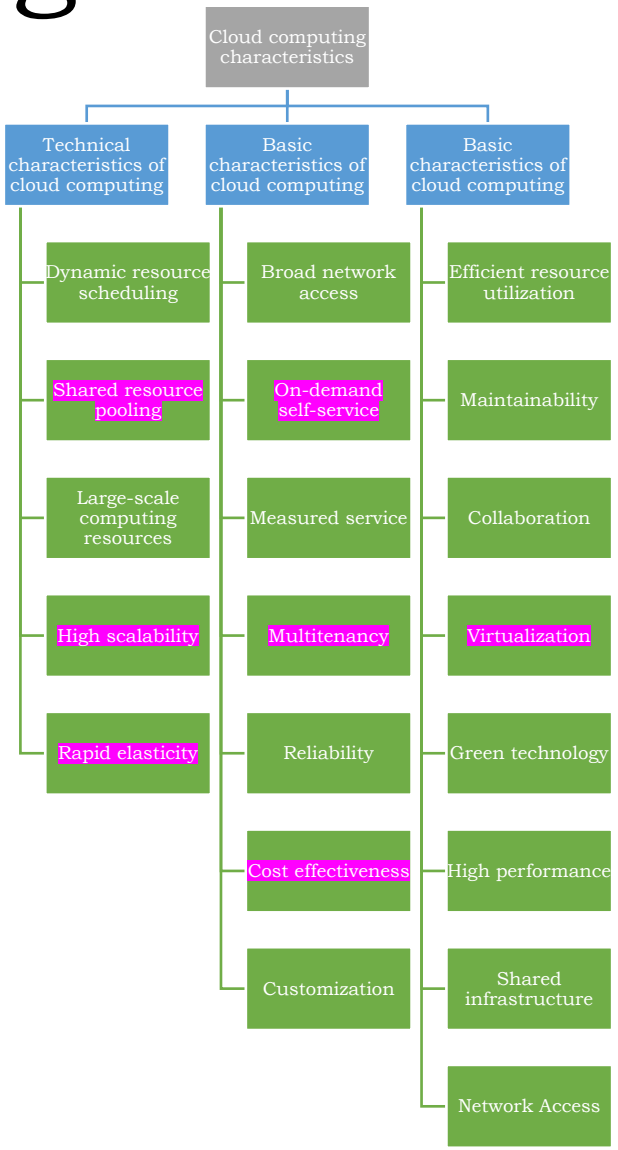
Fundamentals of the cloud computing ecosystem

- 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.

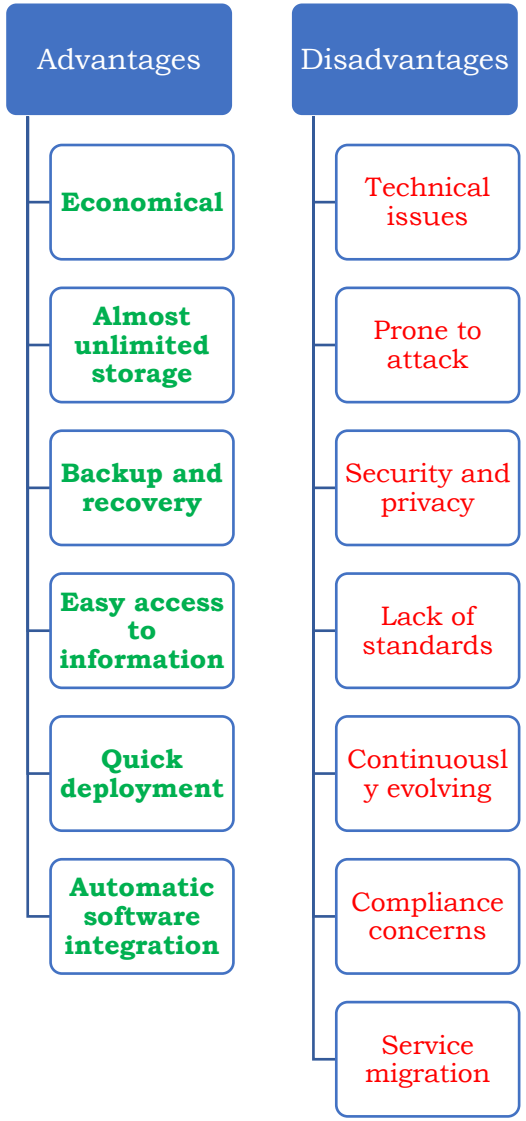
Roles	Responsibilities
Cloud stakeholders	The three primary stakeholders of the cloud include the end users, the cloud users and the cloud providers
Cloud service providers	It provide and render on-demand, payas-you-go utility computing services to cloud users
The cloud users	The provisioned services 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



Advantages and disadvantages of cloud computing

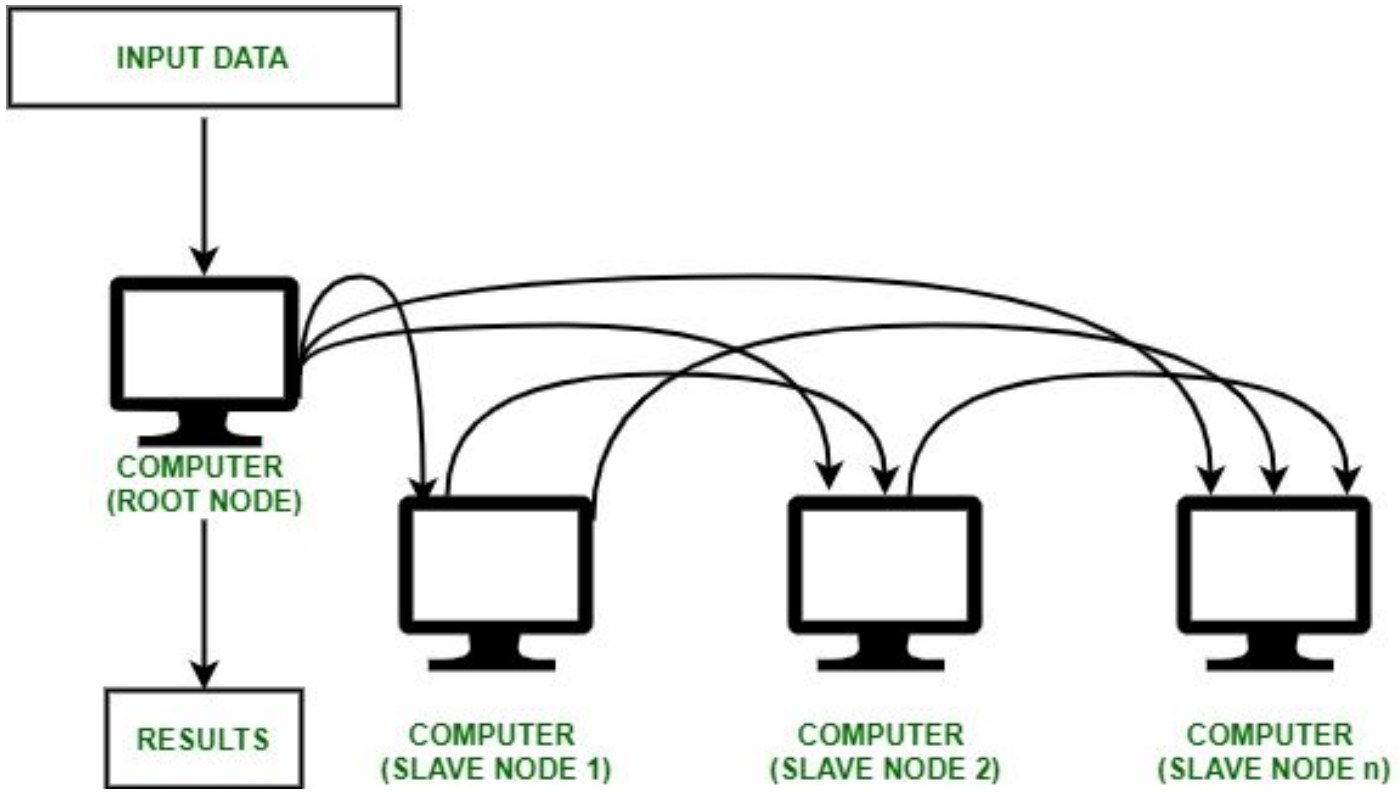


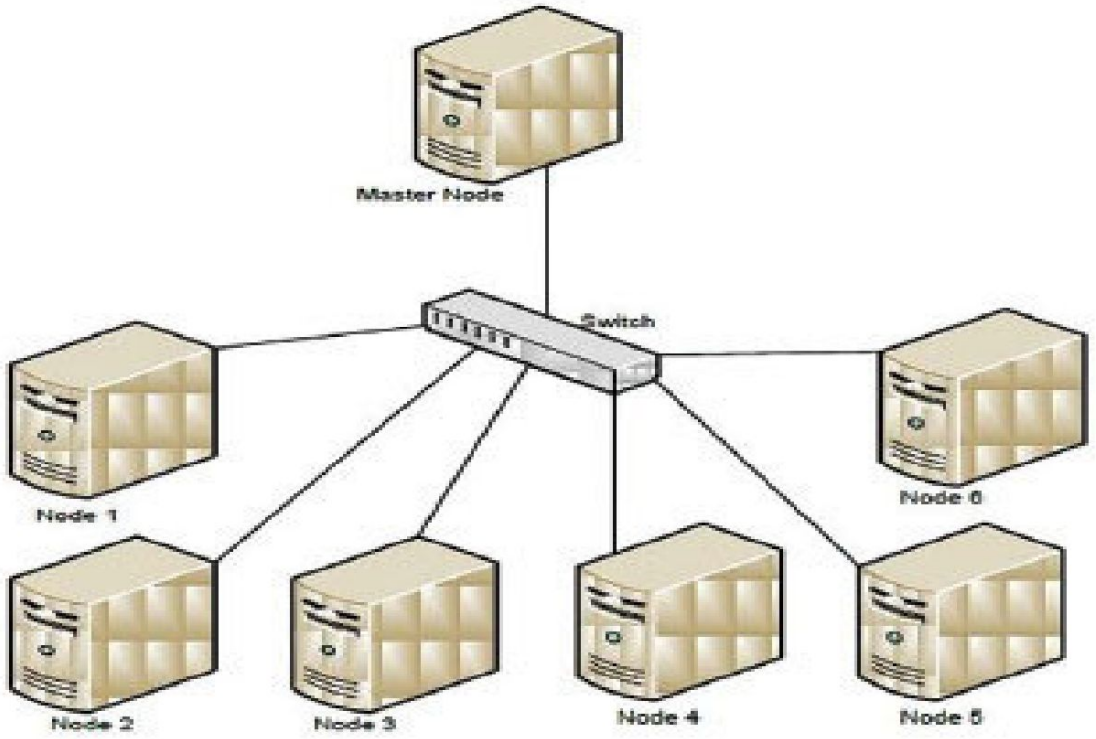
Comparison of traditional and cloud computing paradigms

- 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.

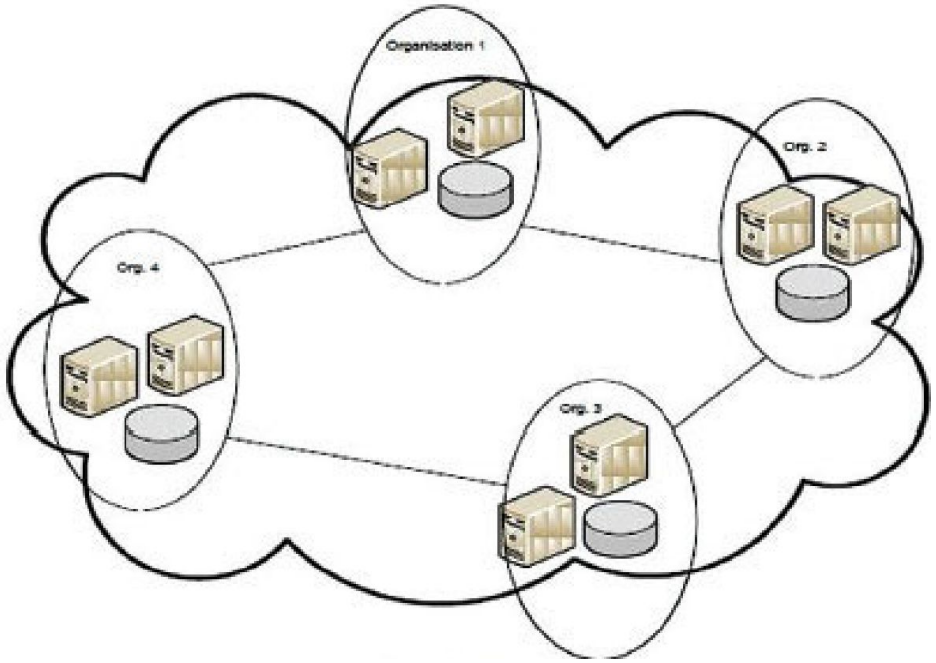
1. ***CLUSTER LOAD BALANCING***
2. ***HIGH–AVAILABILITY CLUSTERS***
3. ***HIGH-PERFORMANCE CLUSTERS***

- **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

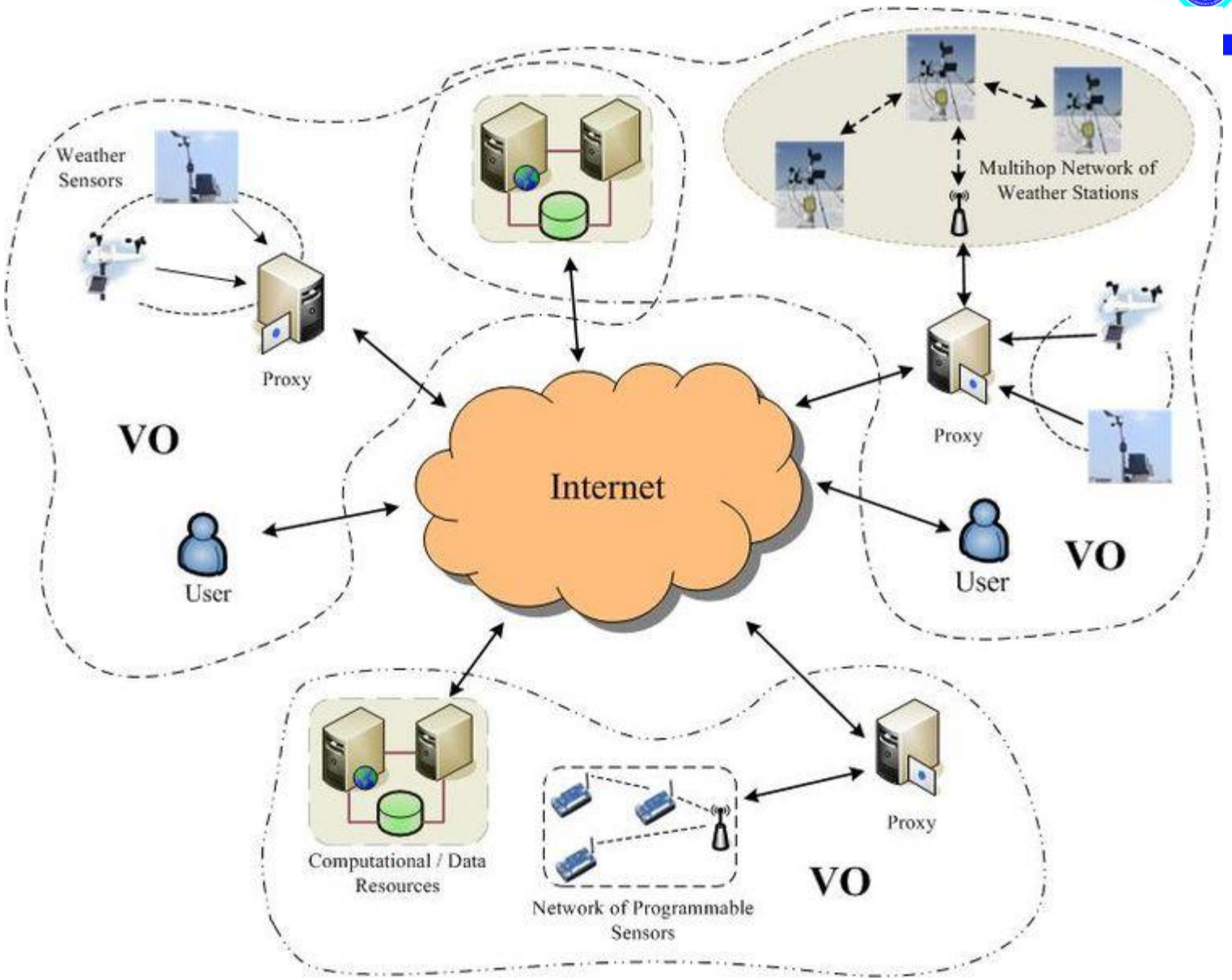
- Grid computing is defined as a distributed architecture of multiple computers connected by networks that work together to accomplish a joint task.
- Grid computing is the unification of several computing resources from several supervisory domains into one or more logical entity, coordinated with a 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
Internet
Multihop Network of Weather Stations
Proxy
User
VO
Computational / Data Resources
Network of Programmable Sensors
Proxy
VO

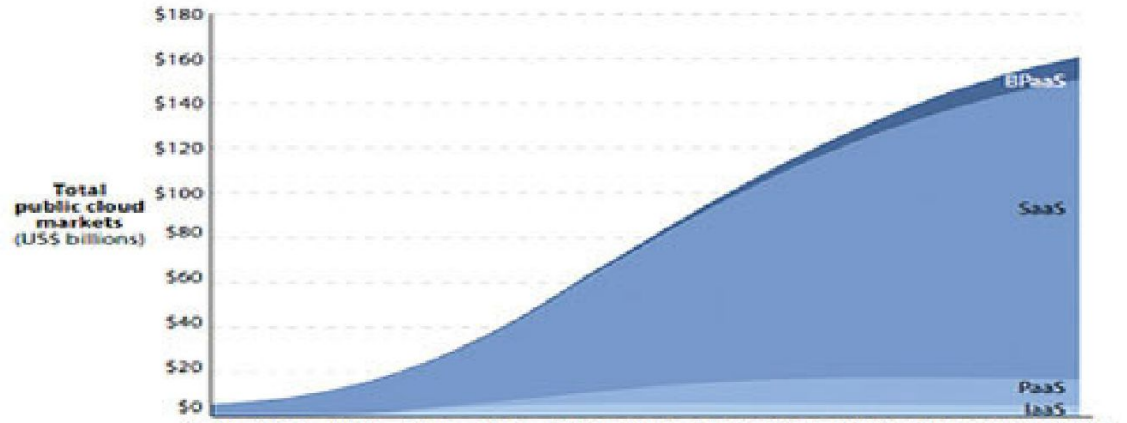
Cloud computing

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

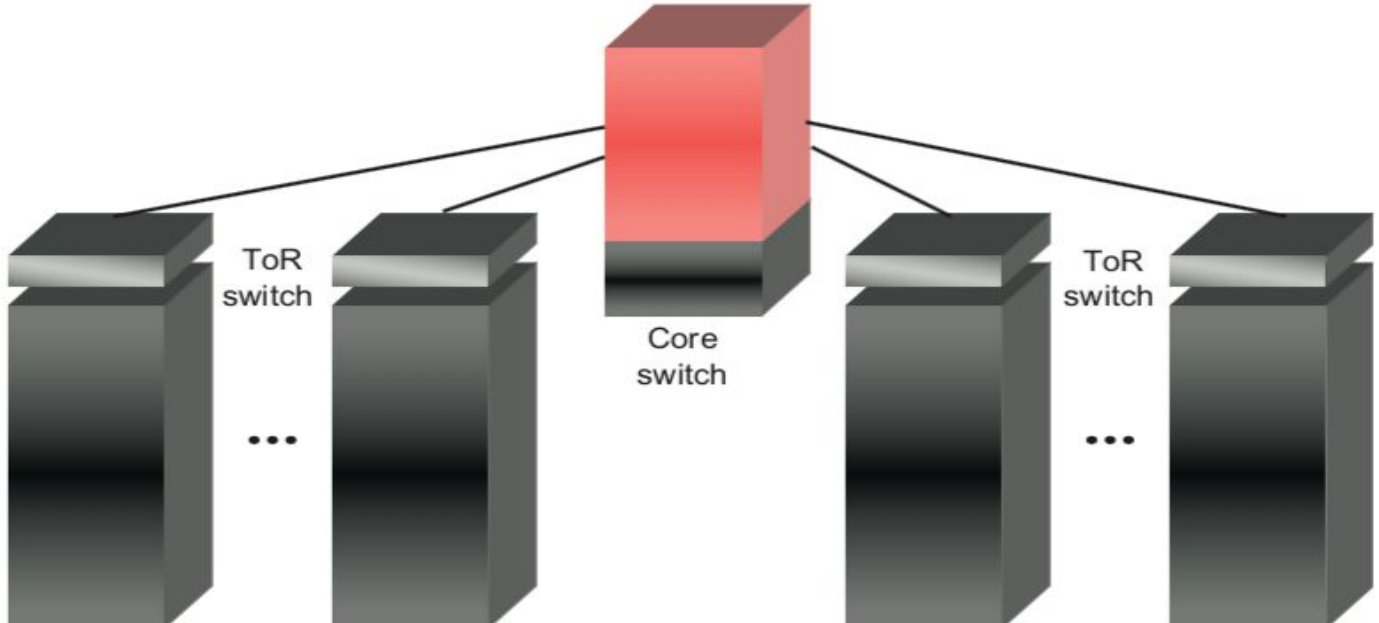
- Key business drivers of cloud computing adoption
- Help to pursue new business opportunities
- Upfront costs reduction
- Potential improvement in business continuity
- Potential reduction in carbon footprint
    - **Server costs**
    - **Storage costs**
    - **Network costs**
    - **Backup and archive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

Future of the cloud (FoC)

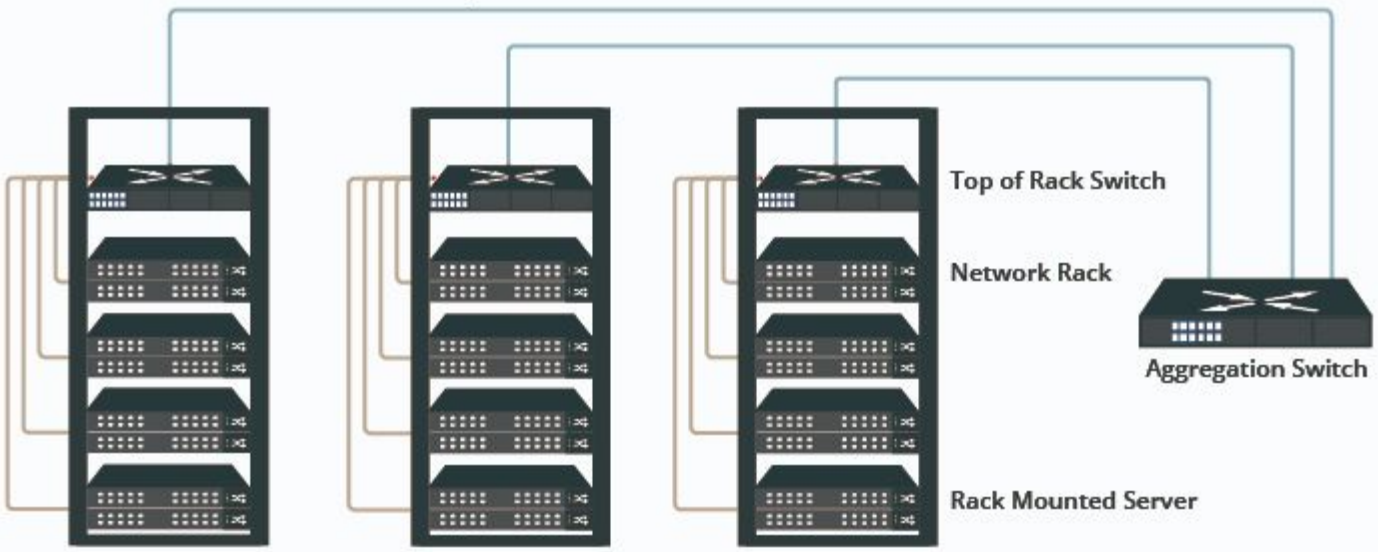
- Factors driving the future of cloud computing
- Defining and comparing services
- Enabling the next generation data centers
- Managing a hybrid world
- Everything as a service

Datacenter

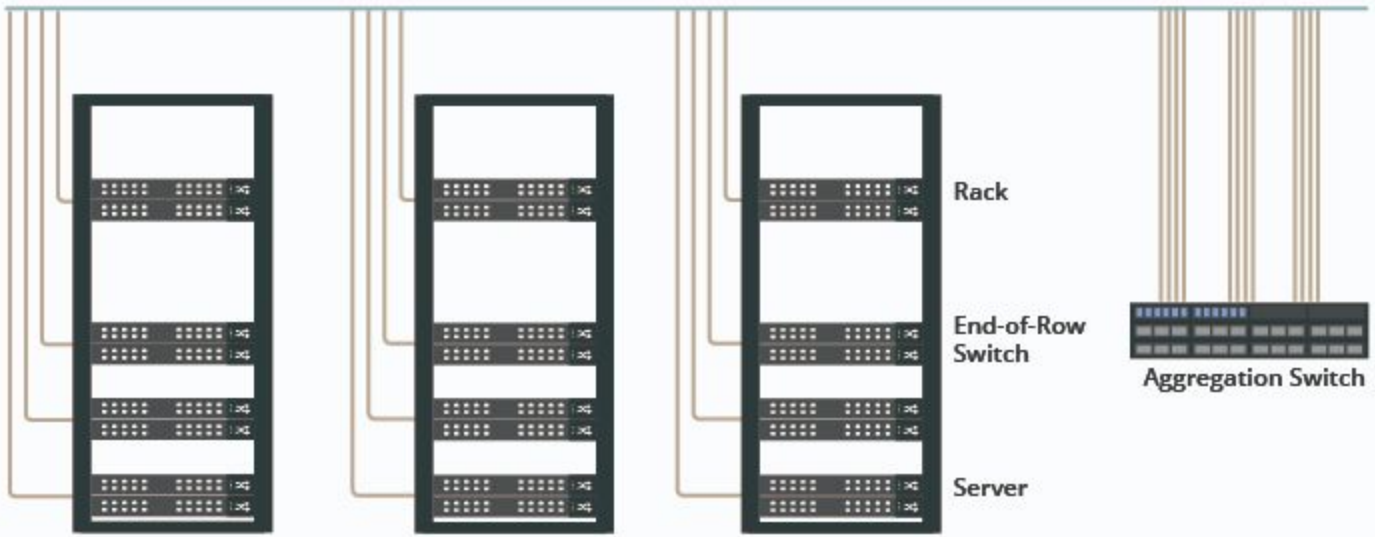
- The *datacenter* is the collection of servers where the application to which you subscribe is housed.
- It could be a large room in **the basement of your building or a room full of servers on the other side of the world that you access via the Internet.**
- A growing trend in the IT world is virtualizing 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.

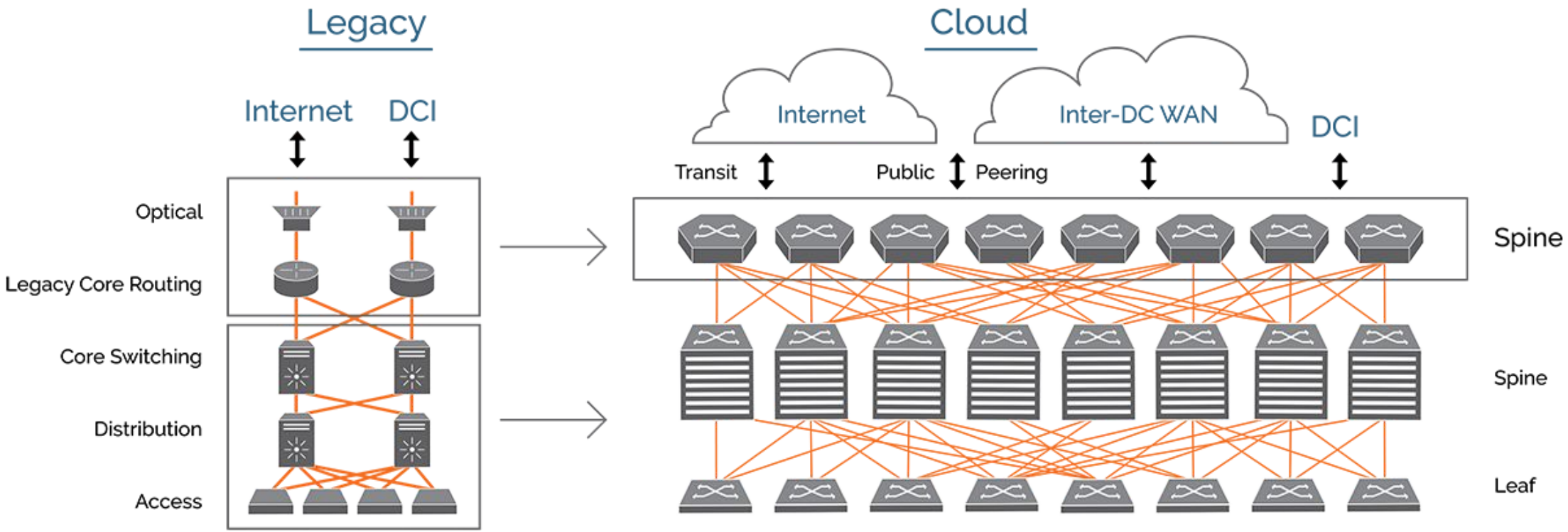


Top-of-Rack(TOR) Architecture



End of Row(EOR) Architecture

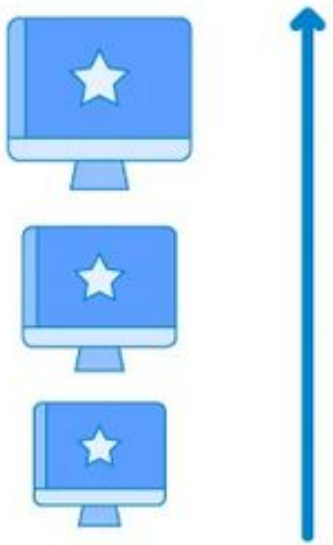




https://www.arista.com/en/products

VERTICAL SCALING

Increase size of instance
(RAM, CPU etc.)

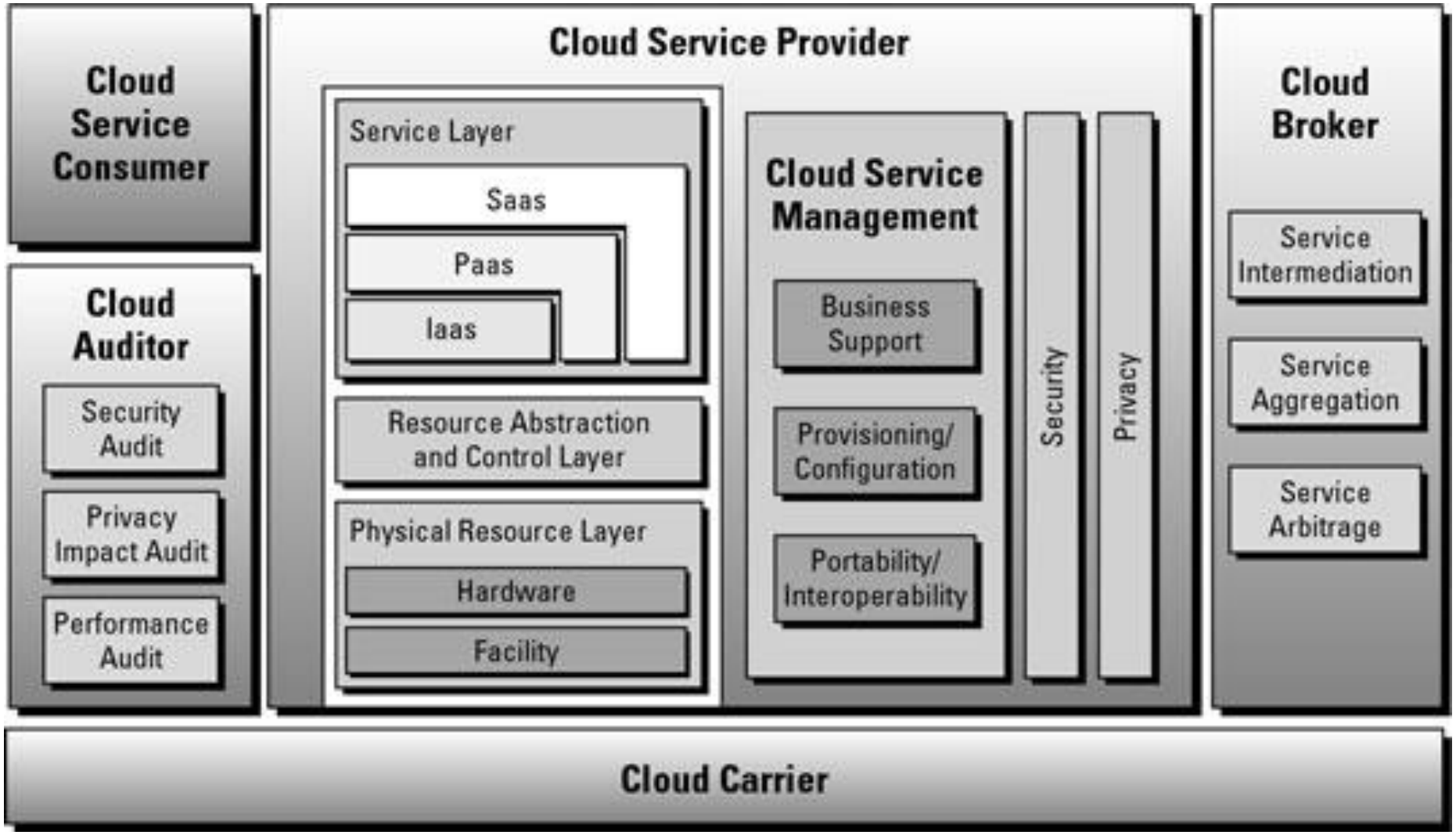


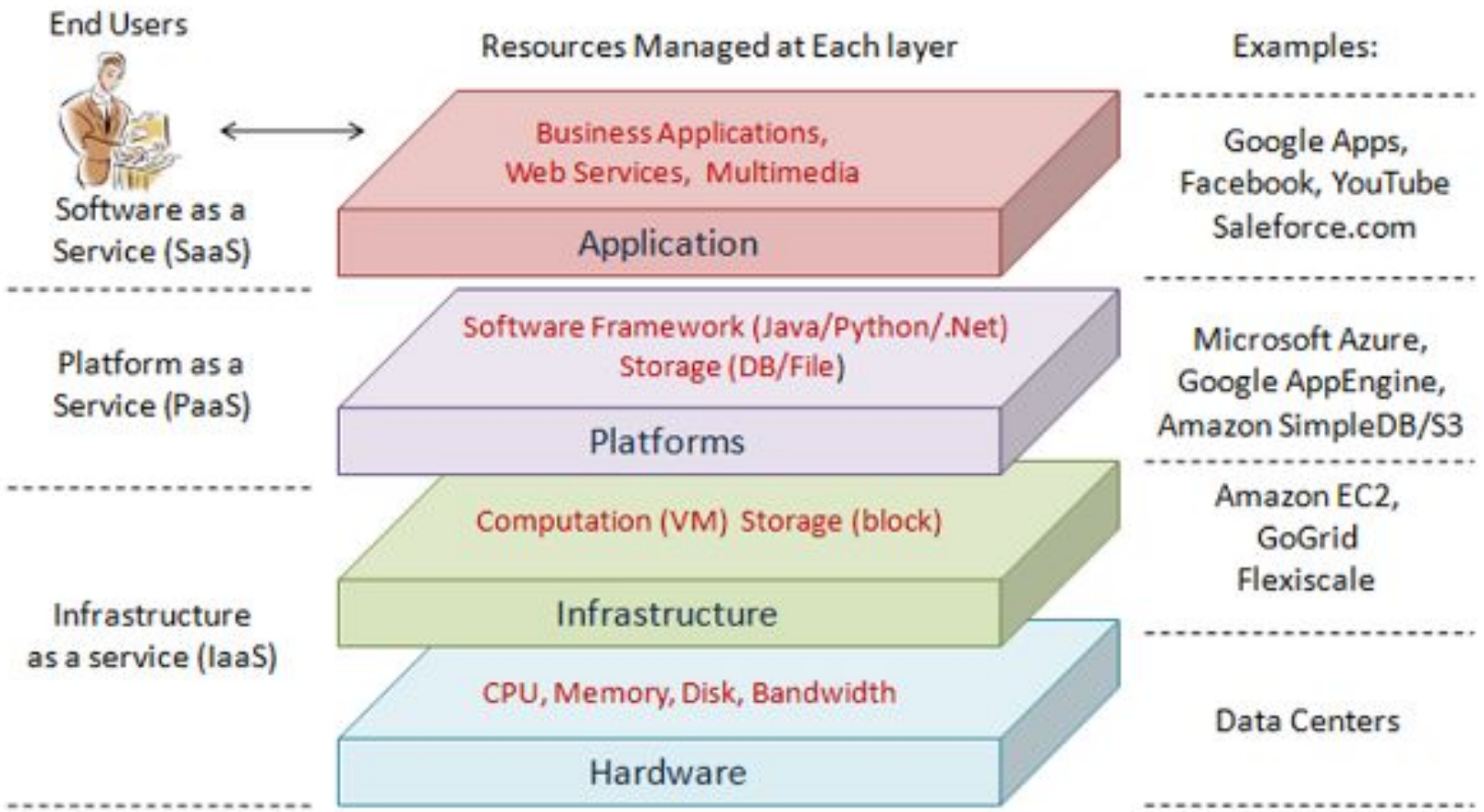
HORIZONTAL SCALING

(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 Service Consumer
Cloud Auditor
Security Audit
Privacy Impact Audit
Performance Audit
Cloud Service Provider
Service Layer
Saas
Paas
Iaas
Resource Abstraction and Control Layer
Physical Resource Layer
Hardware
Facility
Cloud Service Management
Business Support
Provisioning/ Configuration
Portability/ Interoperability
Security
Privacy
Cloud Broker
Service Intermediation
Service Aggregation
Service Arbitrage
Cloud Carrier



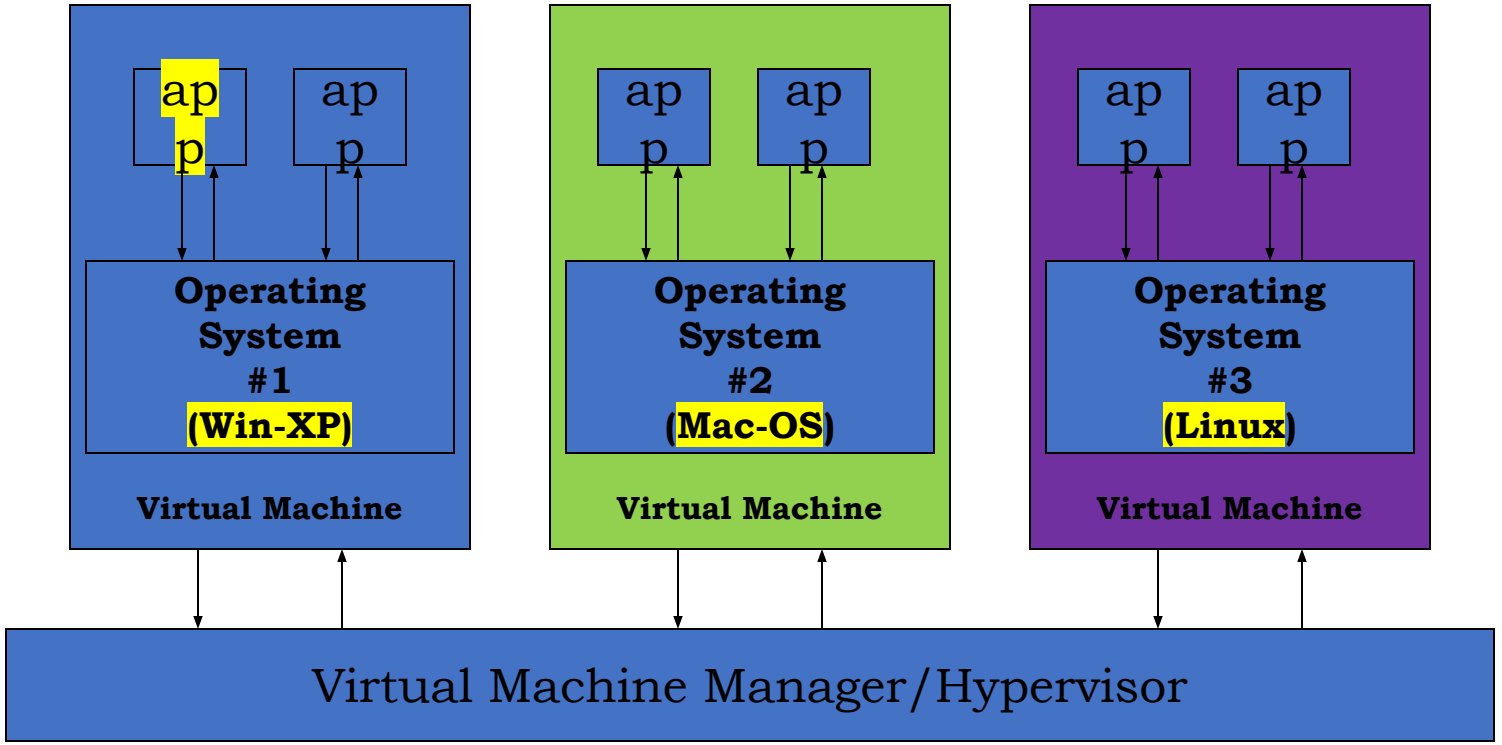
https://link.springer.com/content/pdf/10.1007/s13174-010-0007-6.pdf

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".

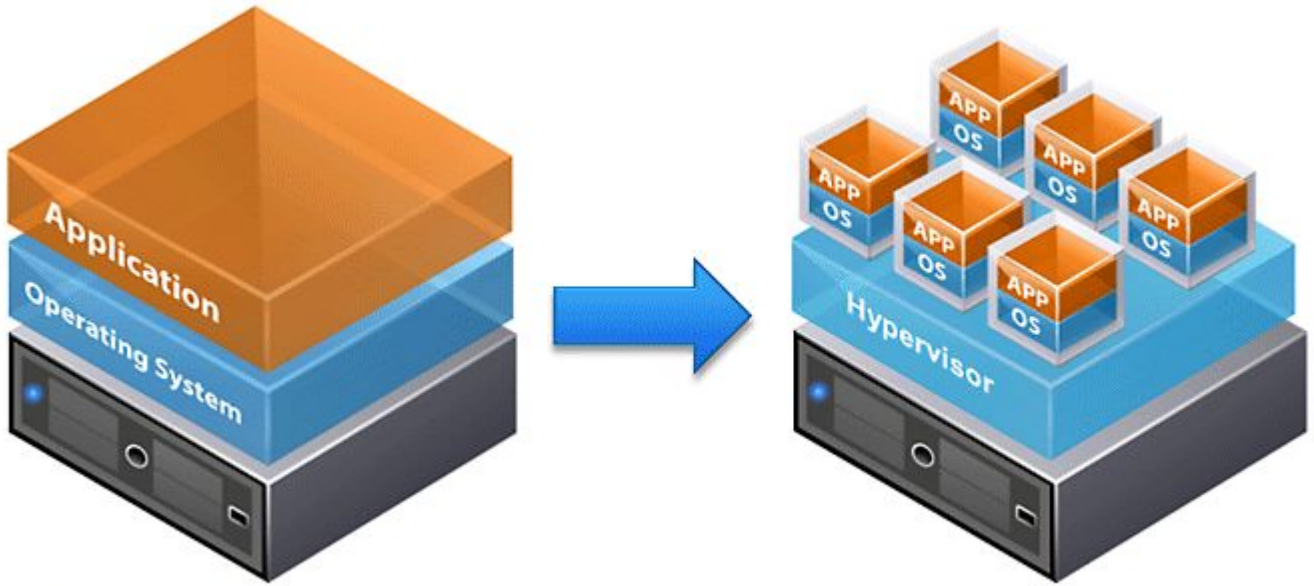
- **"BEING ON OR SIMULATED ON A COMPUTER OR COMPUTER NETWORK "**

What's VT for?



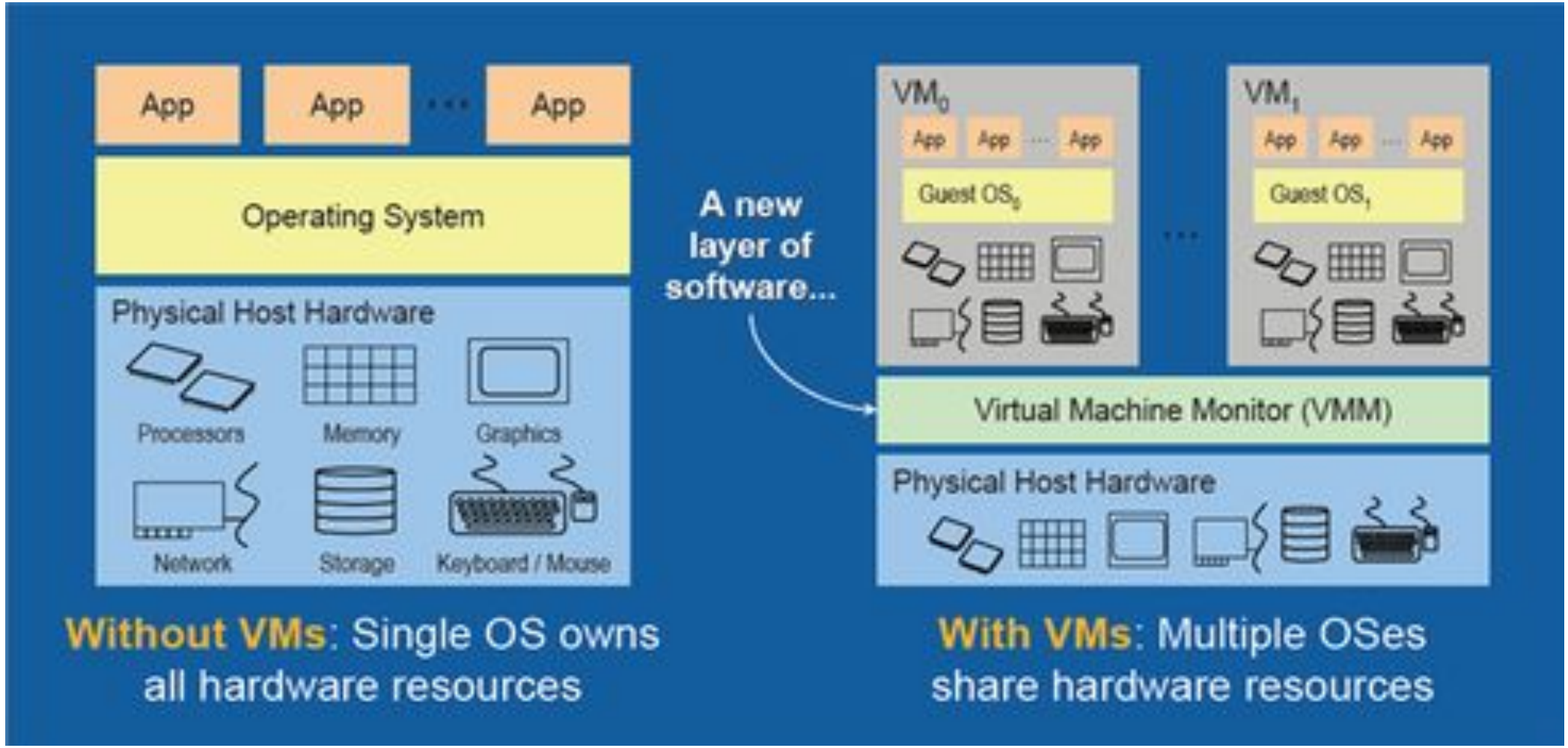
Each operating system was designed to be in total control of the system, which makes it impossible for two or more operating systems to be executing concurrently on the same platform – unless 'total control' is taken away from them by a new layer of control-software: the VMM

- 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.



Traditional Architecture

Virtual Architecture


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

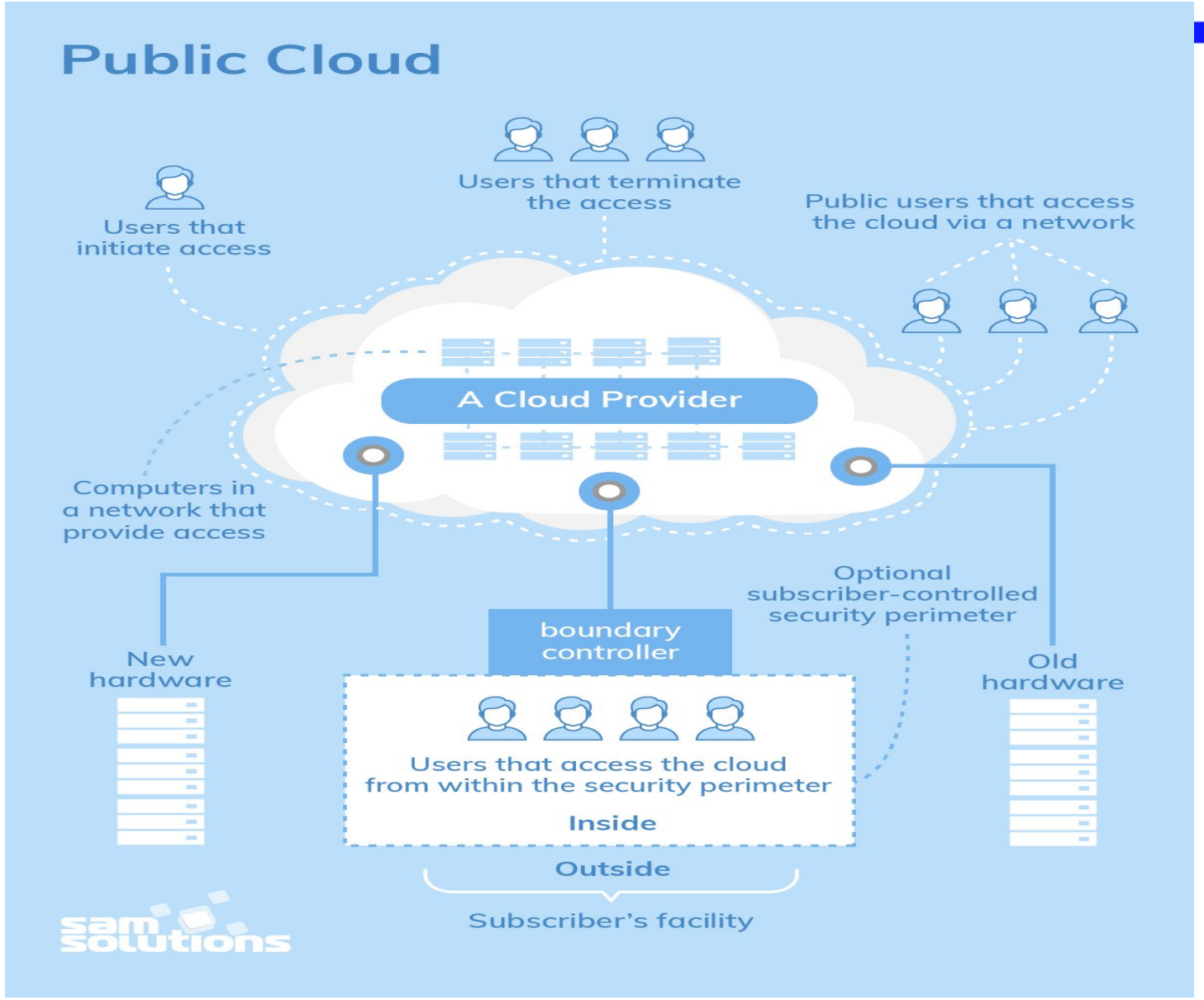
DIFFERENCE	PRIVATE	PUBLIC	HYBRID
Tenancy	Single tenancy: there's only the data of a single organization stored in the cloud.	Multi-tenancy: the data of multiple organizations in stored in a shared environment.	The data stored in the public cloud is usually multi-tenant, which means the data from multiple 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

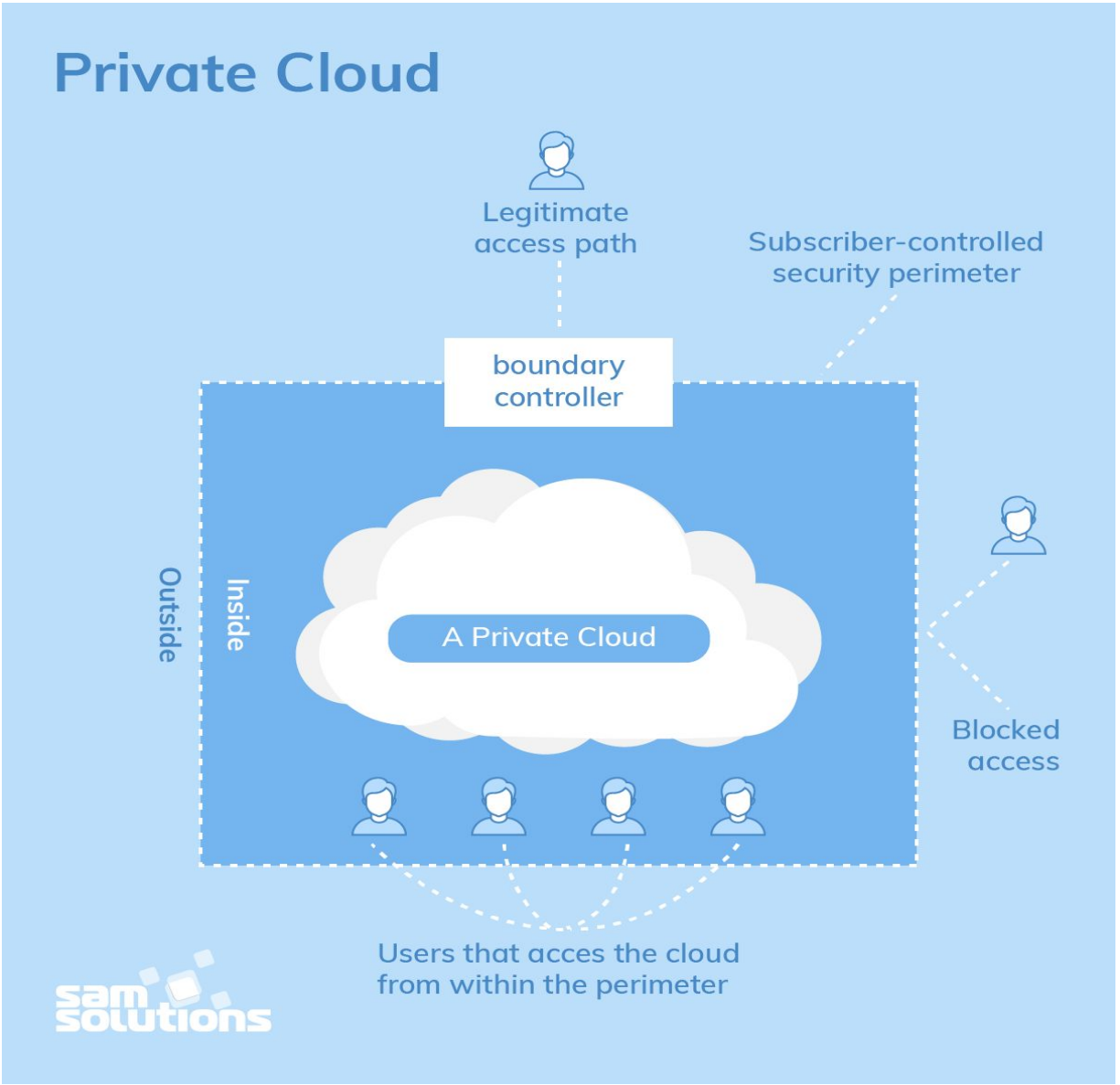
Difference	Private	Public	Hybrid
Cloud Service Management	The organization must have their own administrators managing their private cloud services.	The cloud service provider manages the services, where the organization merely uses them.	The organization itself must manage the private cloud, while the 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

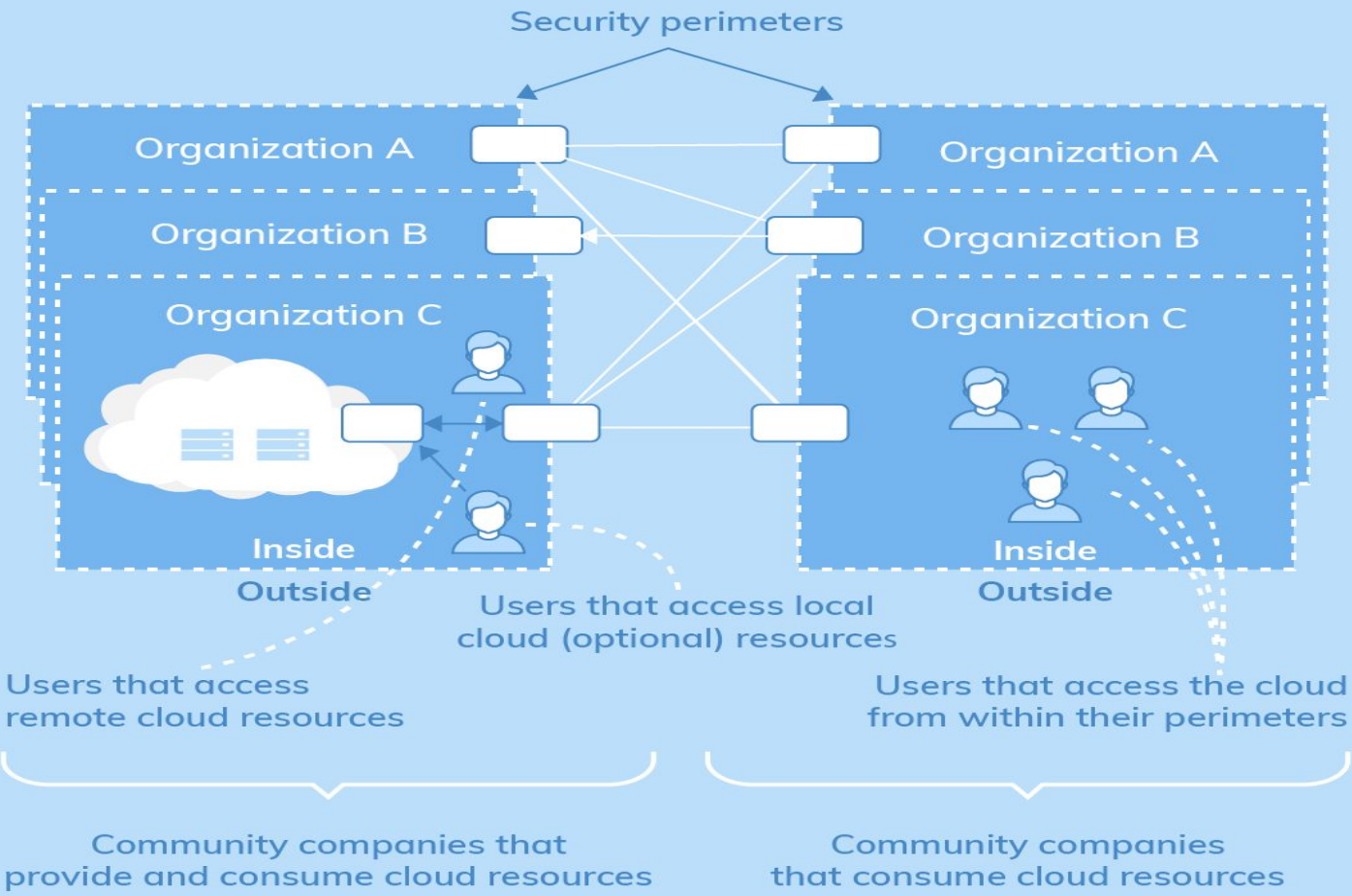
	Pros	Cons	Real time Services Providers
Private	• Control over how a cloud is setup and run • Control over privacy and security practices • Control over the geographical location of data	• Much higher setup and maintenance costs • Less redundancy and resilience (especially if you host the cloud internally) • 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.

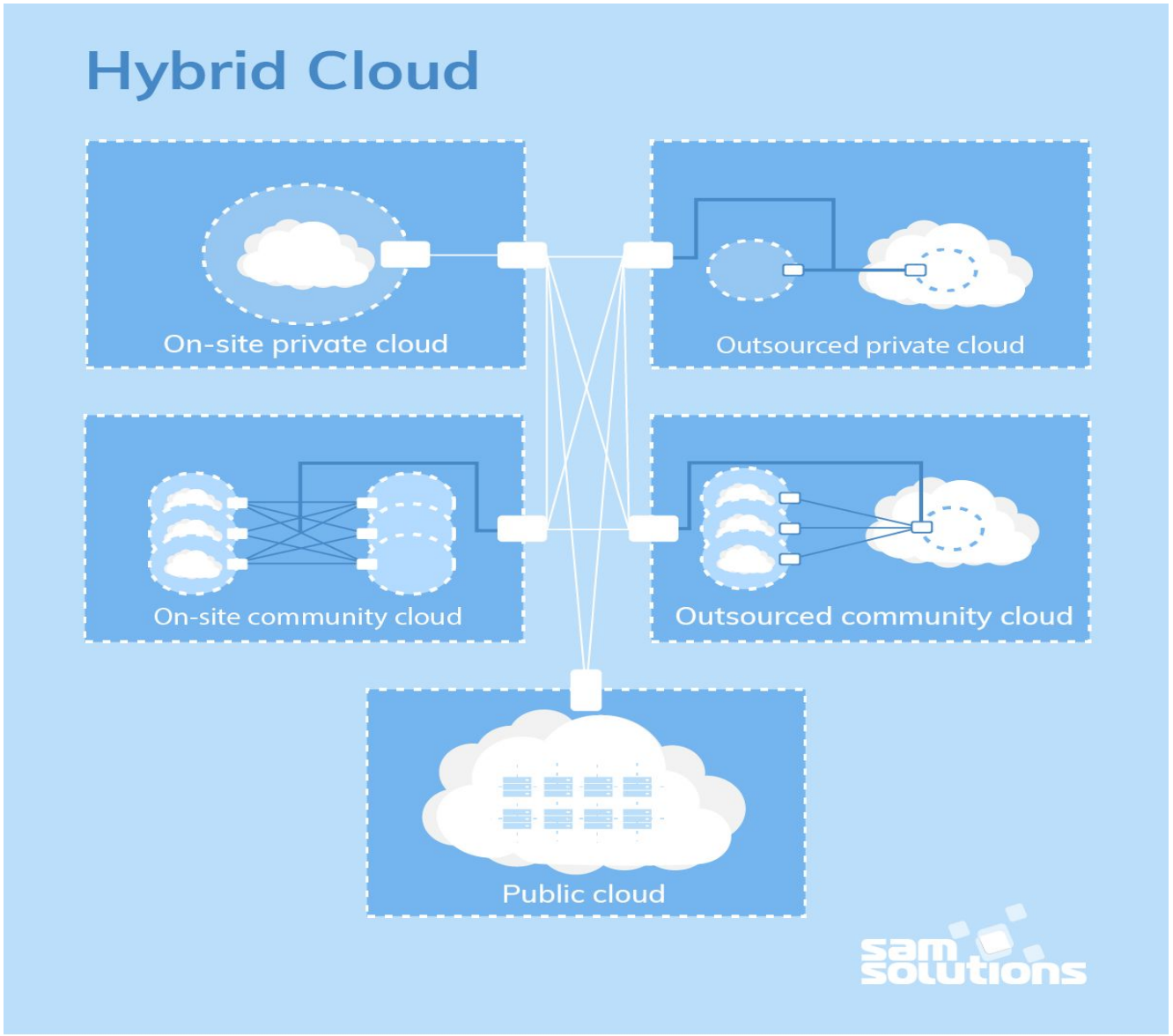

Public Cloud
Users that terminate the access
Users that initiate access
Public users that access the cloud via a network
A Cloud Provider
Computers in a network that provide access
Optional subscriber-controlled security perimeter
boundary controller
New hardware
Old hardware
Users that access the cloud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

Private Cloud



Community Cloud





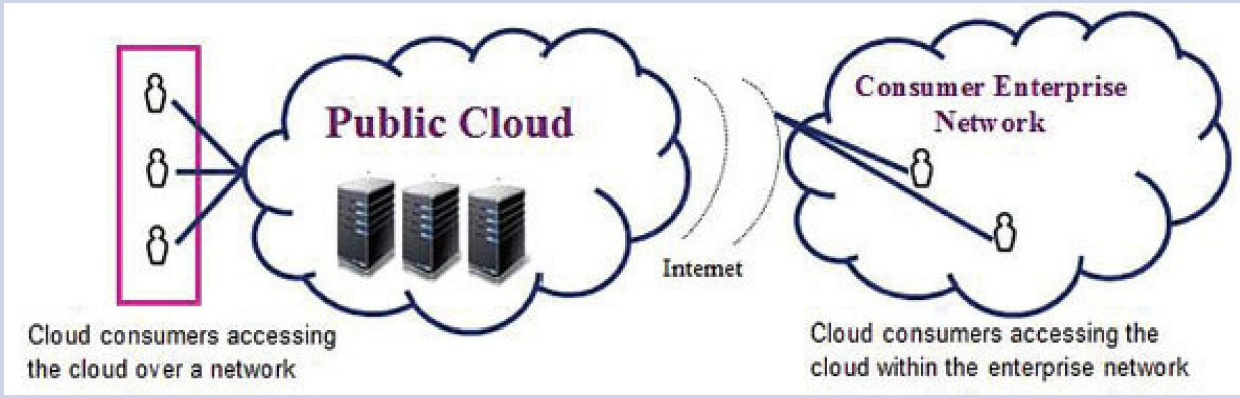
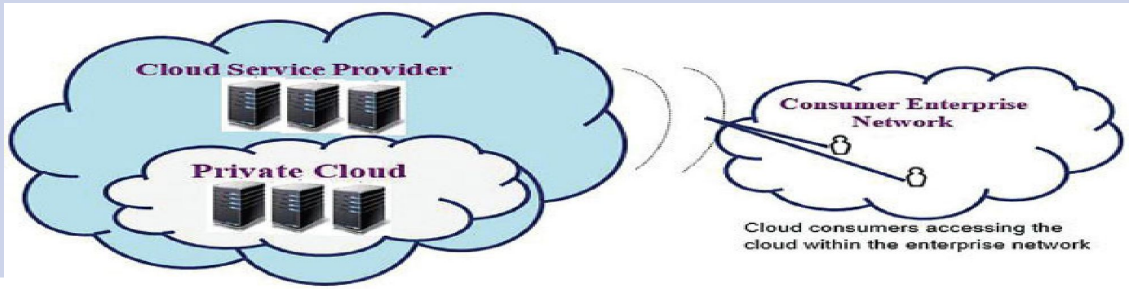
https://www.sam-solutions.com/blog/four-best-cloud-deployment-models-you-need-to-know/

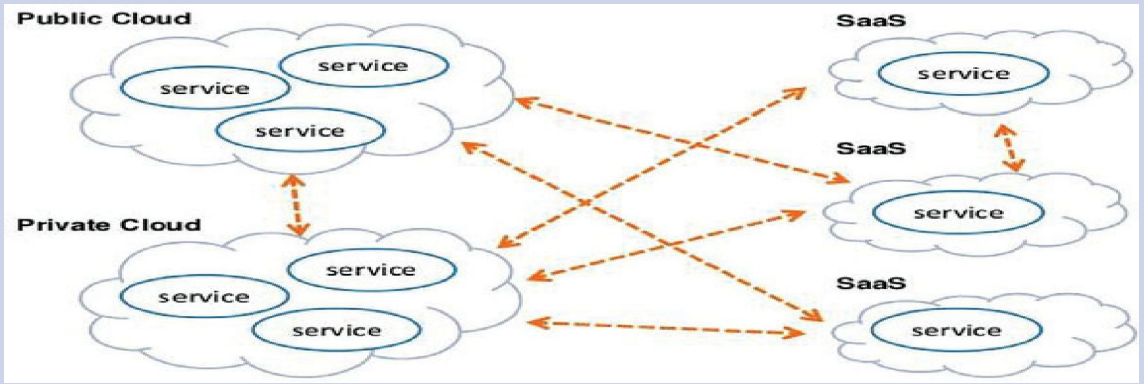
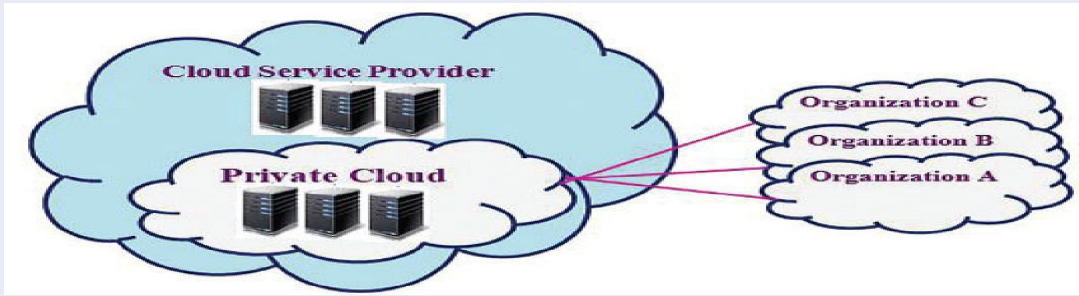
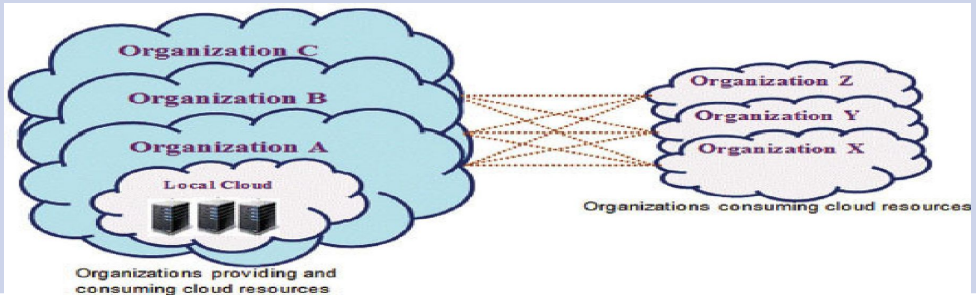
Some Real time deployment models

PUBLIC CLOUD	PRIVATE CLOUD	HYBRID CLOUD	COMMUNITY CLOUD
• Netflix • *Netflix used relational databases in its data centers.* • Netflix migrated its functioning, content, and delivery network to Amazon's public cloud — AWS	• 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.	• *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	• 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	

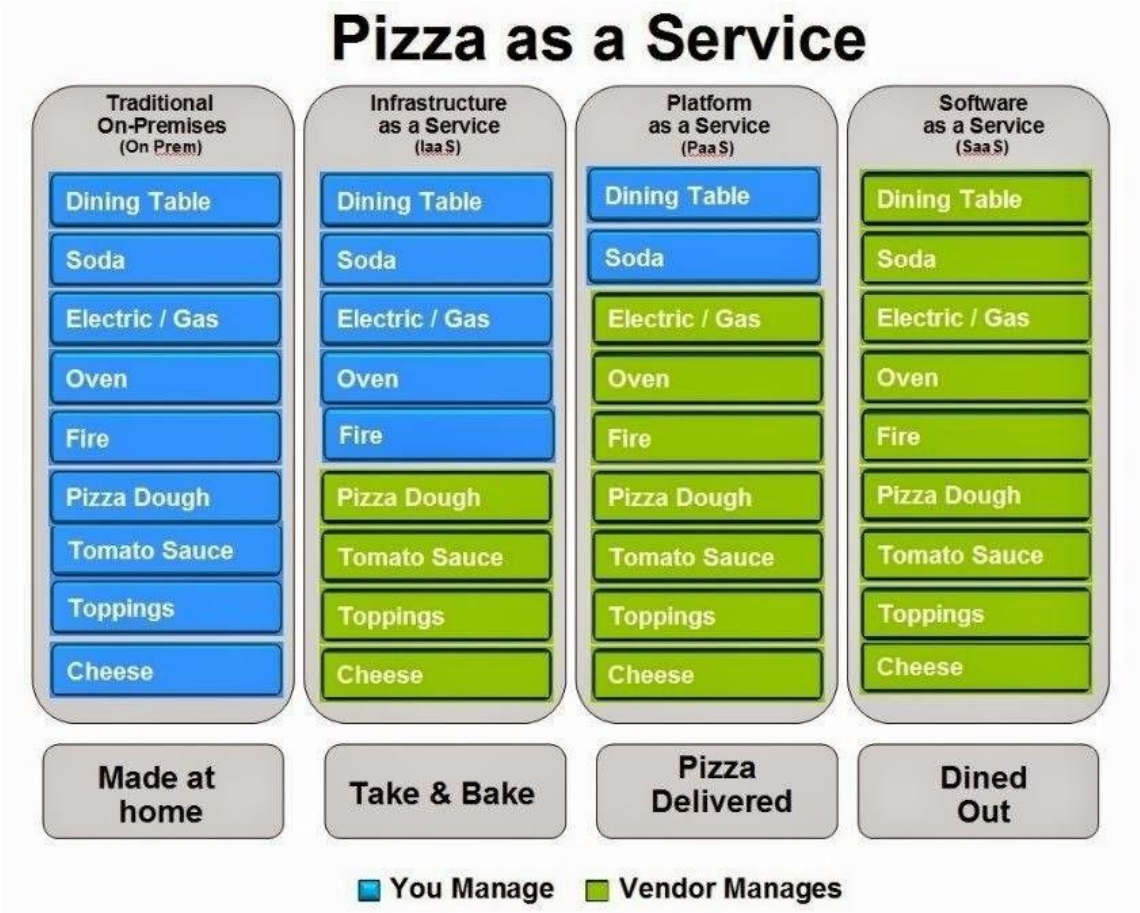
Cloud	Design
Hybrid cloud	
Out-sourced community cloud	
On-site community cloud	

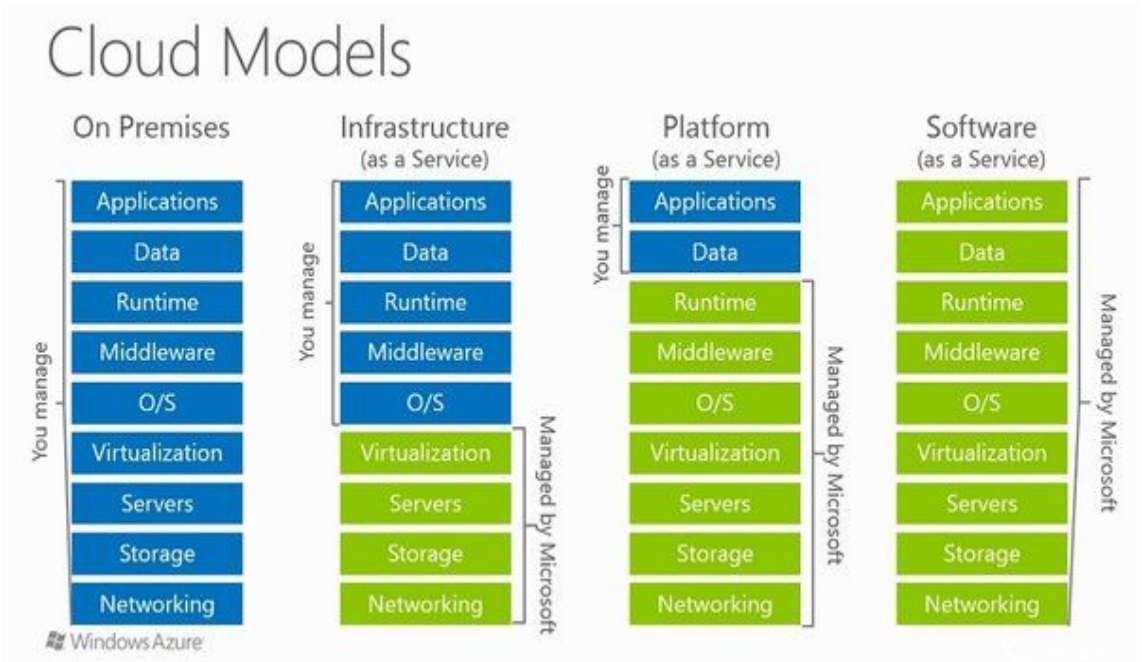
Attribute	Public	Private	Community	Hybrid
Ease of setup and use	Easy	Requires proficiency in IT	Requires proficiency in IT	Requires proficiency in IT
Data privacy and security	Low	High	Relatively high	High
Data control	Little to none	High	Relatively high	Relatively high
Reliability	Vulnerable	High	Relatively 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

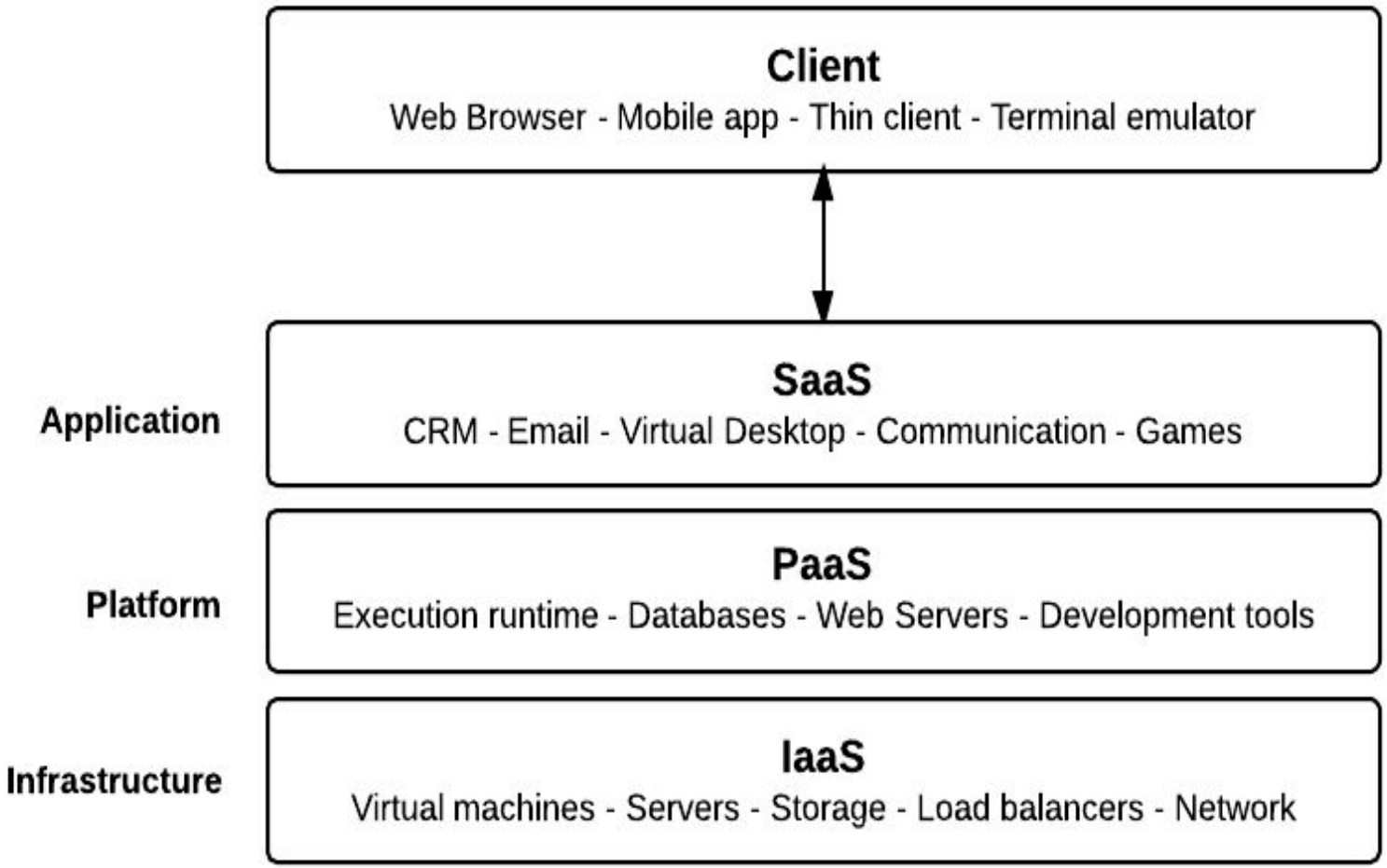
Attribute	Public	Private	Community	Hybrid
Ongoing costs	High	Low	Medium	Medium
Security	Low	High	Medium	Medium
Compliance	Low	High	Medium	Medium
Quality of service	Low	High	Medium	Medium
Integration	Low	High	Medium	Medium
Configurability	Low	Medium	Medium	Medium

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

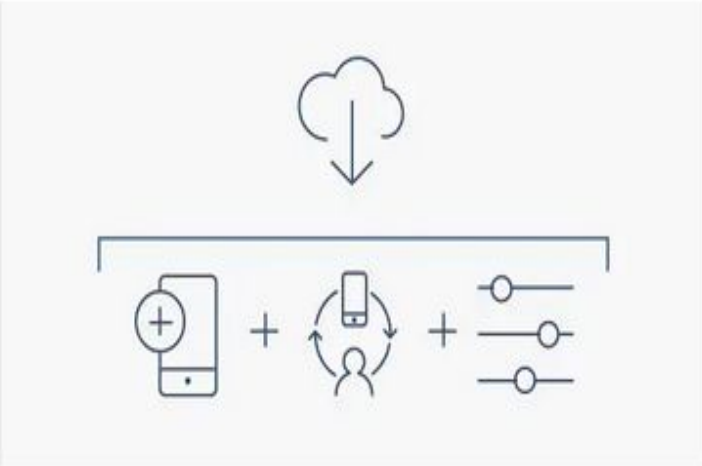

Pizza as a Service
Traditional On-Premises (On Prem)
Infrastructure as a Service (IaaS)
Platform as a Service (PaaS)
Software as a Service (SaaS)
Dining Table
Soda
Electric / Gas
Oven
Fire
Pizza Dough
Tomato Sauce
Toppings
Cheese
Made at home
Take & Bake
Pizza Delivered
Dined Out
You Manage
Vendor Manages


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

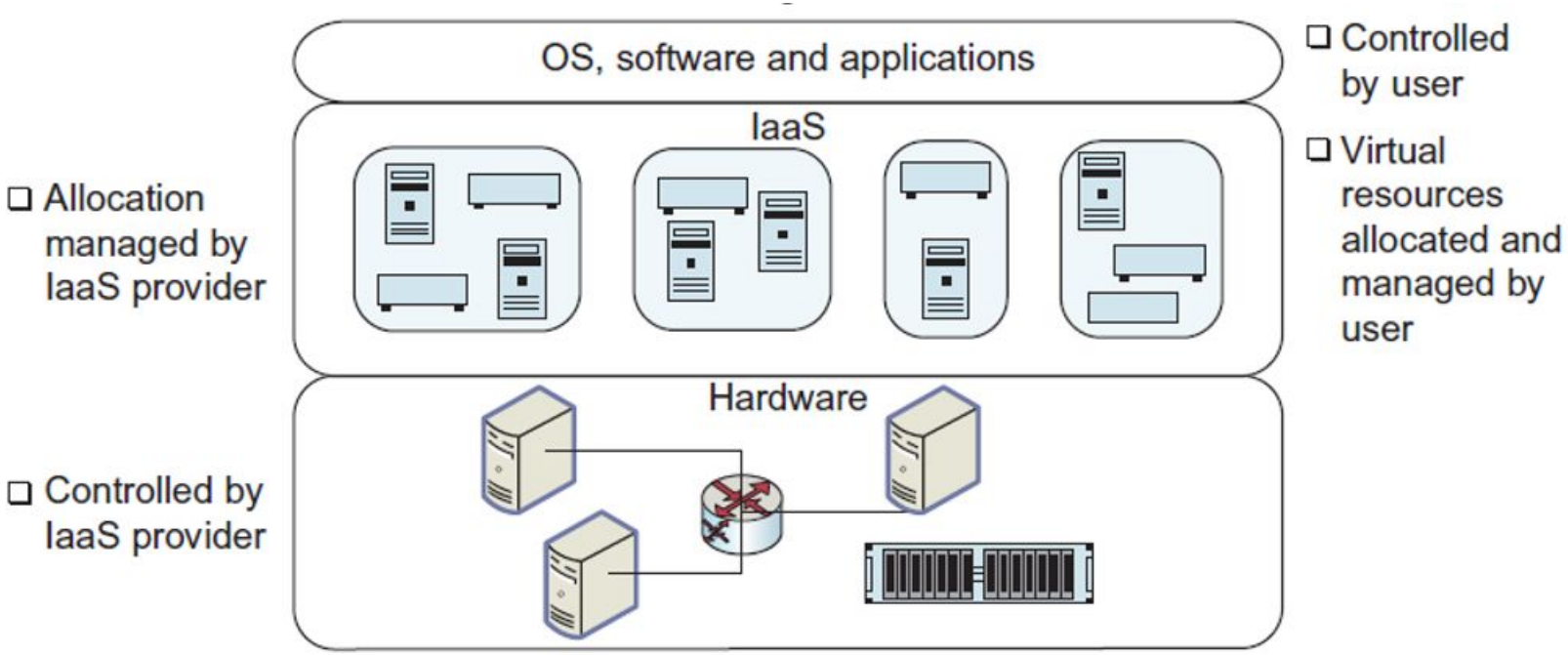
A service provider offers access to a cloud-based environment in which users can build and deliver applications. The provider supplies underlying infrastructure.

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)

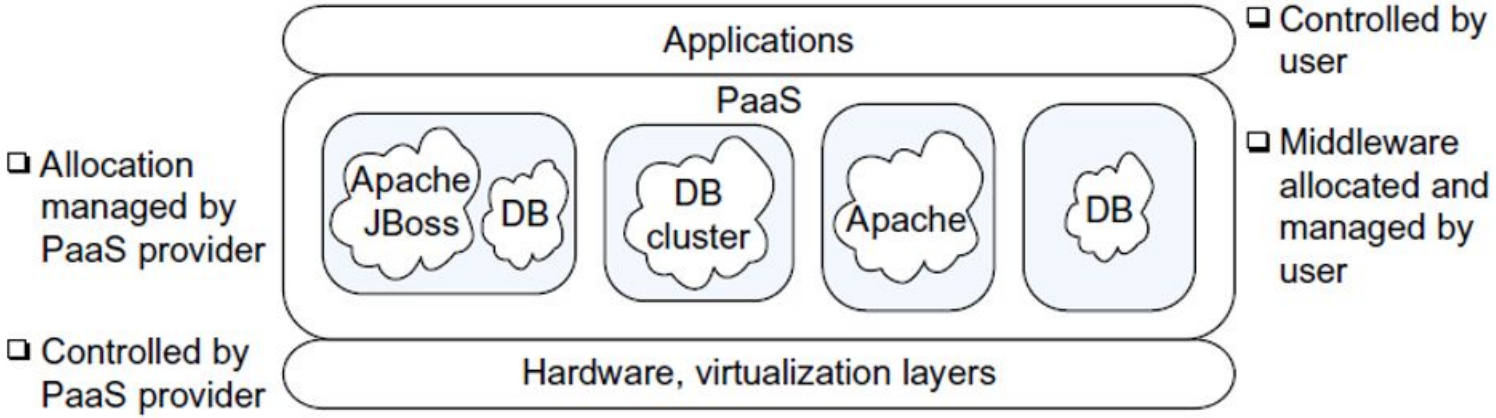
Infrastructure-as-a-Service (IaaS)



Platform as a Service (PaaS)

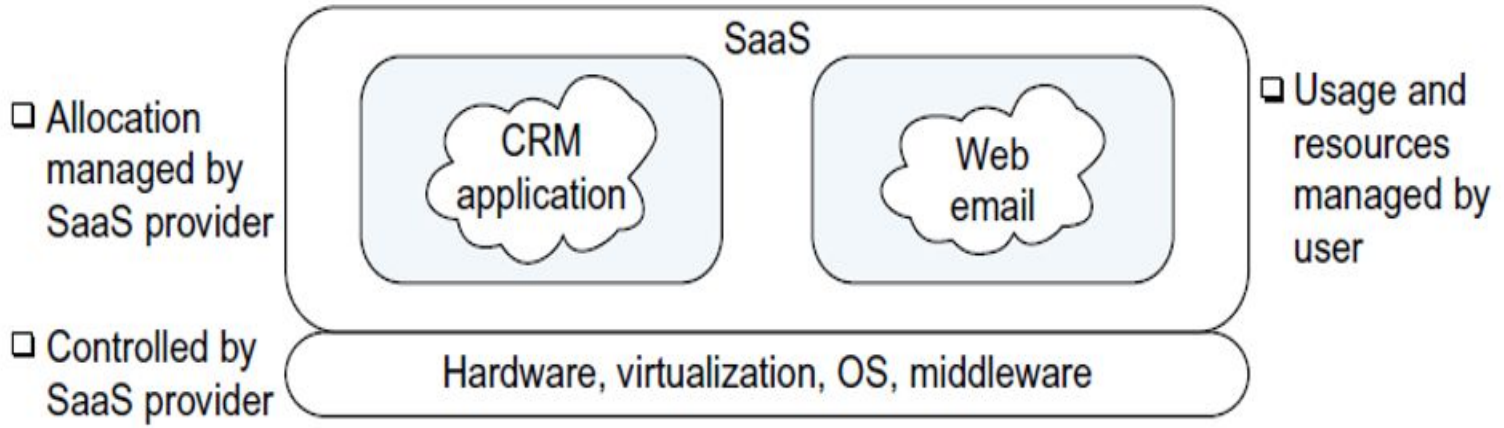
Definition	Characteristics	Services	Service Providers
• PaaS cloud computing platform is created for the programmer to develop, test, run, and manage the applications	• Accessible to various users via the same development application. • Integrates with web services and databases. • Builds on **virtualization 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.

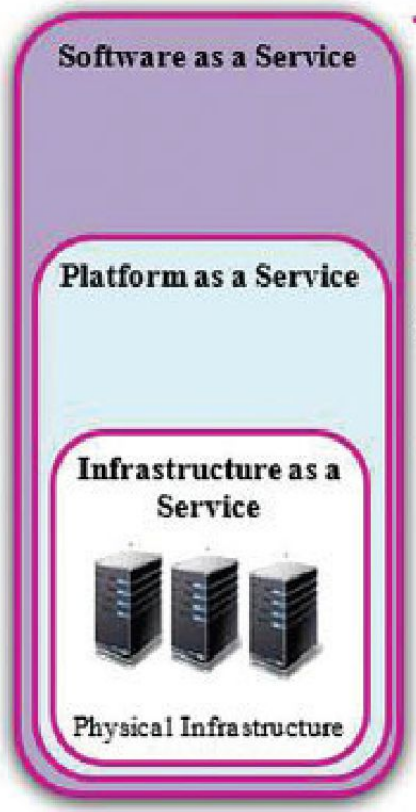
Platform as a Service (PaaS)



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

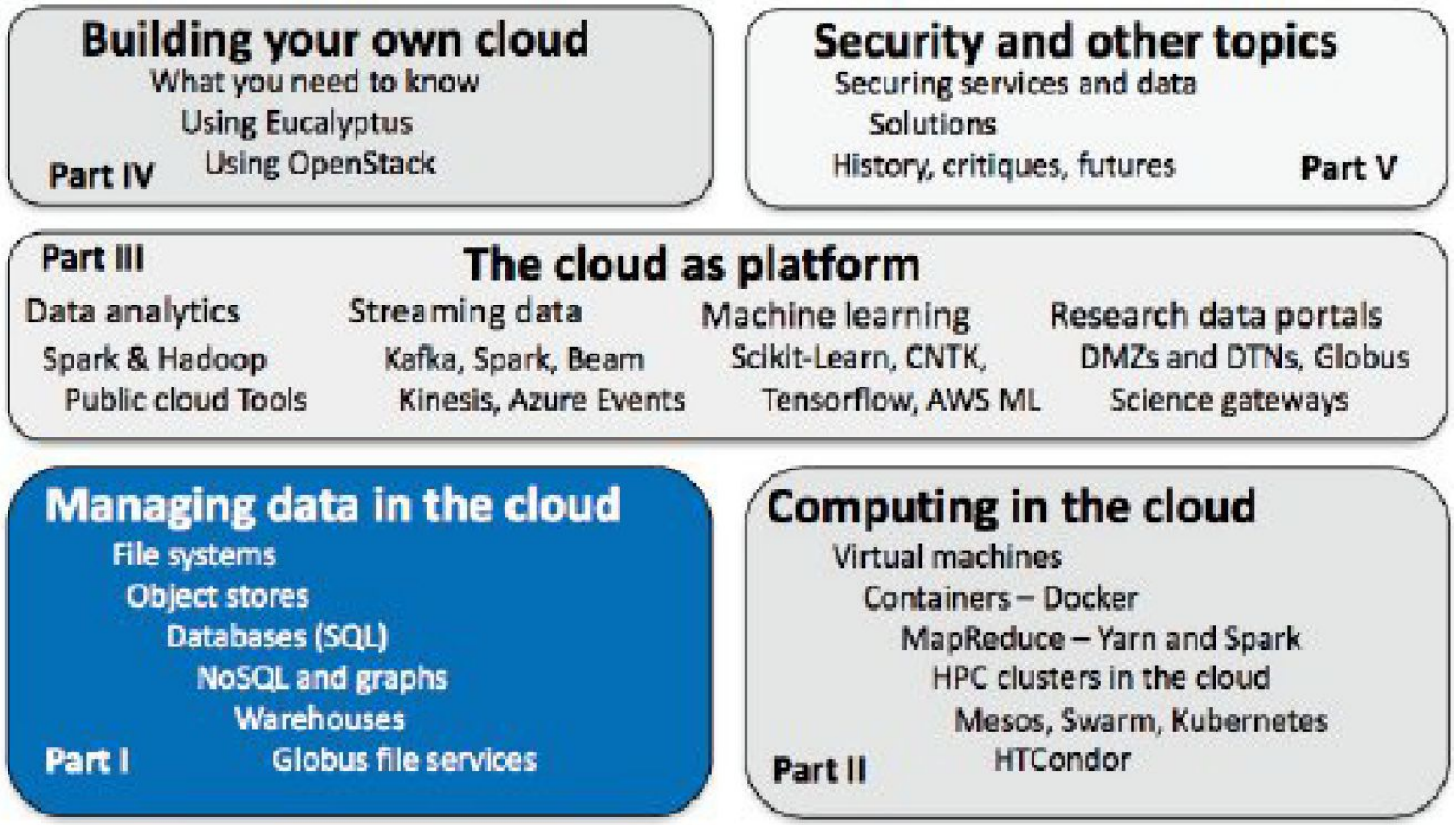
Software as a Service (SaaS)





What	Who
On-Demand access to any application	End user (does not care about hw or sw)
Platform for building and delivering web applications	Developer (no managing of the underlying hardware & software layers)
Raw Computers Infrastructure	System Administrator (complete management of the computer infrastructure)

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

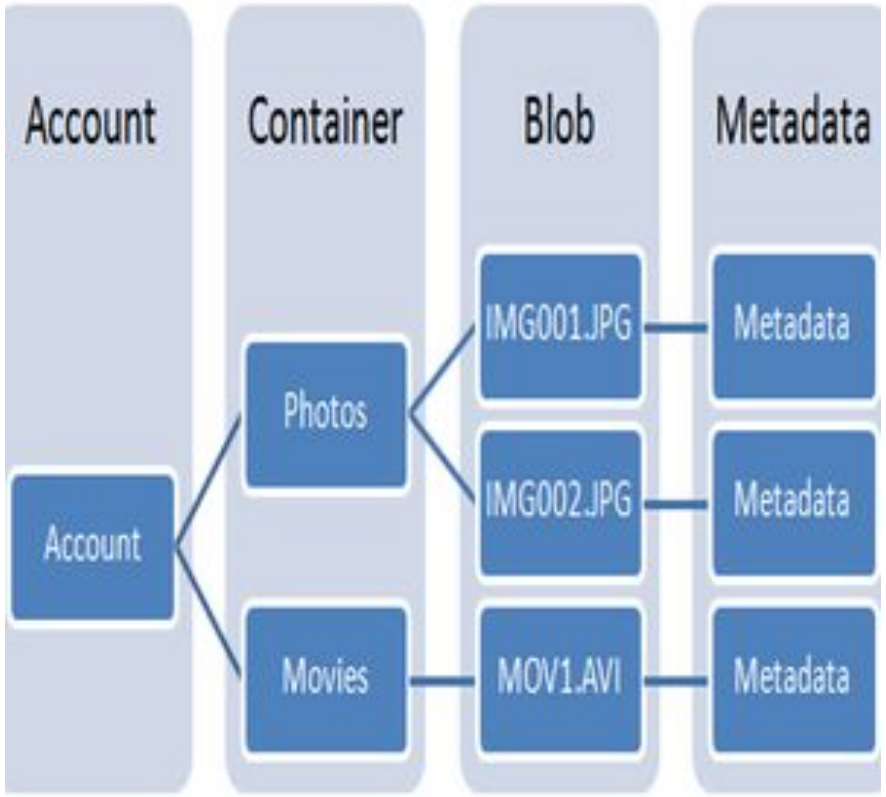
- IaaS platforms for storage as a service and then compute as a service.
- Storage as a Service (sometimes abbreviated as SaaS) takes a detailed look at key Amazon Storage Services:
  - **Amazon Simple Storage Service (S3)**, which provides a highly reliable and highly available object store over HTTP.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

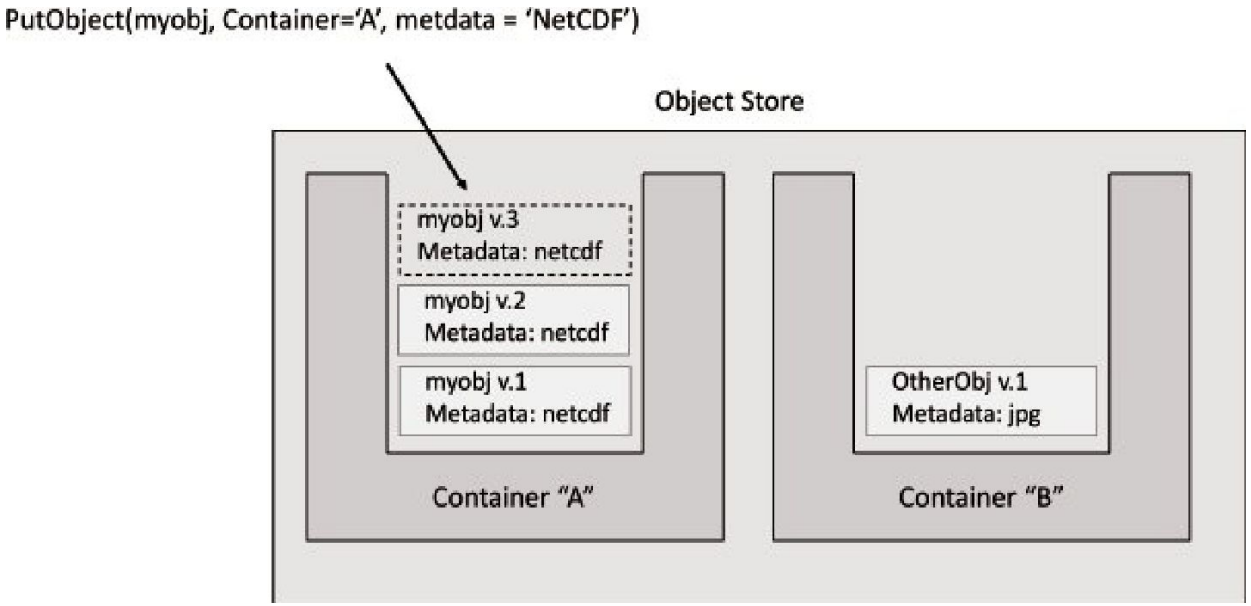
File Systems

- 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.

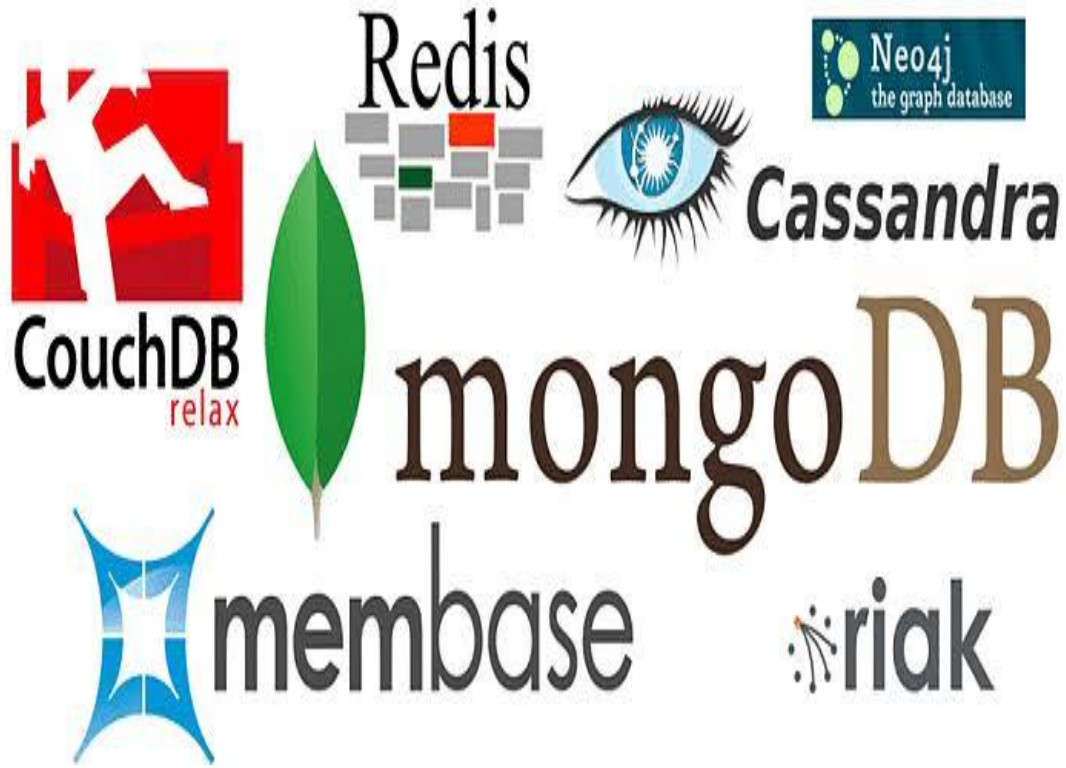
Types of No SQl data base

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

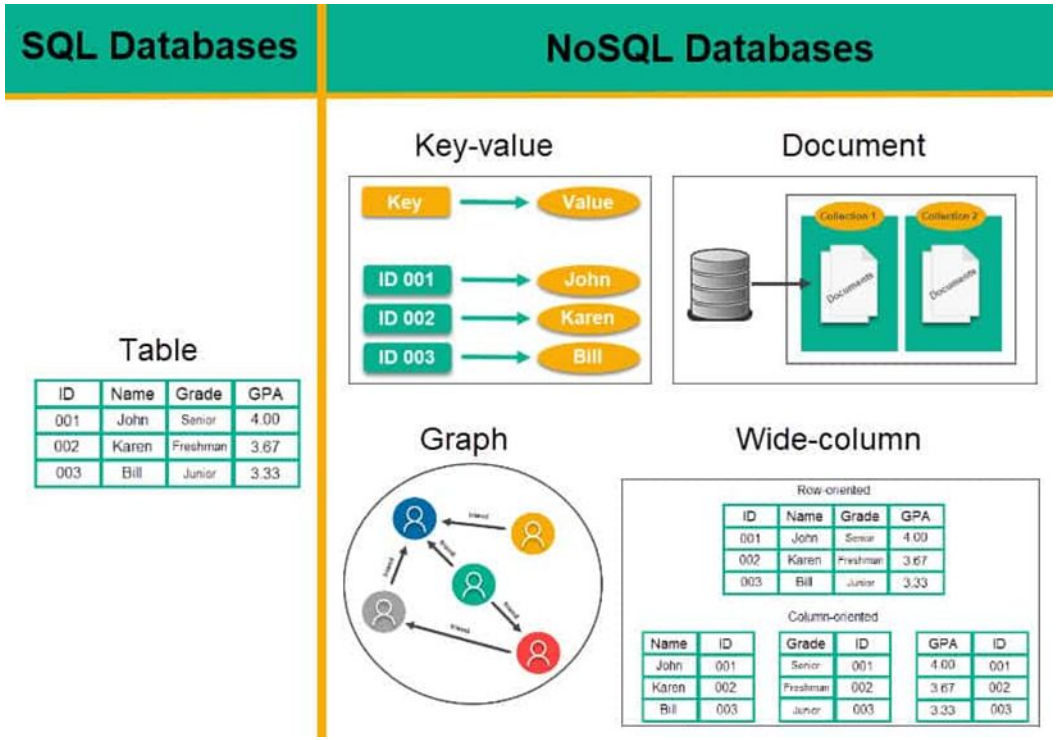
NonRelational

EXAMPLES

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



NoSQL Databases

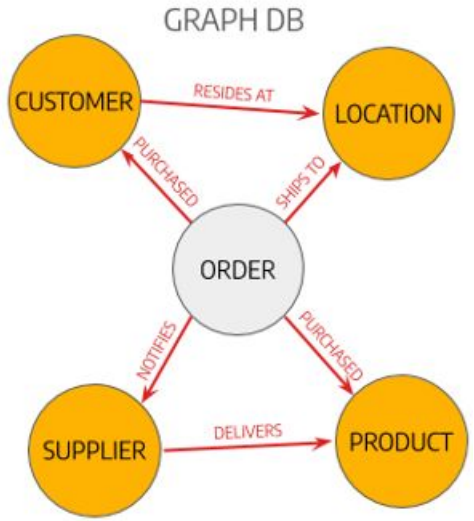


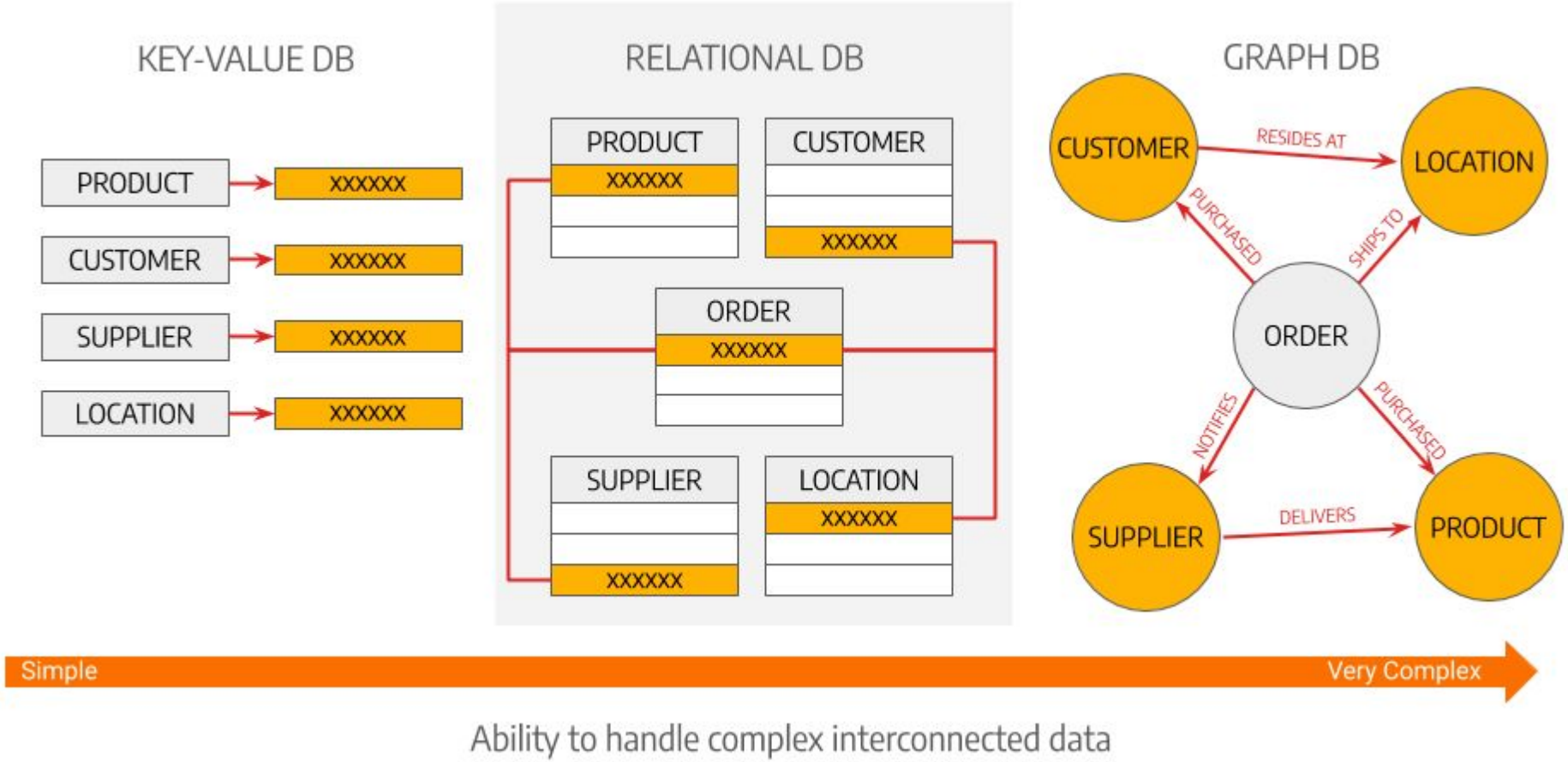
Challenges of scale: The CAP theorem

- *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

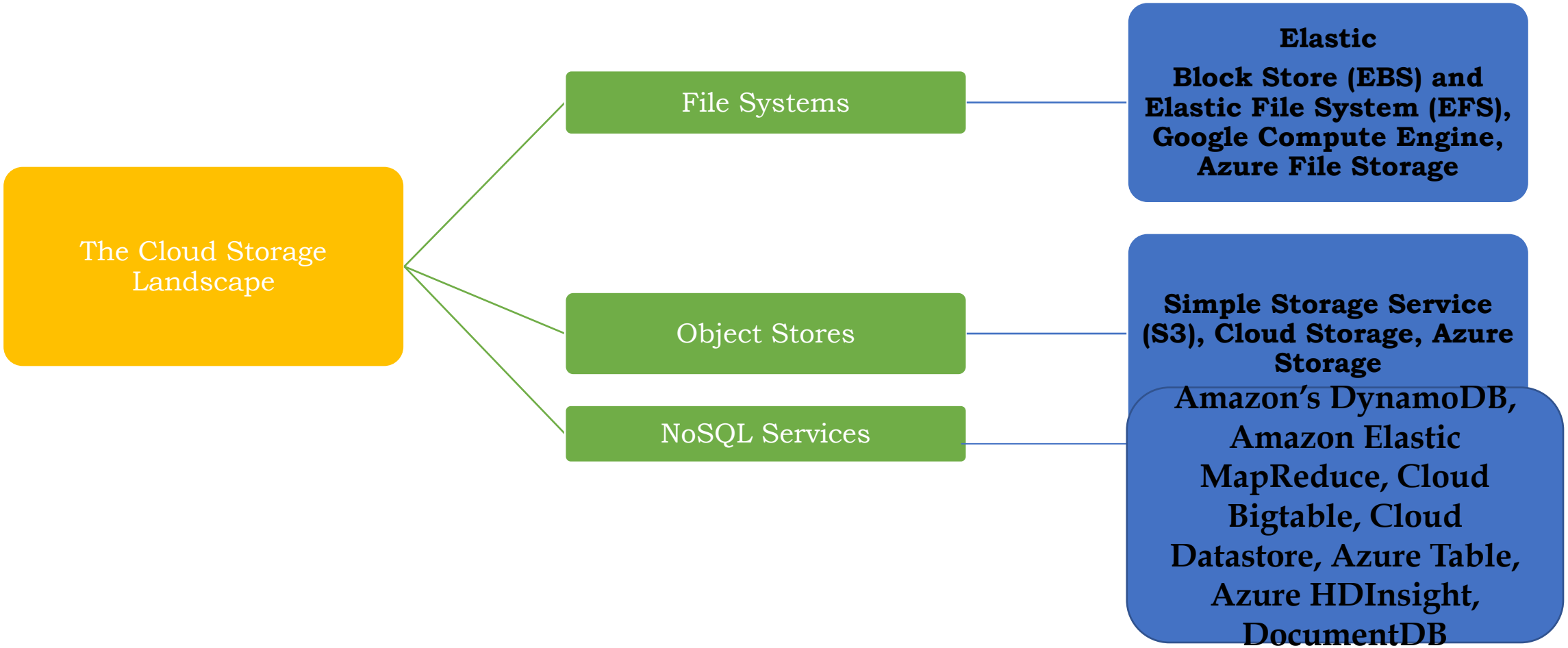



KEY-VALUE DB
PRODUCT
XXXXXX
CUSTOMER
XXXXXX
SUPPLIER
XXXXXX
LOCATION
XXXXXX
RELATIONAL DB
PRODUCT
XXXXXX
CUSTOMER
XXXXXX
ORDER
XXXXXX
SUPPLIER
XXXXXX
LOCATION
XXXXXX
GRAPH DB
CUSTOMER
RESIDES AT
LOCATION
PURCHASED
SHIPS TO
ORDER
NOTIFIES
PURCHASED
SUPPLIER
DELIVERS
PRODUCT
Simple
Very Complex
Ability to handle complex interconnected data

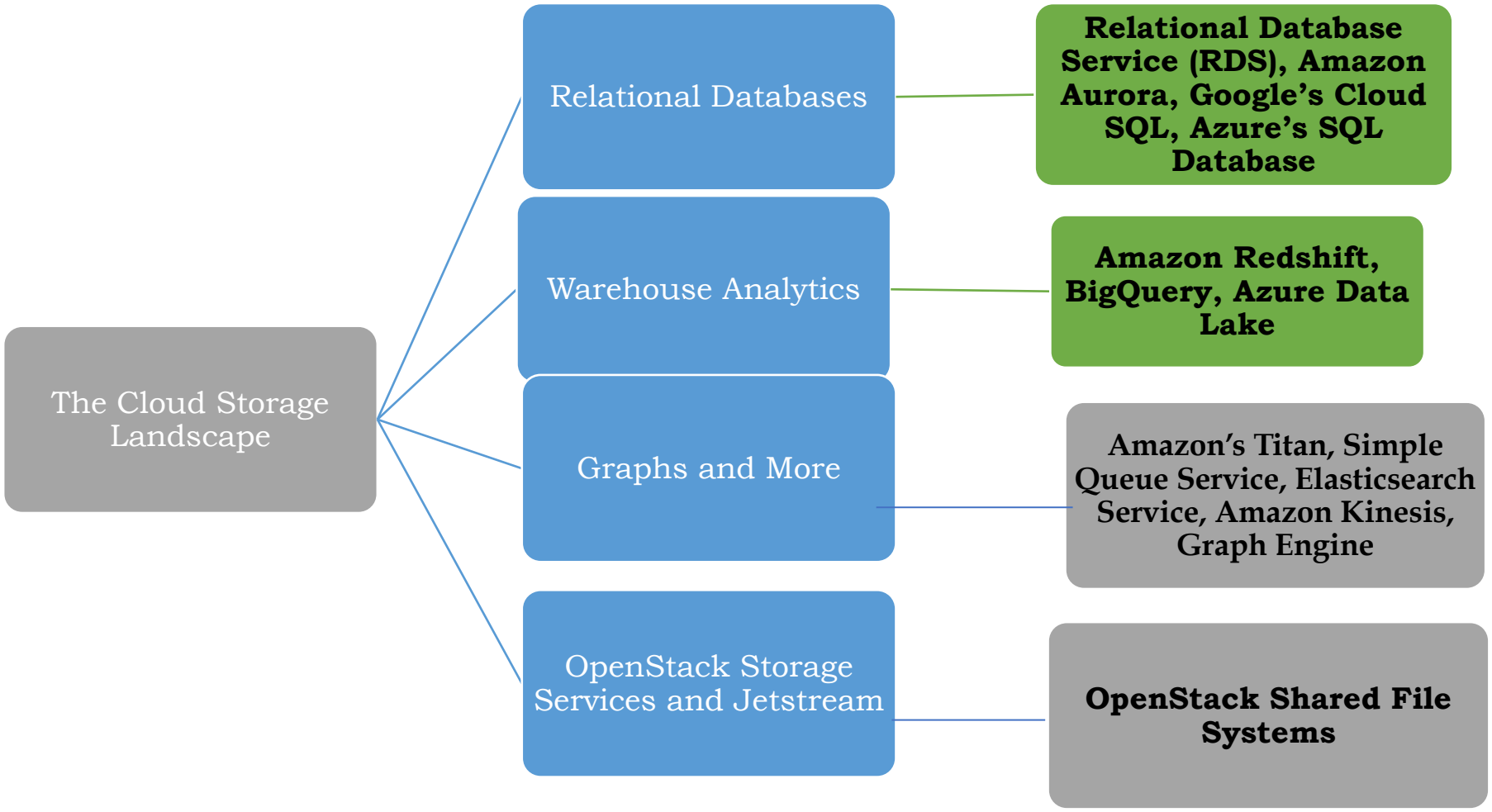
Data Warehouses

- 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

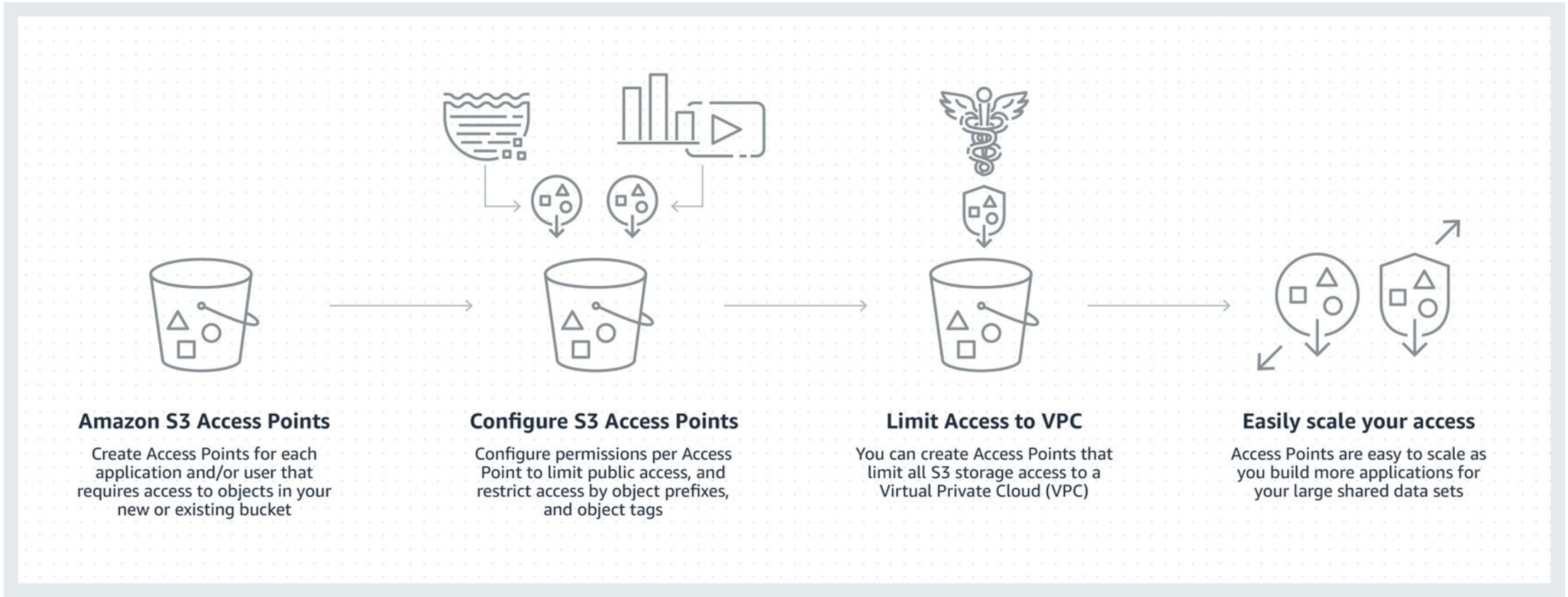
- File systems (also referred to as file shares) are virtual data drives that can be attached to virtual machines
  - ***Amazon's Elastic Block Store (EBS) and Elastic File System (EFS)***
- EBS provides a **BLOCK STORAGE SERVICE FOR EC2**.
- It is possible to request an EBS disk volume of a particular 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

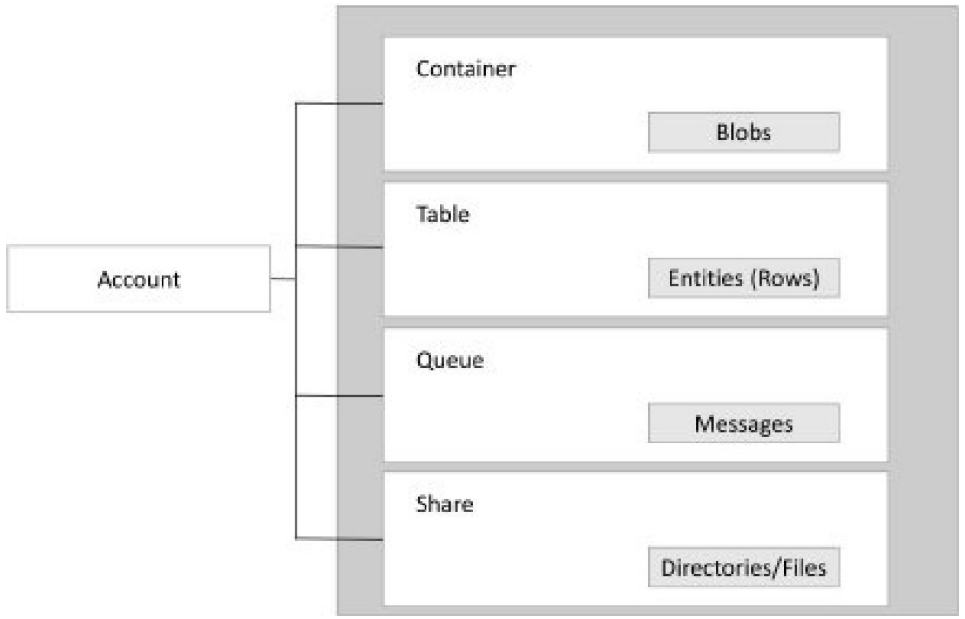
You can create Access Points that limit all S3 storage access to a Virtual Private Cloud (VPC)

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

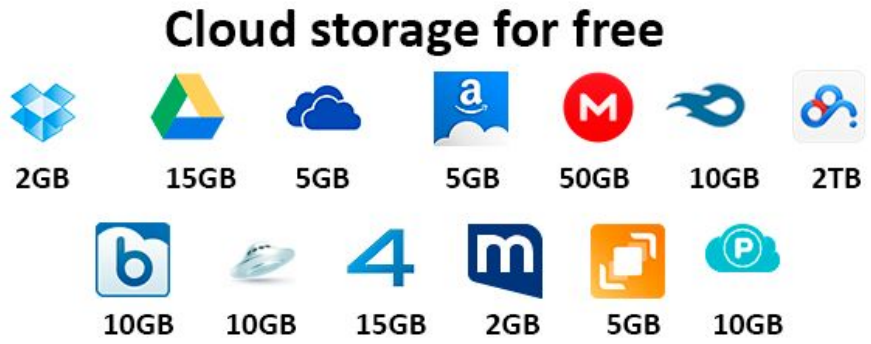
- Messaging services allow applications to send and receive messages using what are referred to as **publish/subscribe semantics.**
- **AMAZON'S TITAN** extension to DynamoDB supports graph databases.
- **Elasticsearch** open-source search and analytics engine
- **Amazon Kinesis** supports analysis of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

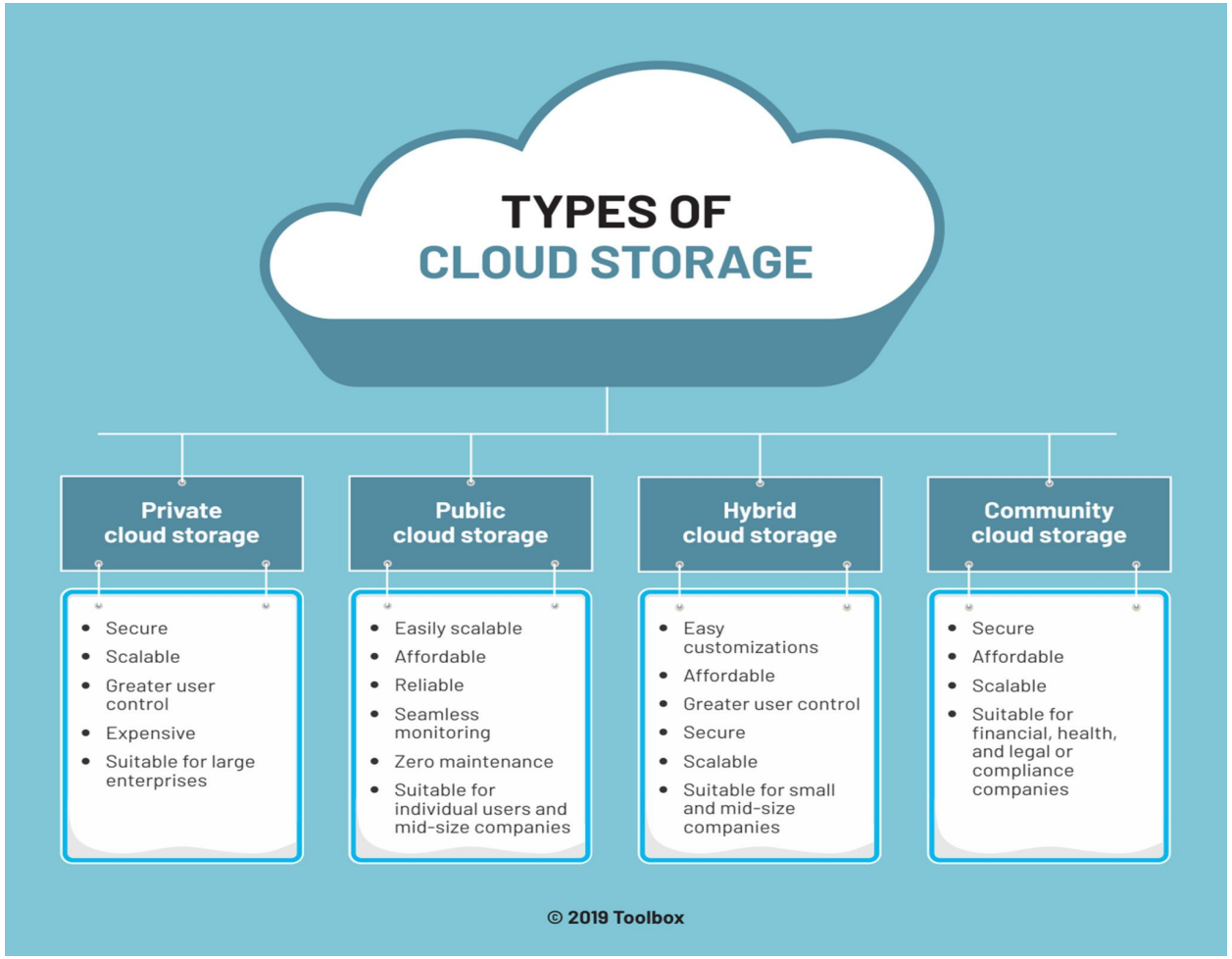
OpenStack Storage Services and Jetstream

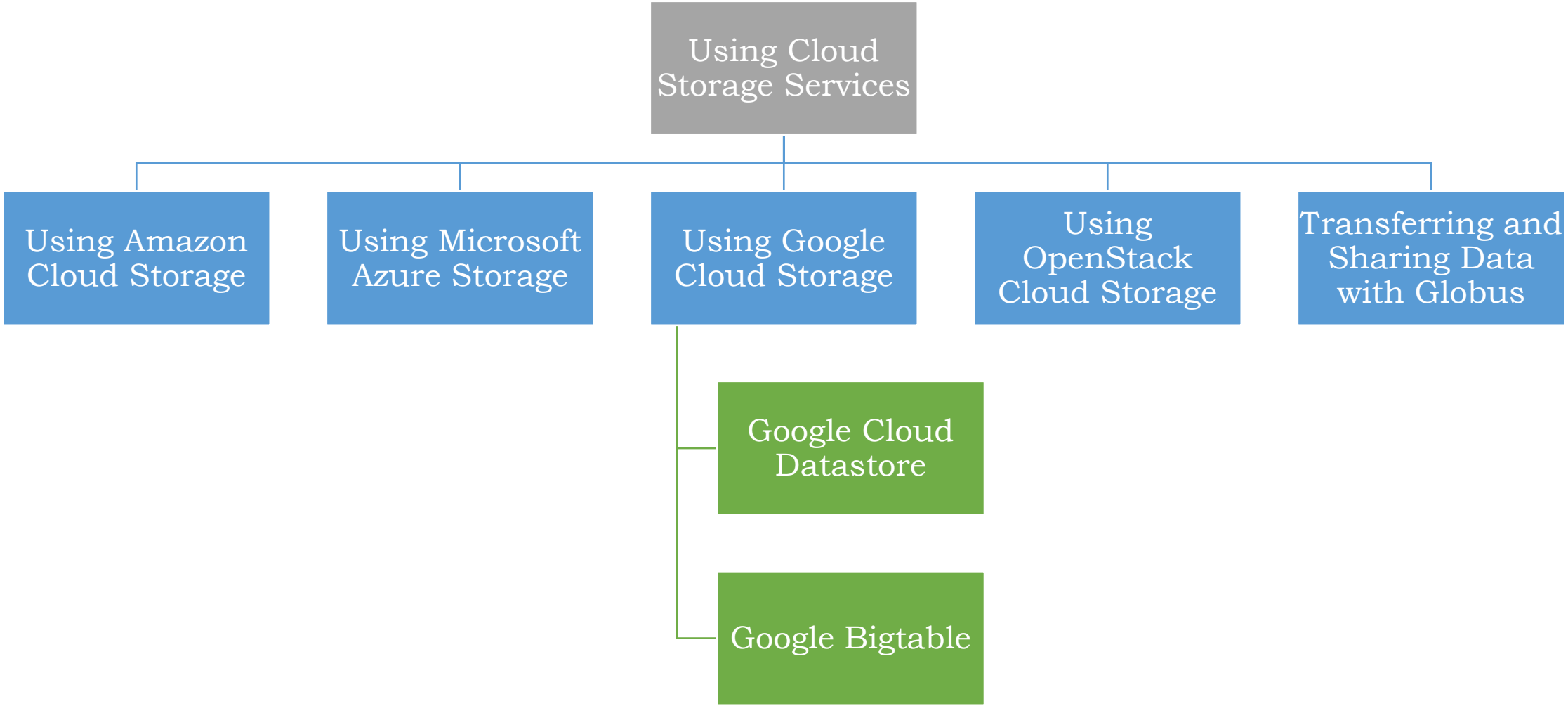
- The OpenStack open source cloud software supports only a few standard storage services***: object storage, block storage, and file system storage.***
- The OpenStack object storage service is called **Swift**.
- The OpenStack Shared File Systems service, like the **Amazon EFS and Azure File service**, 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
Using OpenStack Cloud Storage
Transferring and Sharing Data with Globus
Google Cloud Datastore
Google Bigtable

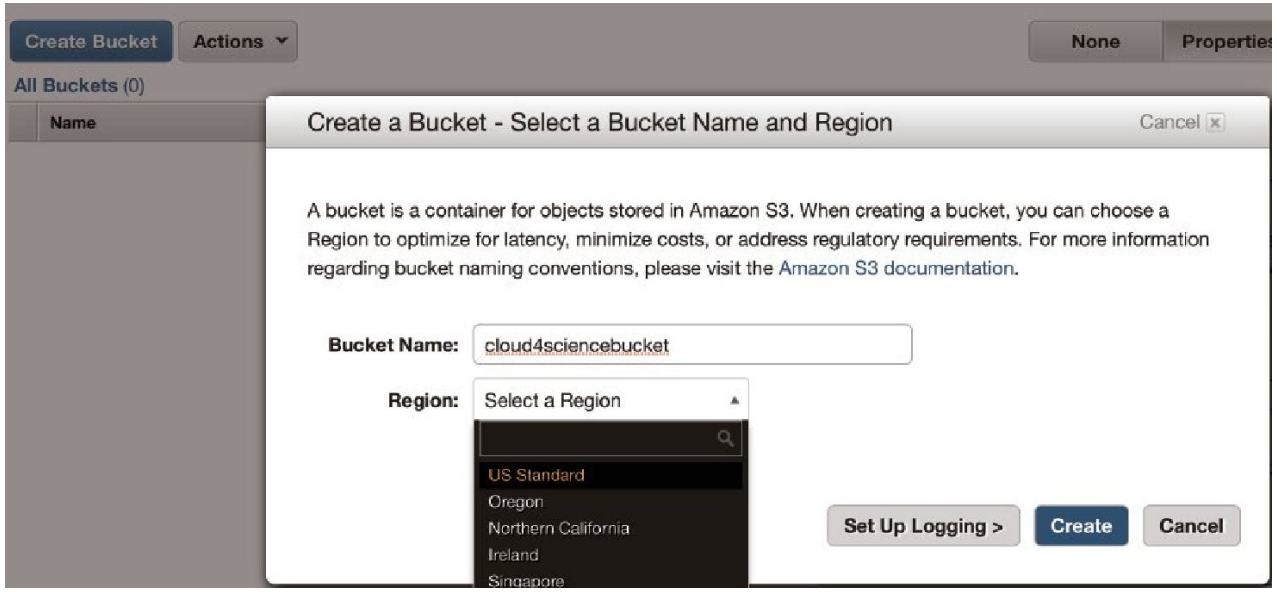
Two Access Methods: Portals and APIs

- 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

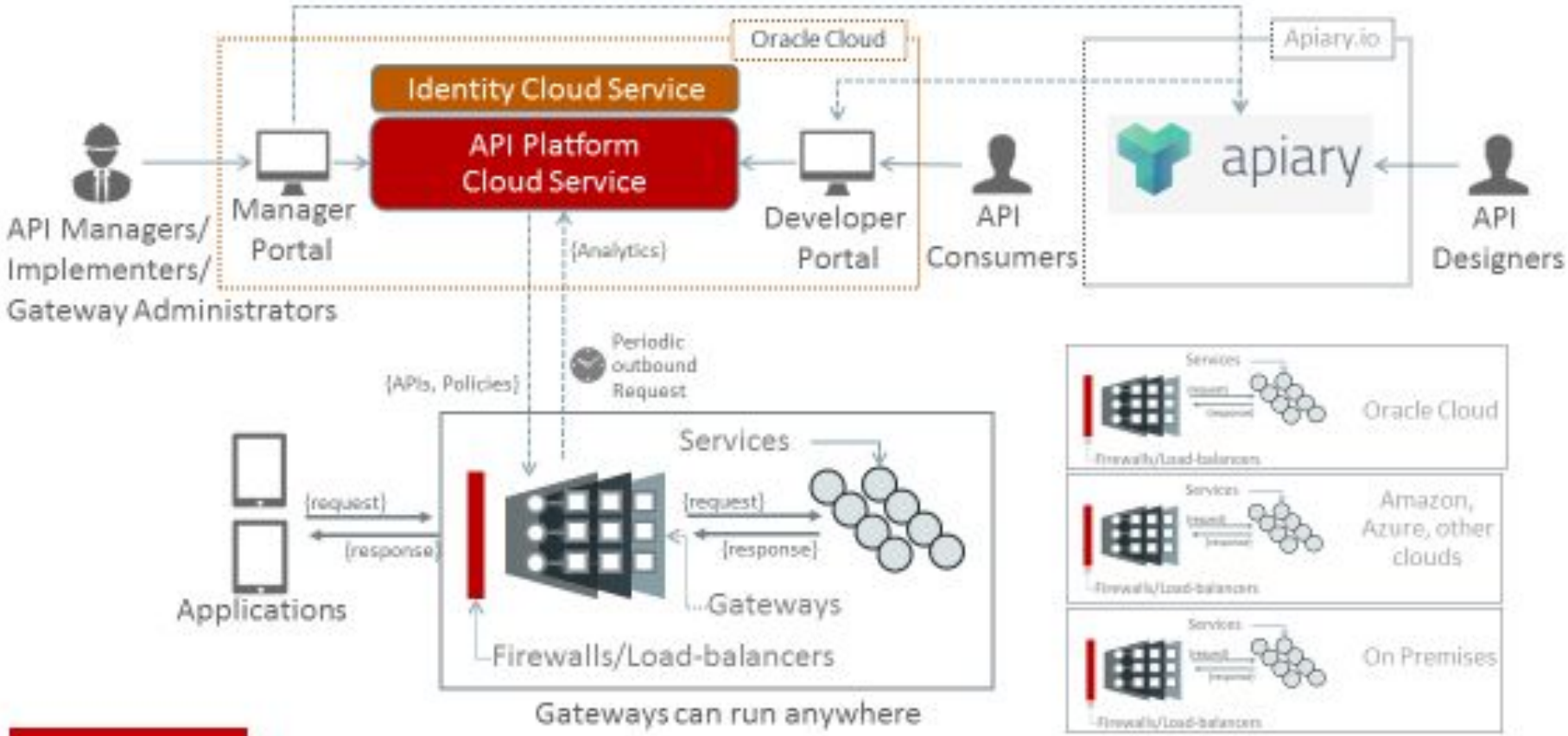
- **Representational State Transfer** (REST) application programming interface (API) that permits requests to be transmitted via the secure Hypertext Transfer Protocol (HTTPS) that is used by web browsers.
- **LOCAL AND CLOUD-HOSTED APPLICATIONS**

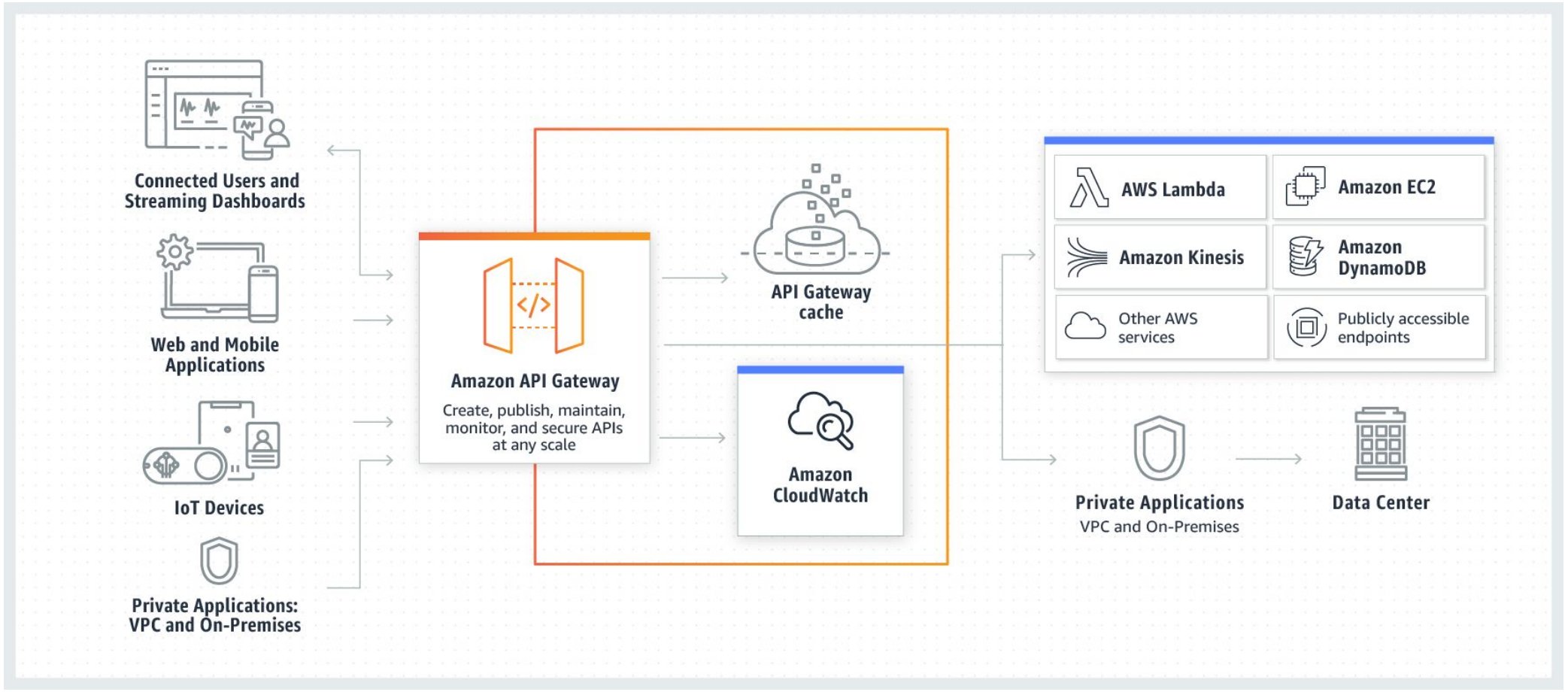
```
PUT / HTTP/1.1
Host: cloud4sciencebucket.s3.amazonaws.com
Content-Length: length
Date: date
Authorization: authorization string
<CreateBucketConfiguration
              xmlns="http://s3.amazonaws.com/doc
              /2006-03-01/">
  <LocationConstraint>US
  Standard</LocationConstraint>
  </CreateBucketConfiguration>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

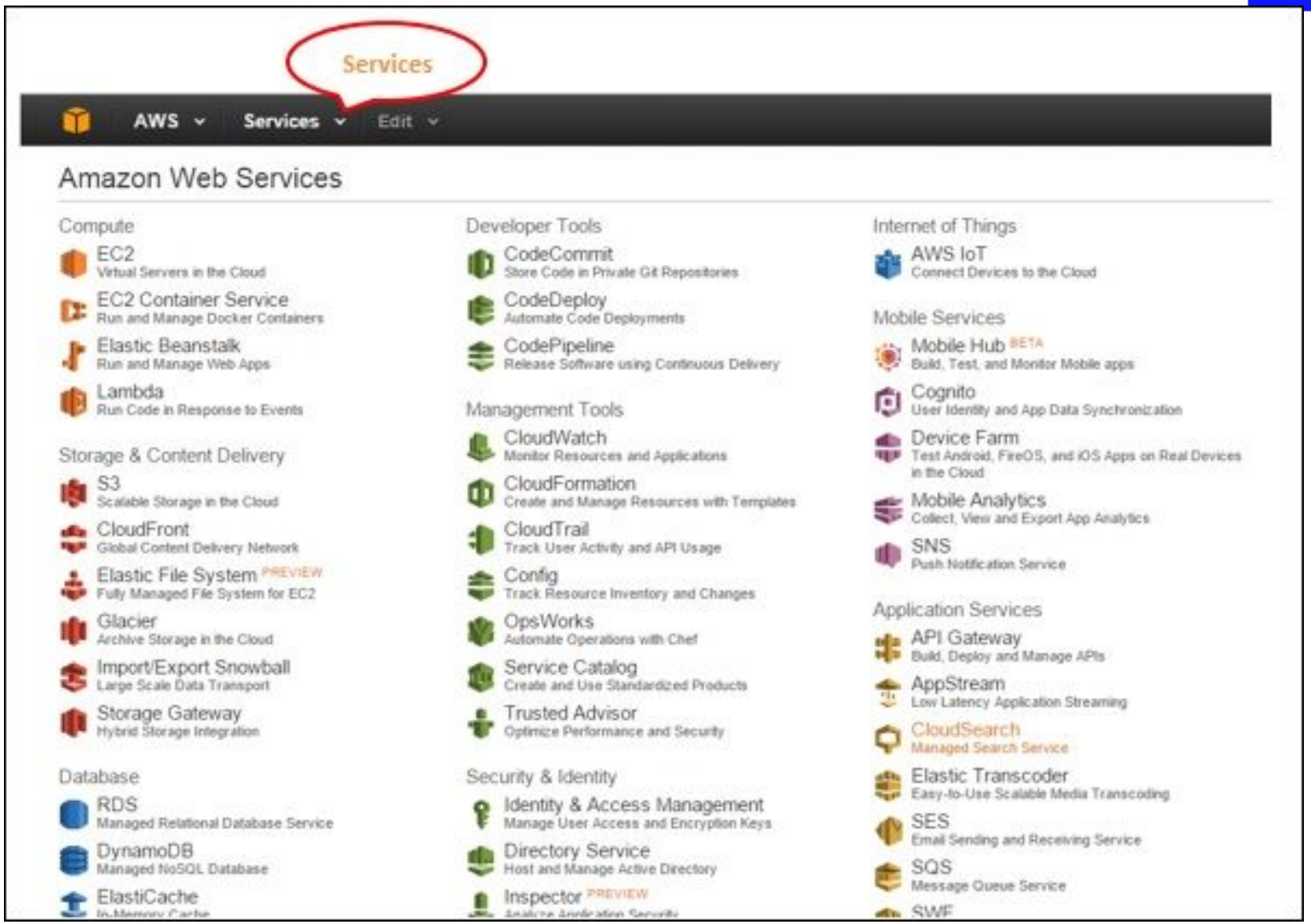
```
import boto3
s3 = boto3.resource('s3')

# 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'))
```



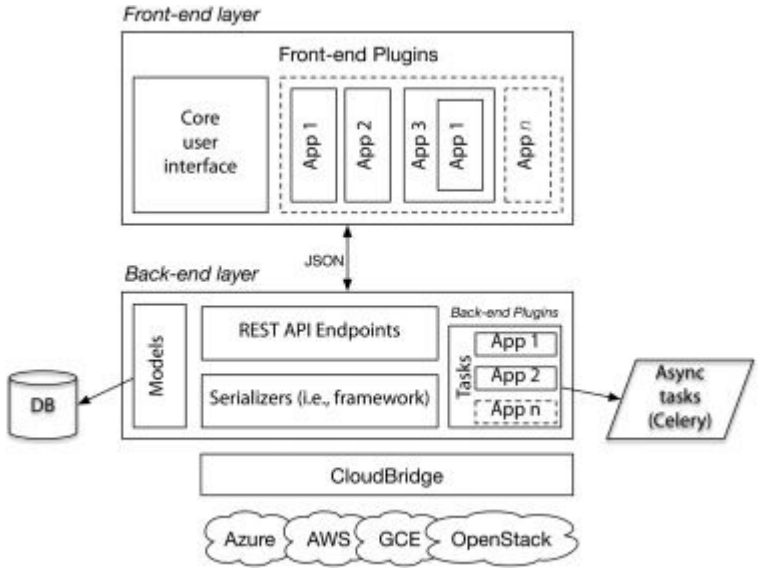
Architecture of API Platform Cloud Service




Connected Users and Streaming Dashboards
Web and Mobile Applications
IoT Devices
Private Applications: VPC and On-Premises
Amazon API Gateway
Create, publish, maintain, monitor, and secure APIs at any scale
API Gateway cache
Amazon CloudWatch
AWS Lambda
Amazon EC2
Amazon Kinesis
Amazon DynamoDB
Other AWS services
Publicly accessible endpoints
Private Applications
VPC and On-Premises
Data Center


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
SWF

- 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.

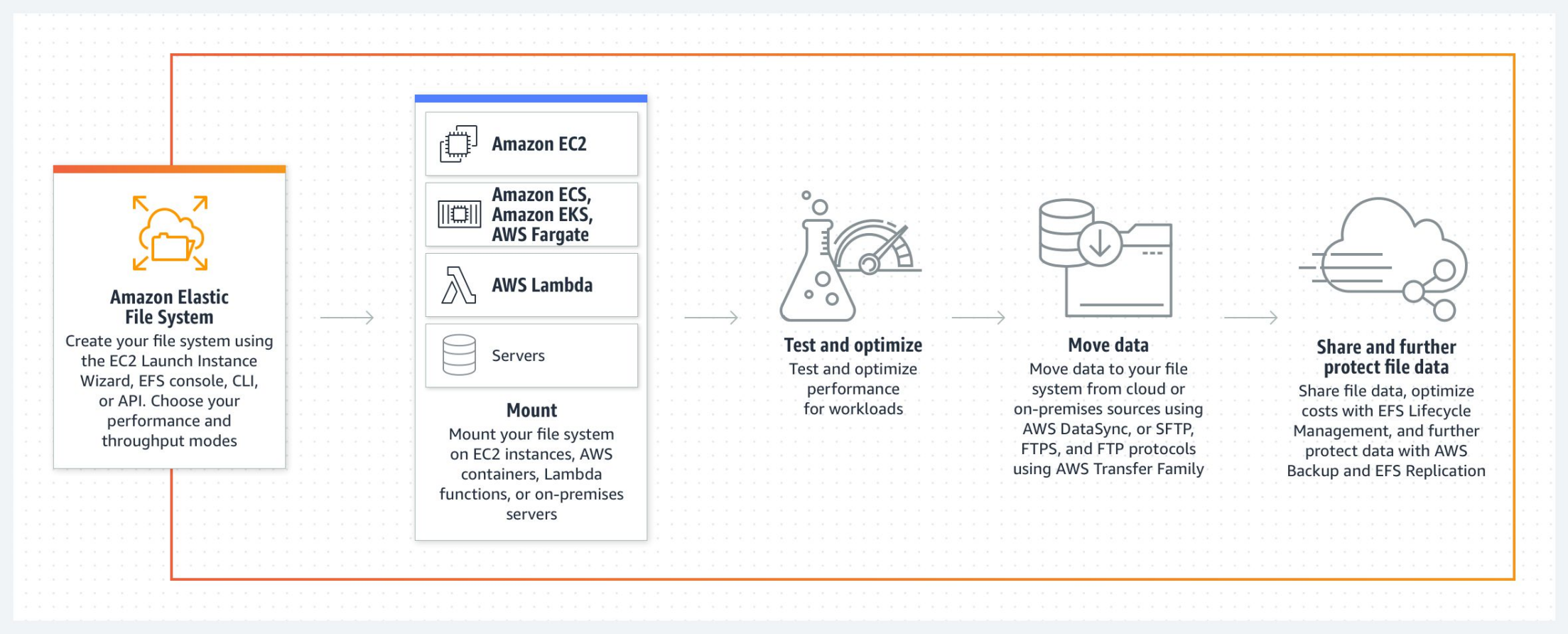


- Building a data sample collection in the cloud
  - *Have a collection of data samples stored on our personal computer*
  - *For each sample use* **METADATA**: *item number, creation date, experiment id, and a text string comment.*
  - *To enable access to these samples by our collaborators, we want 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***

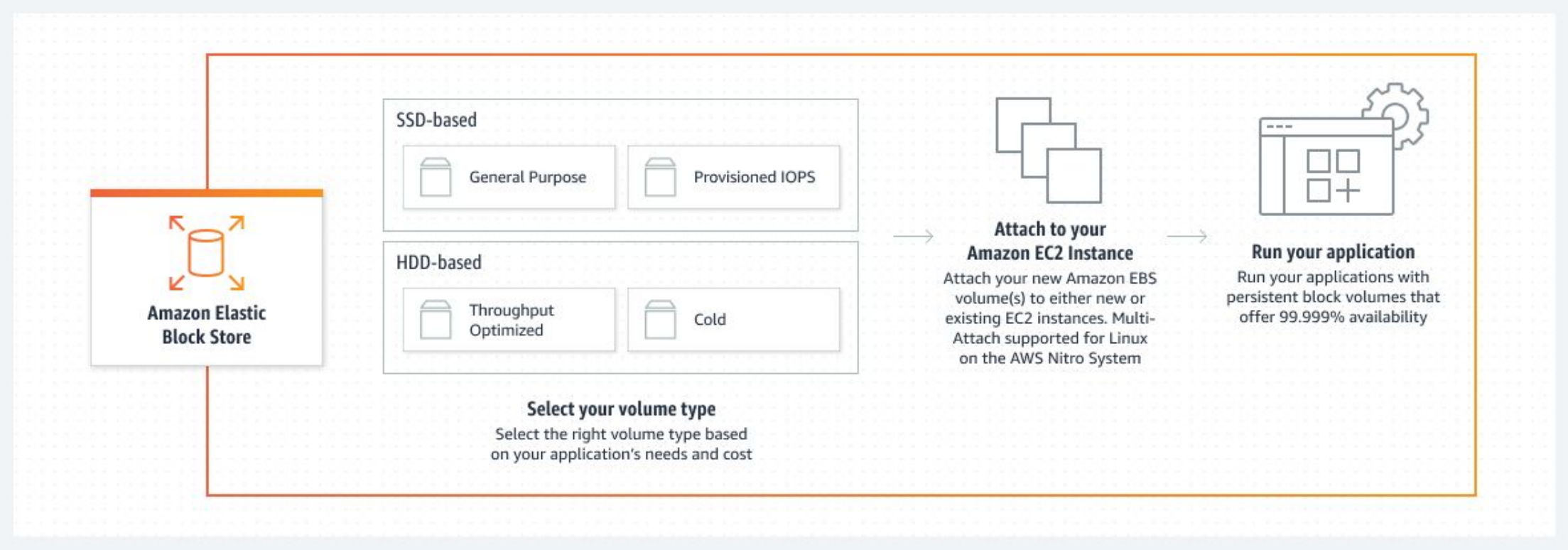
Amazon S3


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.

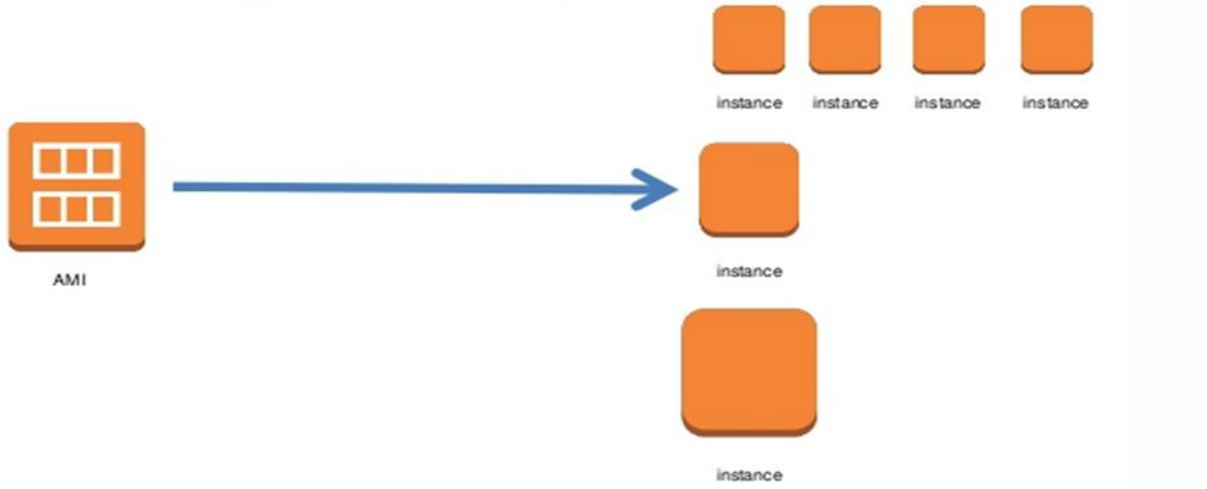
Delete an Object

- **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.

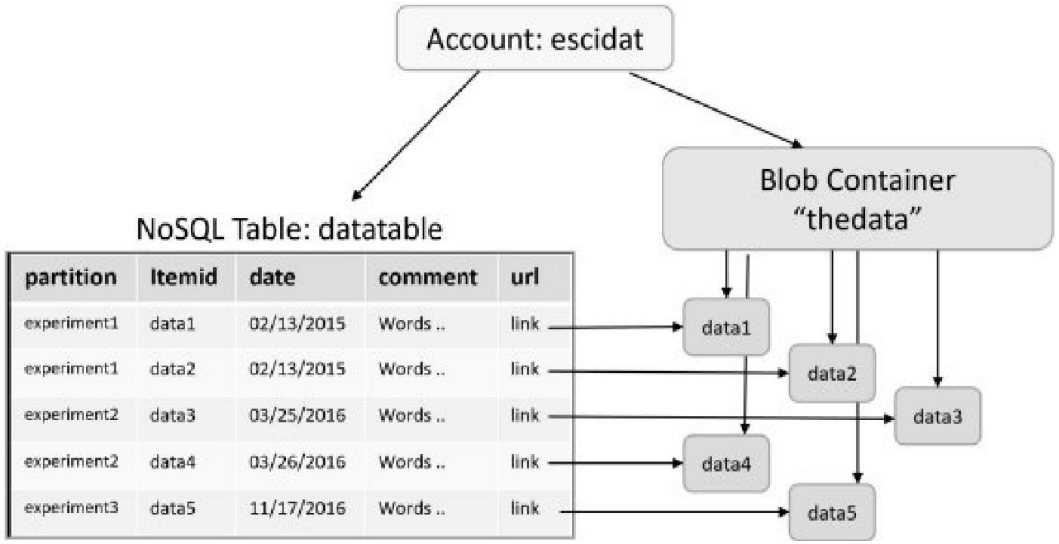
LAUNCH INSTANCE in EC2

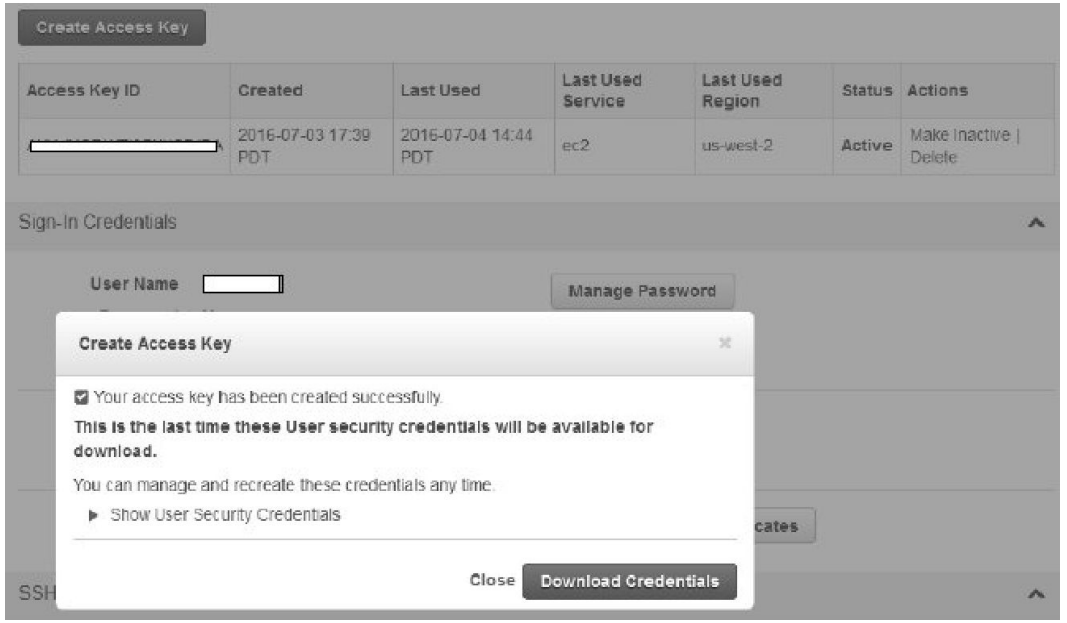


Using Amazon Cloud Storage Services

- 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





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**

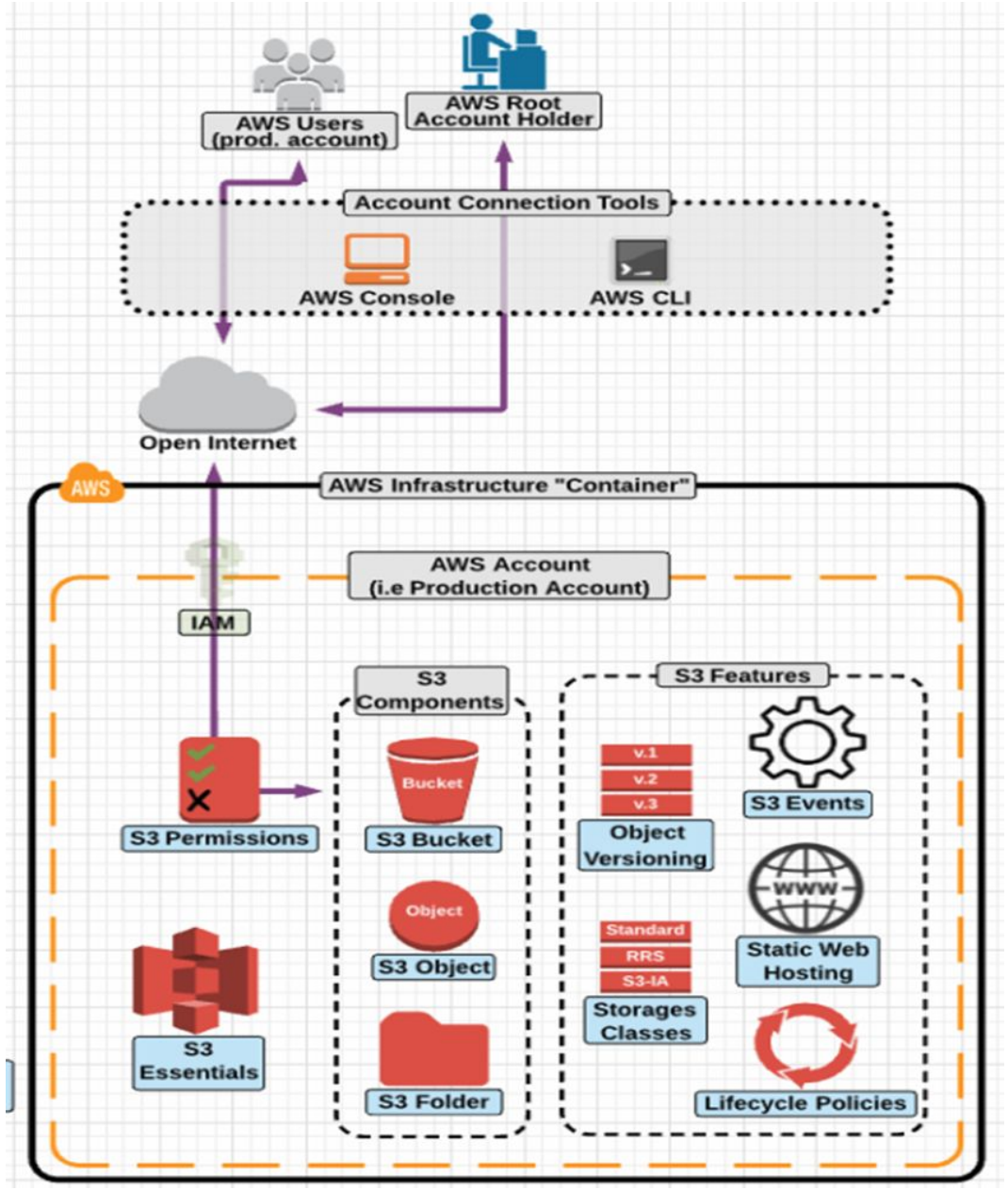
Before March 1, 2018, buckets created in the US East (N. Virginia) Region could have names that were up to 255 characters long and included uppercase letters and underscores. Beginning March 1, 2018, new buckets in US East (N. Virginia) must conform to the same rules applied in all other Regions.

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

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.
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.

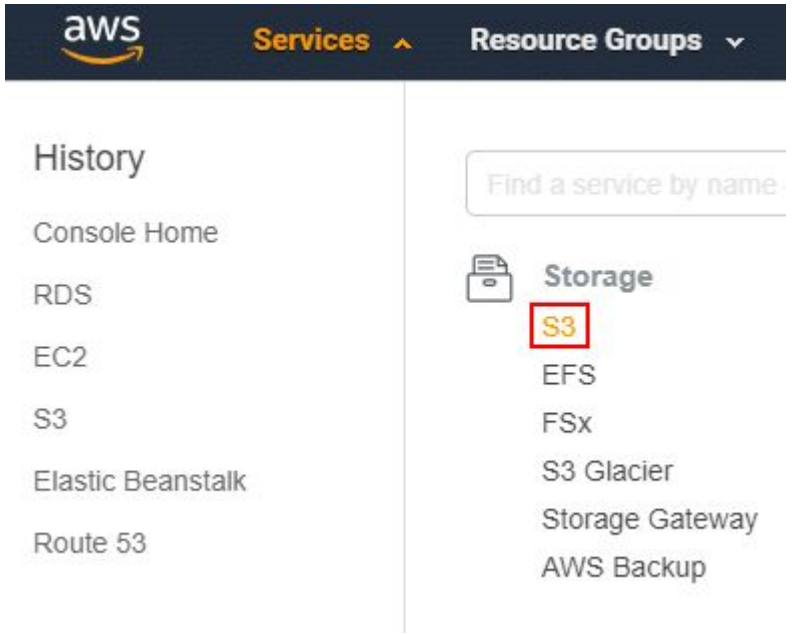
Features	Description
transfer acceleration	Transfer Acceleration enables fast, easy, and secure transfers of files over long distances between your client and an S3 bucket
versioning	Versioning helps you recover accidental overwrites and deletes.
website	You can configure your bucket for static website hosting. Amazon S3 stores this configuration by creating a *website* sub resource.



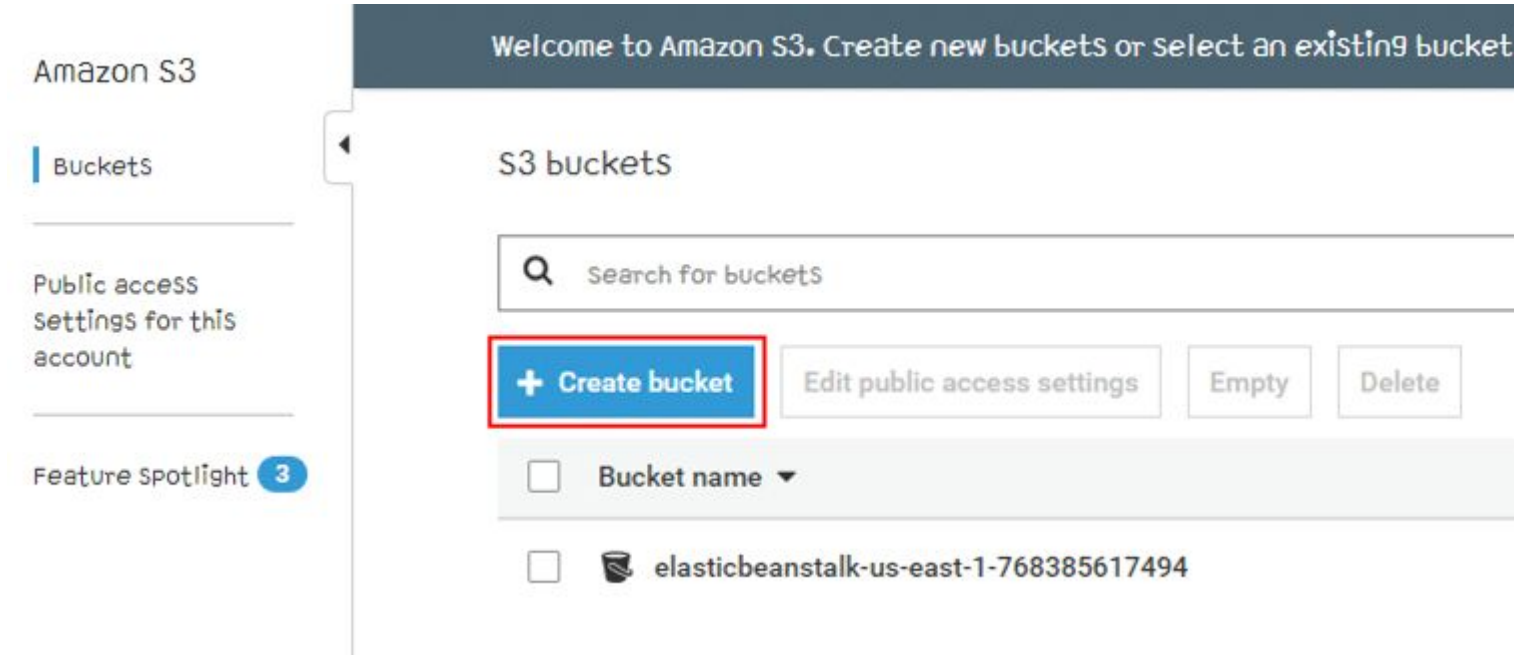
https://techvinixblog.wordpress.com/2018/07/02/amazon-simple-storage-services3-and-amazon-glacier-storage/

General setup for Amazon S3

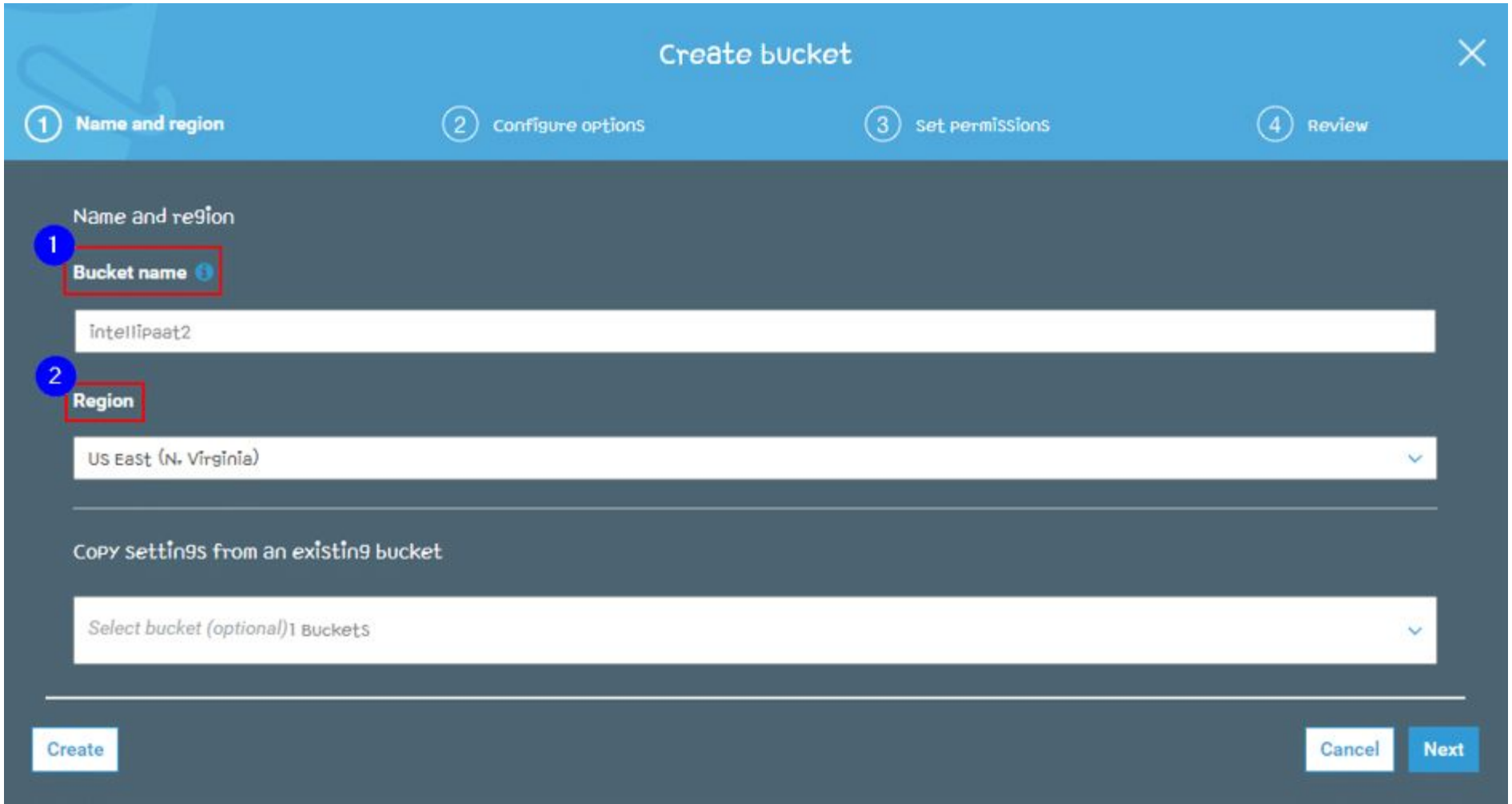
- Step 1: Login to the AWS Management Console
- Step 2: Select S3 from the Services section



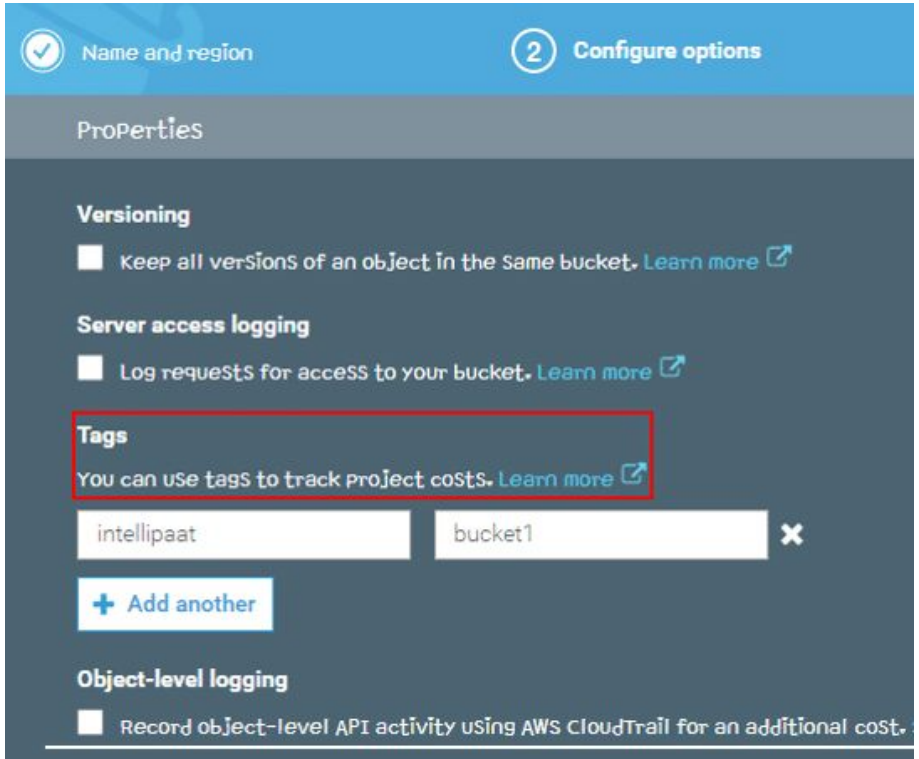
- Step 3: Click on the Create bucket button to start with creating an AWS S3 bucket



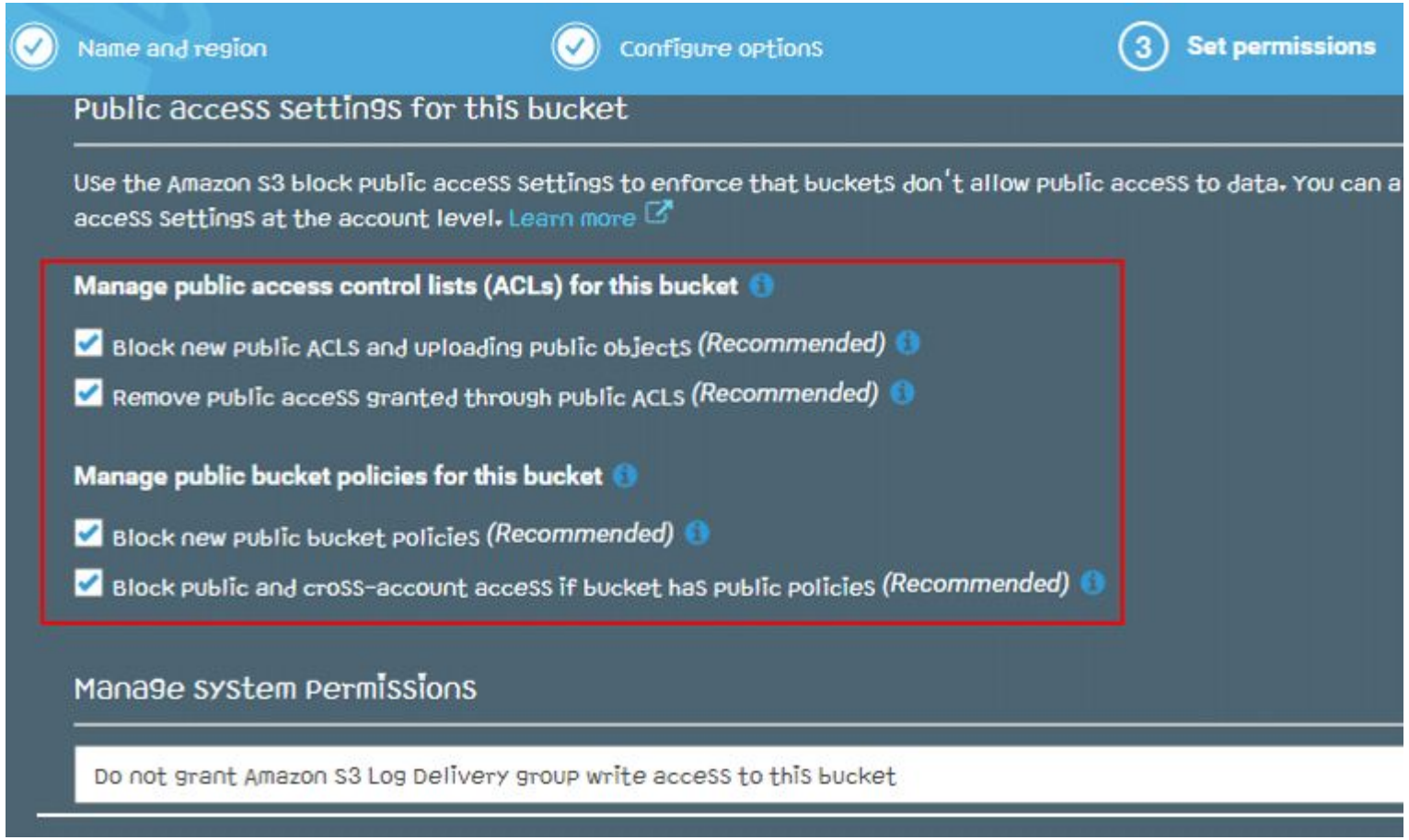
- Step 4: Now, provide a **unique Bucket name** and select the Region in which the bucket should exist. After providing the details click on Next



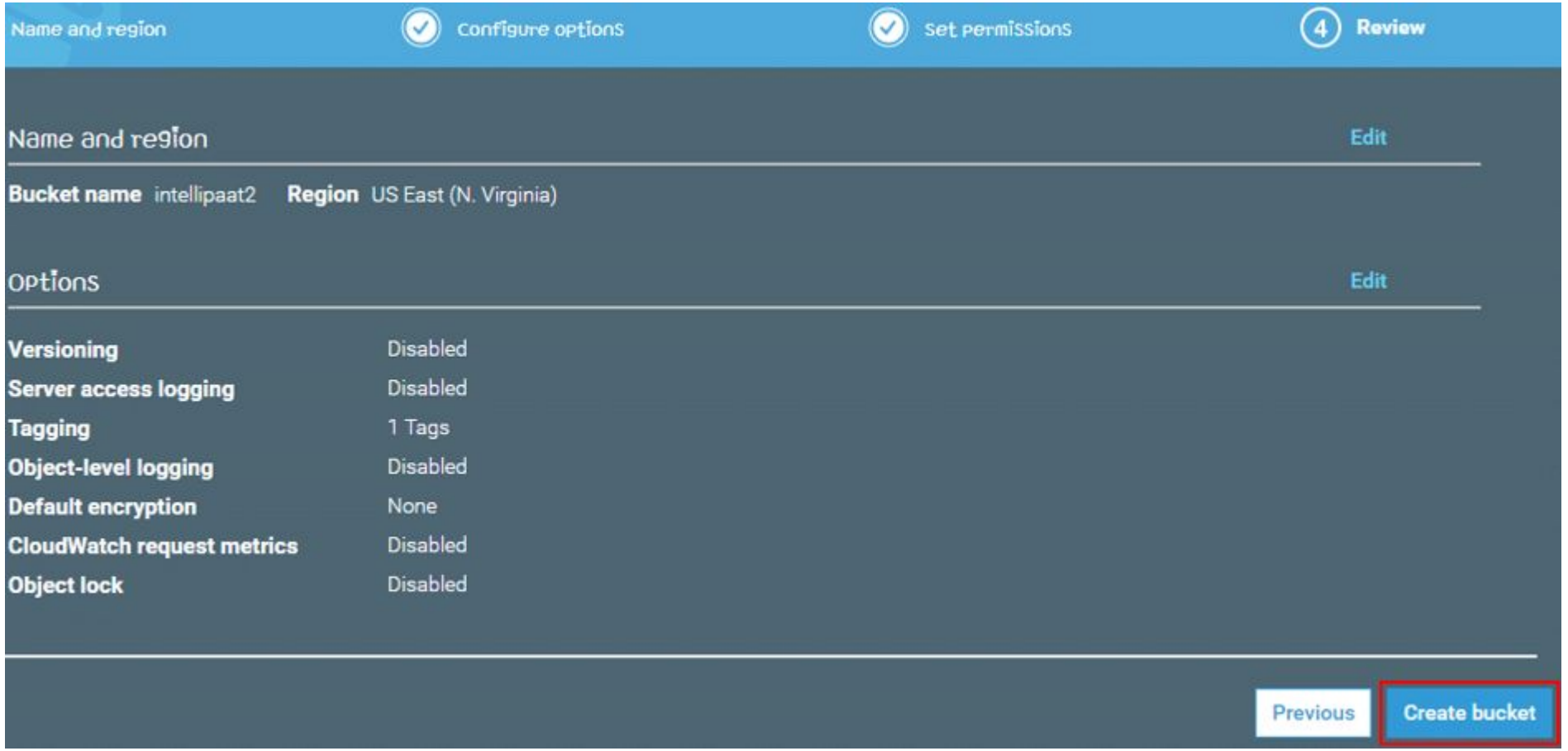
- Step 5: Next is Configure options, and there is no need to provide any details here. You can just click on Next
- But if you want to track costs for this bucket, provide a tag to identify it
- Also, you can choose Versioning. Check the Versioning section of this blog to learn what versioning provides

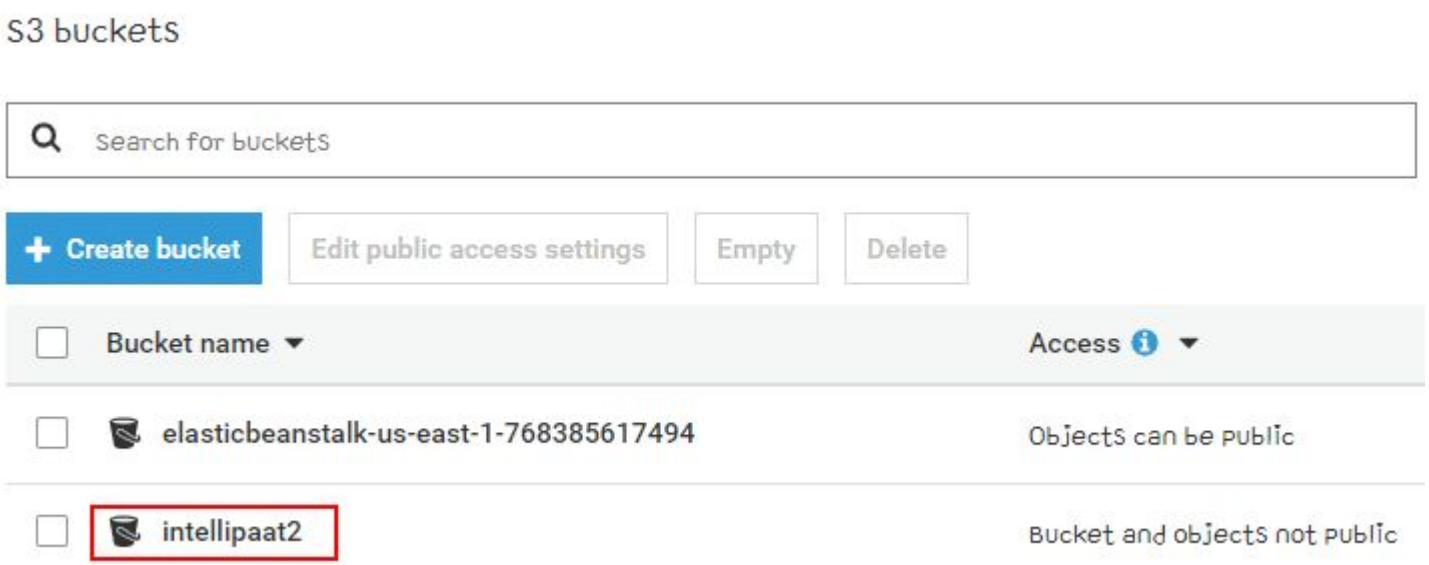


- Step 6: In permissions, keep all checkboxes ticked. This makes the **objects in your bucket inaccessible for the public**



- Step 7: Review once and click on Create bucket





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

- 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

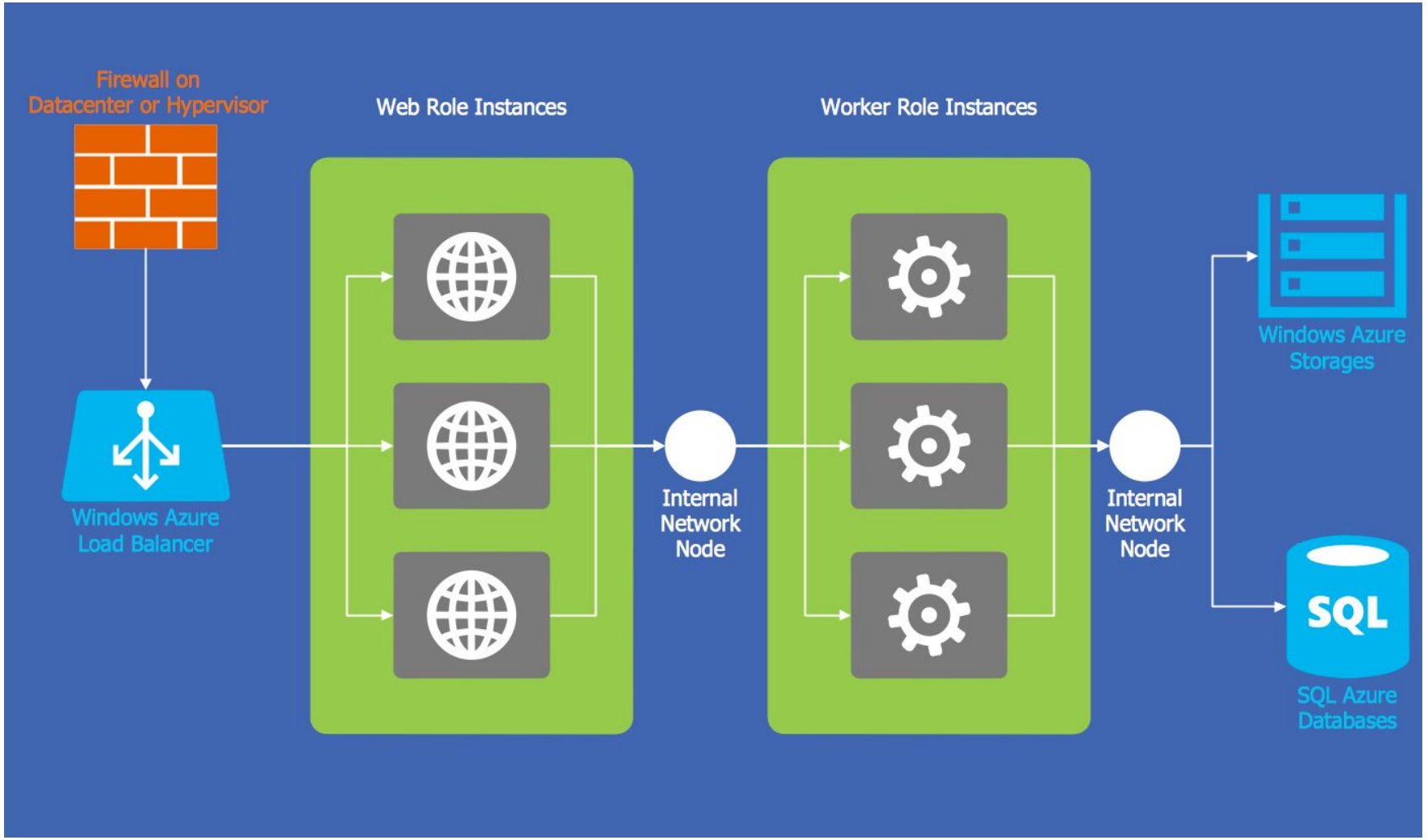
```
import boto3
s3 = boto3.resource('s3')
s3.create_bucket(Bucket='datacont',
CreateBucketConfiguration={
'LocationConstraint': 'us-west-2'})
# Upload a file, 'test.jpg' into the newly
created bucket
s3.Object('datacont', 'test.jpg').put(
Body=open('/home/mydata/ test.jpg',
'rb'))
```

- 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

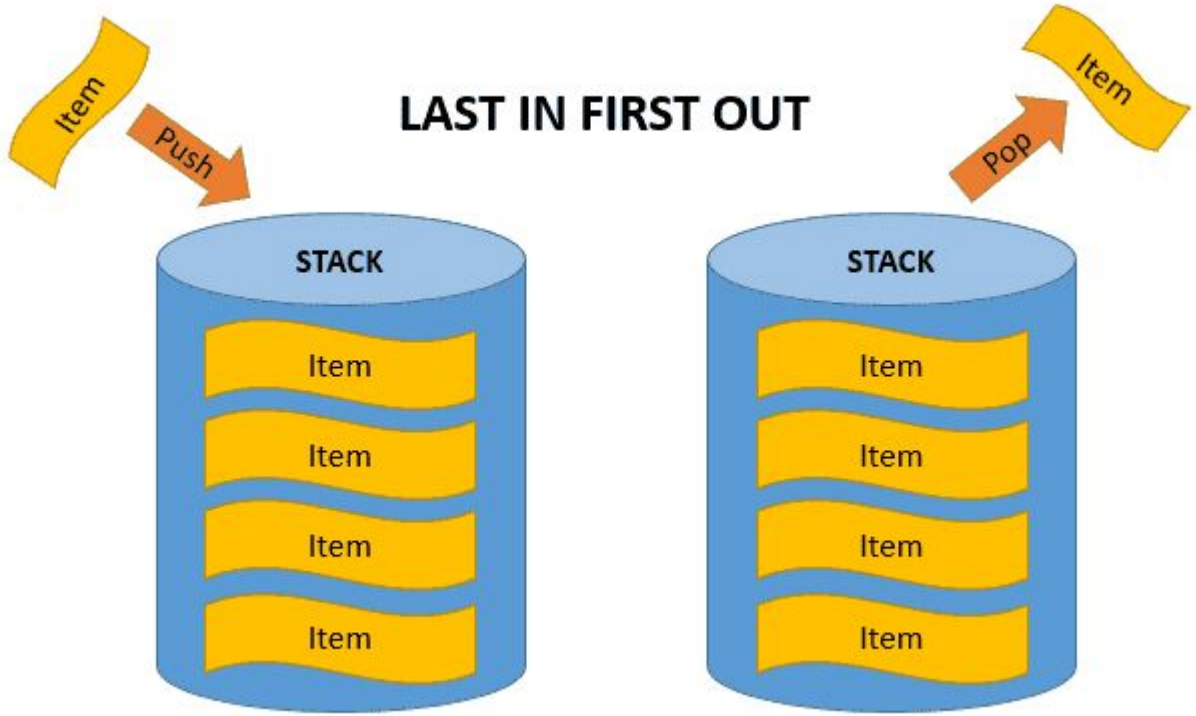


- Azure Storage Services – Accessing services
- Azure Storage Services – Storage account
- Azure Storage Services – Blob storage
- Azure Storage Services – Table storage
- Azure Storage Services – File storage
- Azure Storage Services – Queue storage

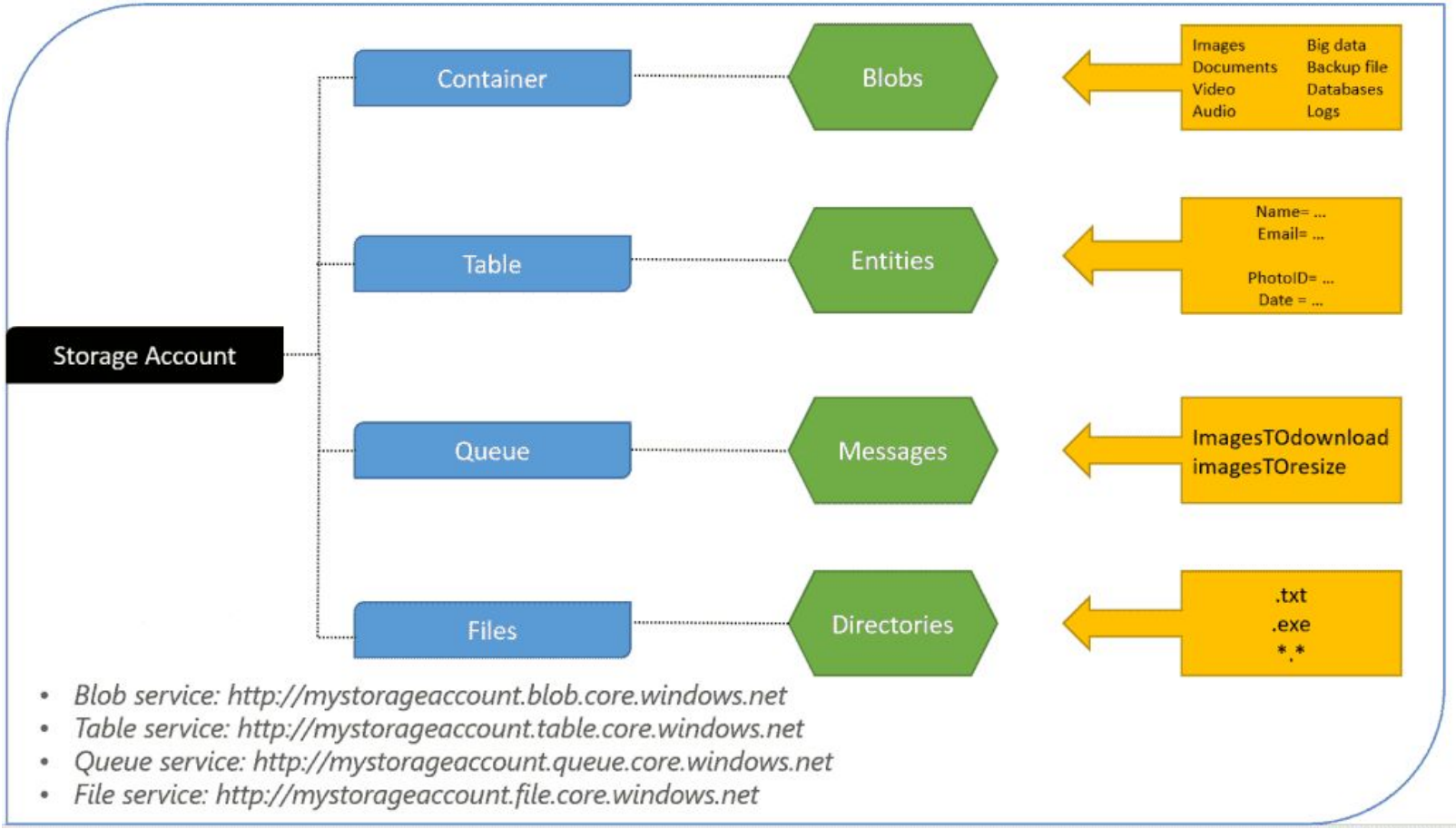
Blob types

- 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

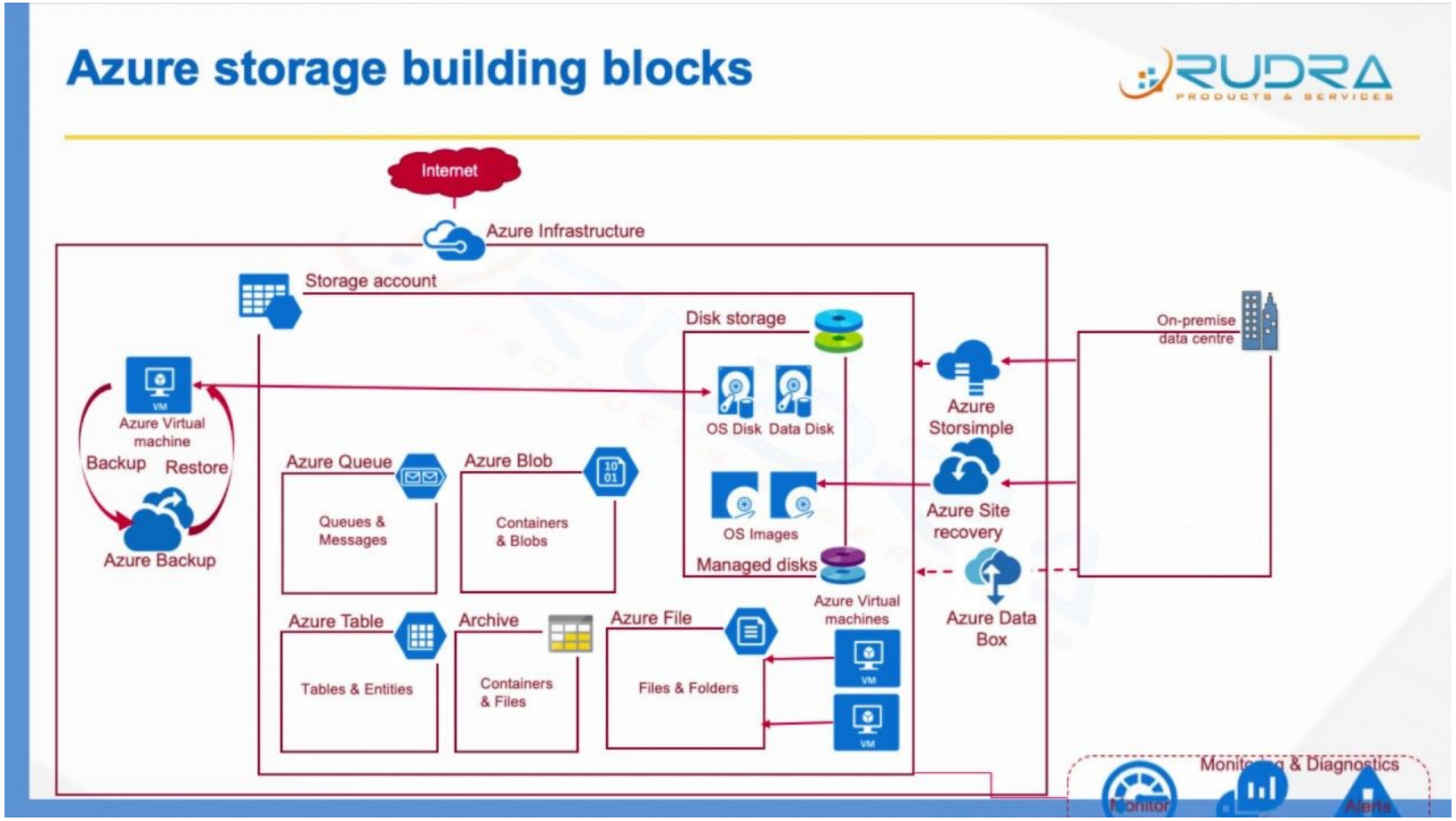
Queue data structure



Azure Storage Services – File storage



Microsoft Azure Storage



Using Microsoft Azure Storage Services

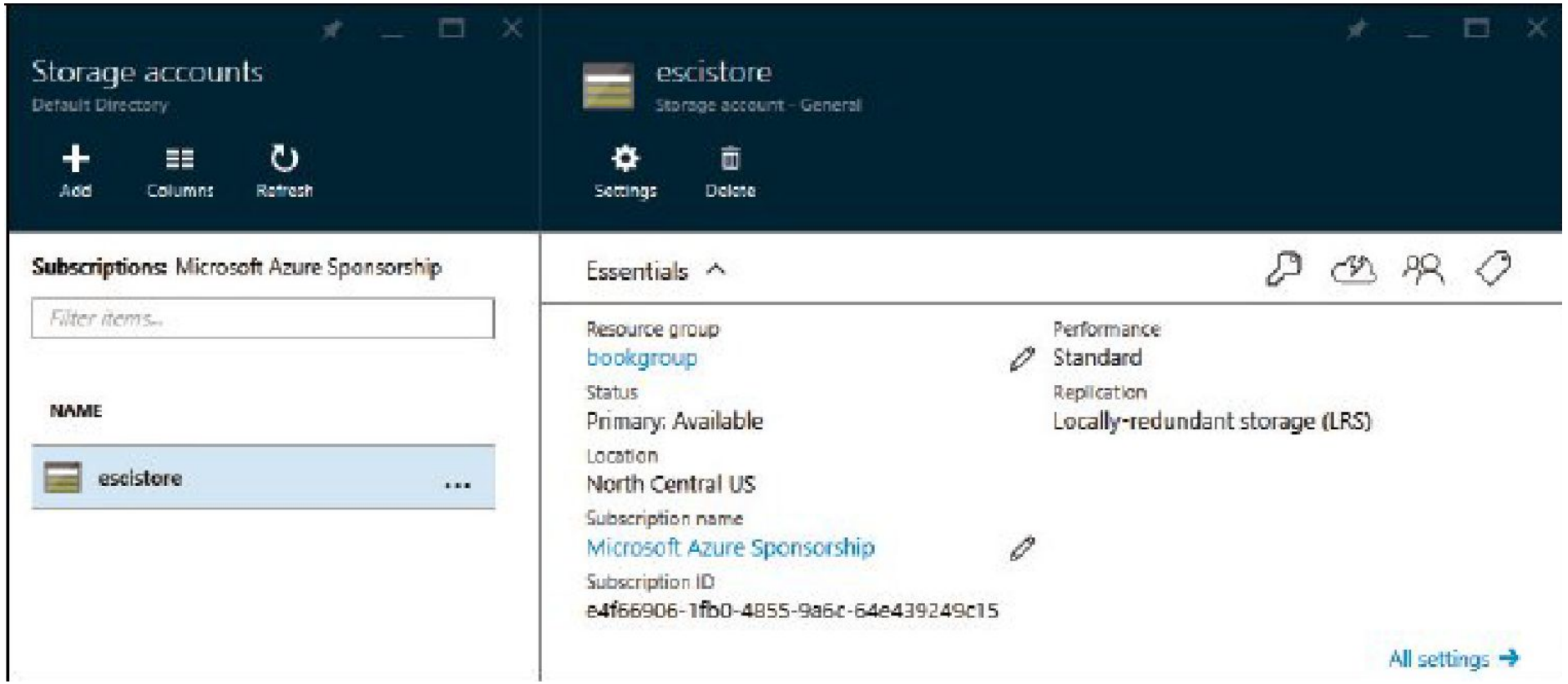
- Azure account is defined by your personal ID and a subscription ID.
- Your personal ID is probably your email address, so that is public
- The subscription ID is something to keep secret.
- The differences between Amazon DynamoDB and the Azure Table service are subtle.
- With the Azure Table service, 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

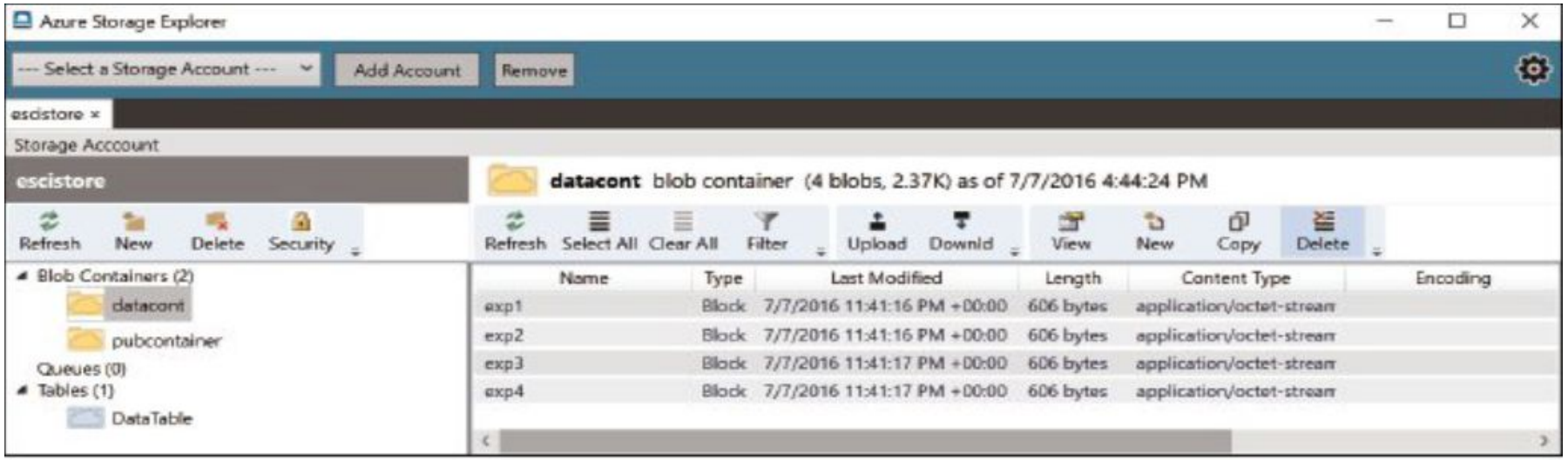
Components	Description
FABRIC CONTROLLER	• **Fabrics** are group of machines in Microsoft's datacenter which are aggregated by a switch. • The group of these machines is called **CLUSTER**. • Each cluster is managed and owned by a **fabric controller.**
BLOB	'Blob' expands to **B**inary **L**arge **OB**ject. 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.

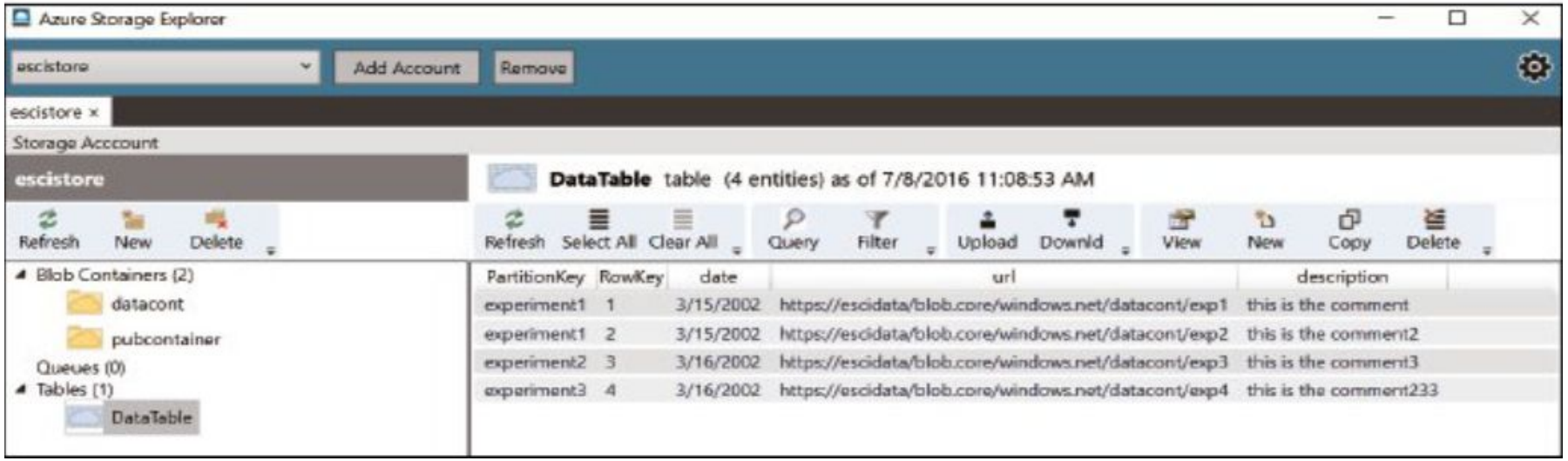
Components	Description
Tables	• Storing a table does not mean **RELATIONAL DATABASE** here. • Azure Storage can store just a table without any foreign keys or any other kind of relation. • These tables are highly scalable and ideal for handling large amount of data. • Tables can be stored and queried 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
escistore
Storage account - General
Settings
Delete
Subscriptions: Microsoft Azure Sponsorship
Filter items...
NAME
escistore
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 =
TableService(account name='escistore',
                        account_key='your account
                        key')
if table_service.create_table('DataTable'):
    print("Table created")
else:
    print("Table already there")
```

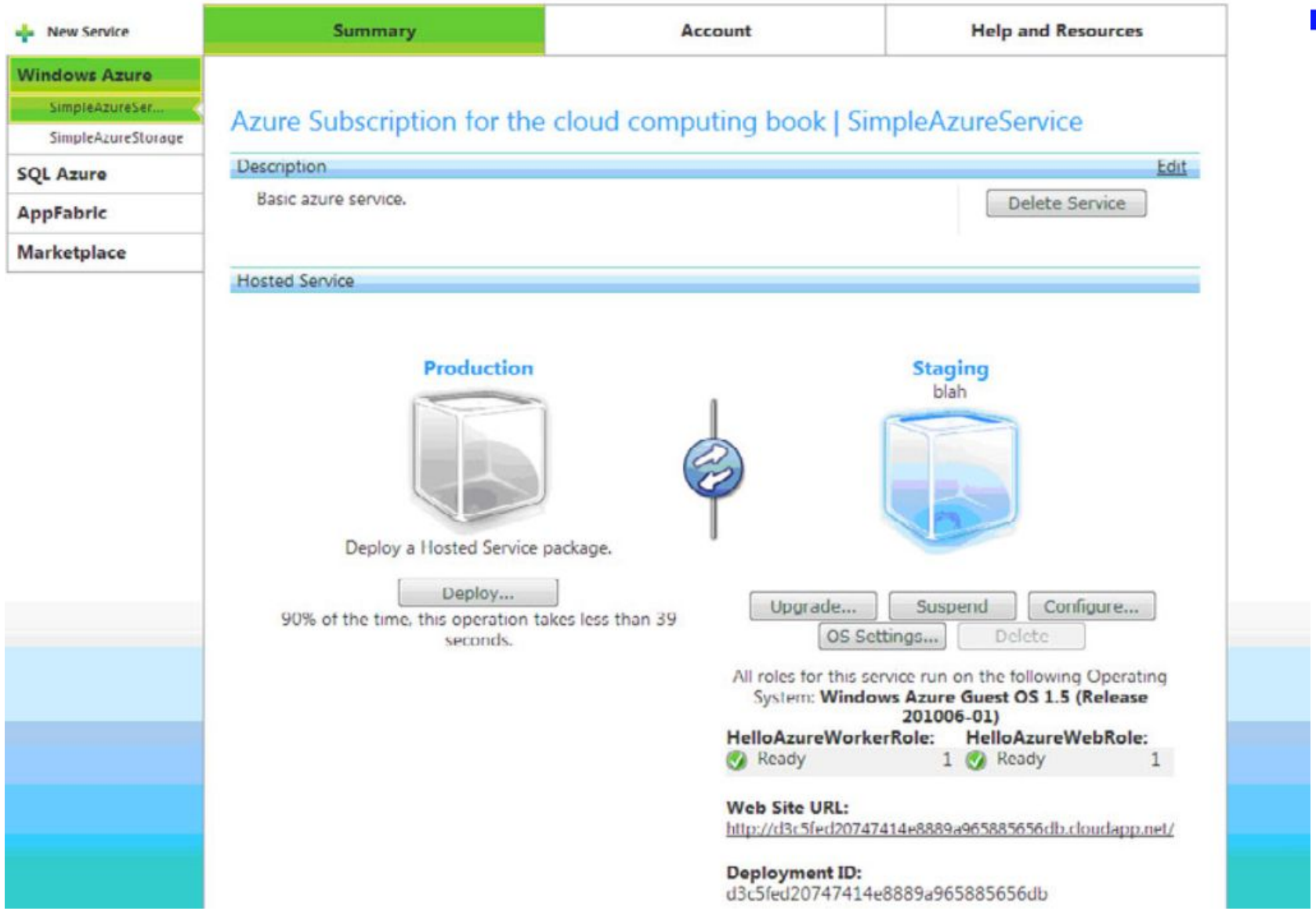
```
import csv
with open('\path-to-your-data\experiments.csv',
'rb') as csvfile:
    csvf = csv.reader(csvfile, delimiter=',',
    quotechar='|')
    for item in csvf:
      print(item)
      block_blob_service.create_blob_from_path(
            'datacont', item[3],
            "\path-to-your-
            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
datacont blob container (4 blobs, 2.37K) as of 7/7/2016 4:44:24 PM
Refresh
New
Delete
Security
Refresh
Select All
Clear All
Filter
Upload
Downld
View
New
Copy
Delete
Blob Containers (2)
datacont
pubcontainer
Queues (0)
Tables (1)
DataTable
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
DataTable table (4 entities) as of 7/8/2016 11:08:53 AM
Refresh
New
Delete
Refresh
Select All
Clear All
Query
Filter
Upload
Downld
View
New
Copy
Delete
Blob Containers (2)
datacont
pubcontainer
Queues (0)
Tables (1)
DataTable
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**

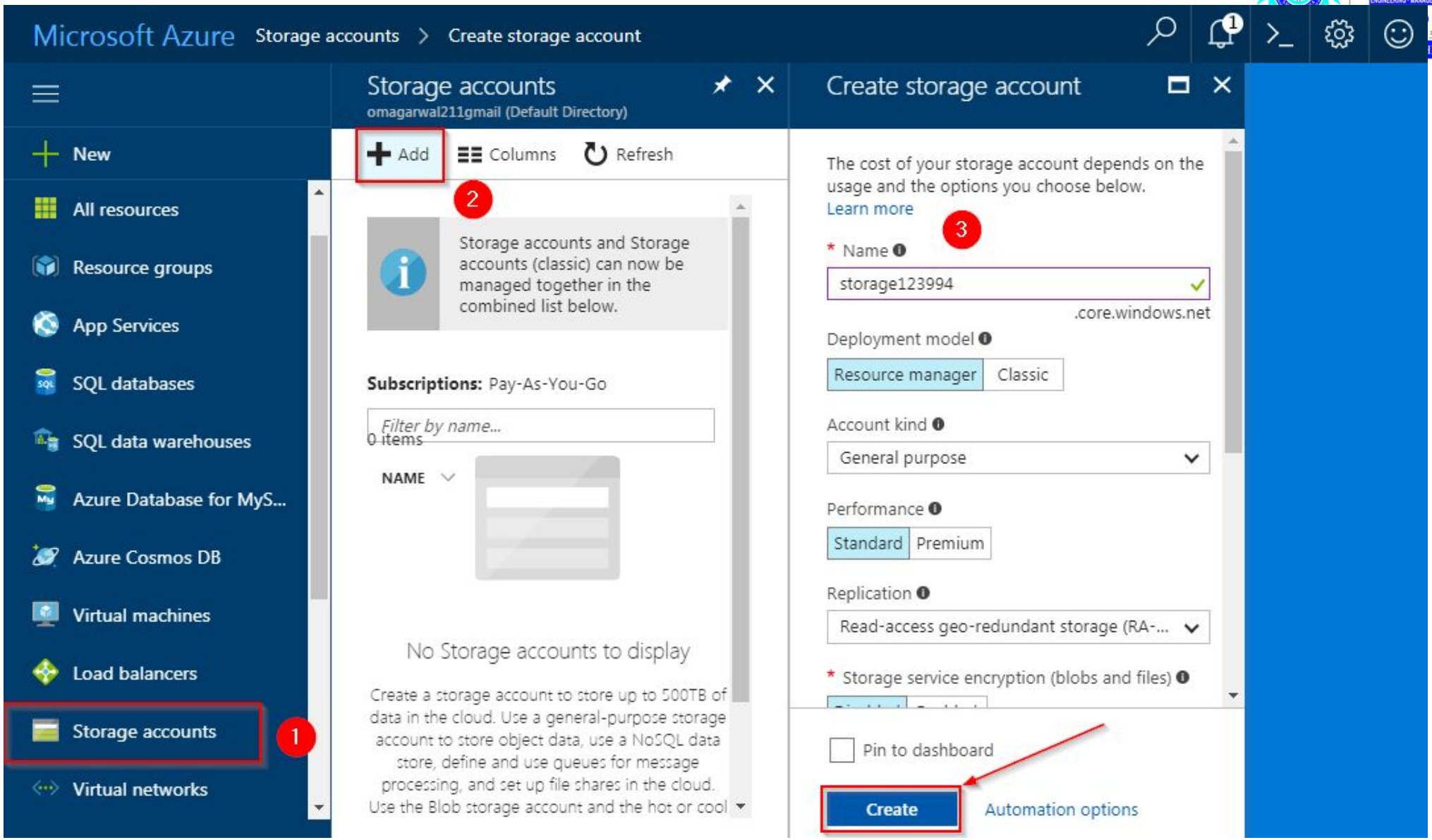
Key Features	Description
Generating an Access Key	• **Azure Storage Service Encryption (SSE)** can automatically encrypt data before it is stored, and it automatically decrypts the data when you retrieve it. • The process is completely transparent to users. • Storage Service Encryption uses **256-bit 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 CreateIfNotExist() GetQueueReference method() ADD A MESSAGE() getmessage() peekmessage() Resultlabel OnStart() 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
Staging
blah
Deploy a Hosted Service package.
Deploy...
90% of the time, this operation takes less than 39 seconds.
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

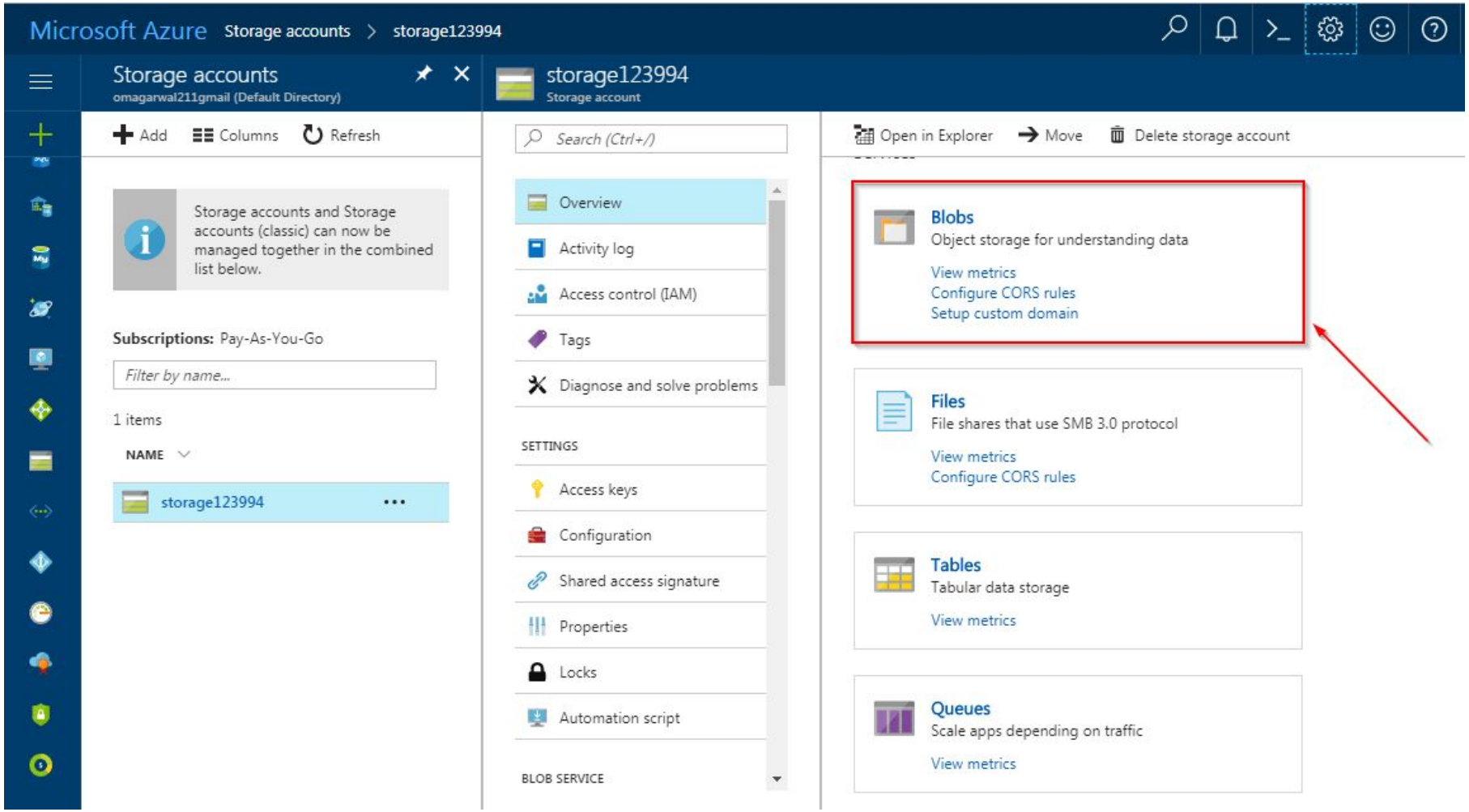
General setup for Microsoft Azure Storage Services

- 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**.

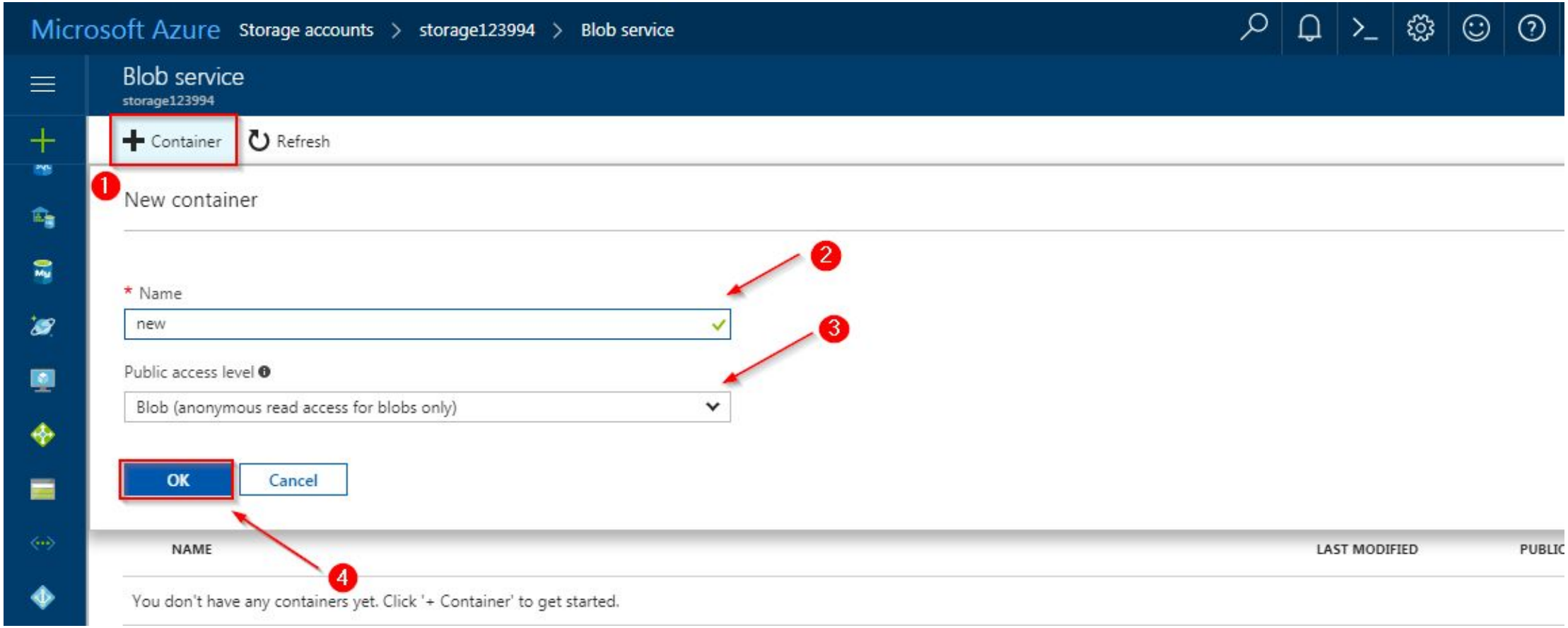
- First, in the left pane click on Storage Accounts
- Then, click on Add
- Finally, enter all the relevant fields and Click on Create


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

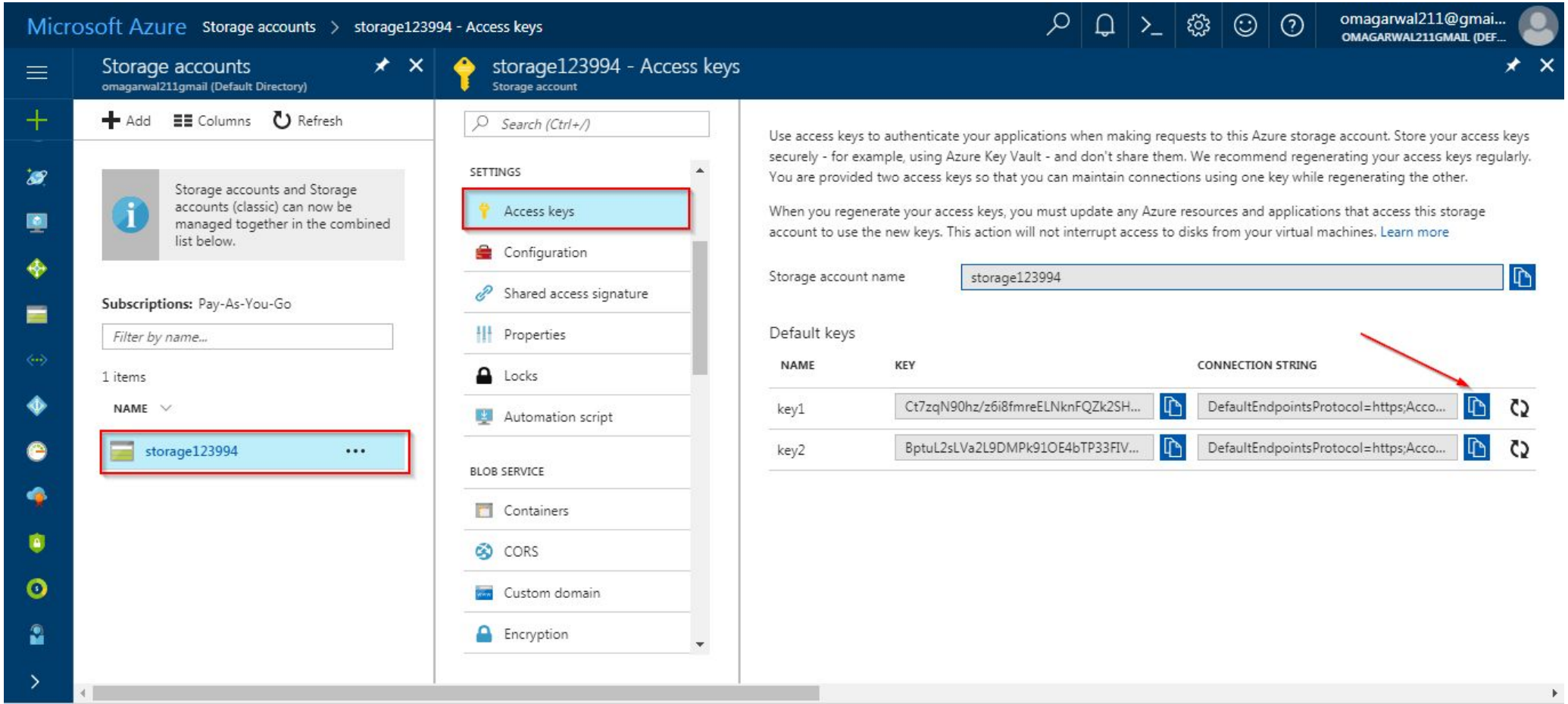
- Step 2:There are four types of storage services in our account, i.e **Blobs, Queues, Files and Tables.**



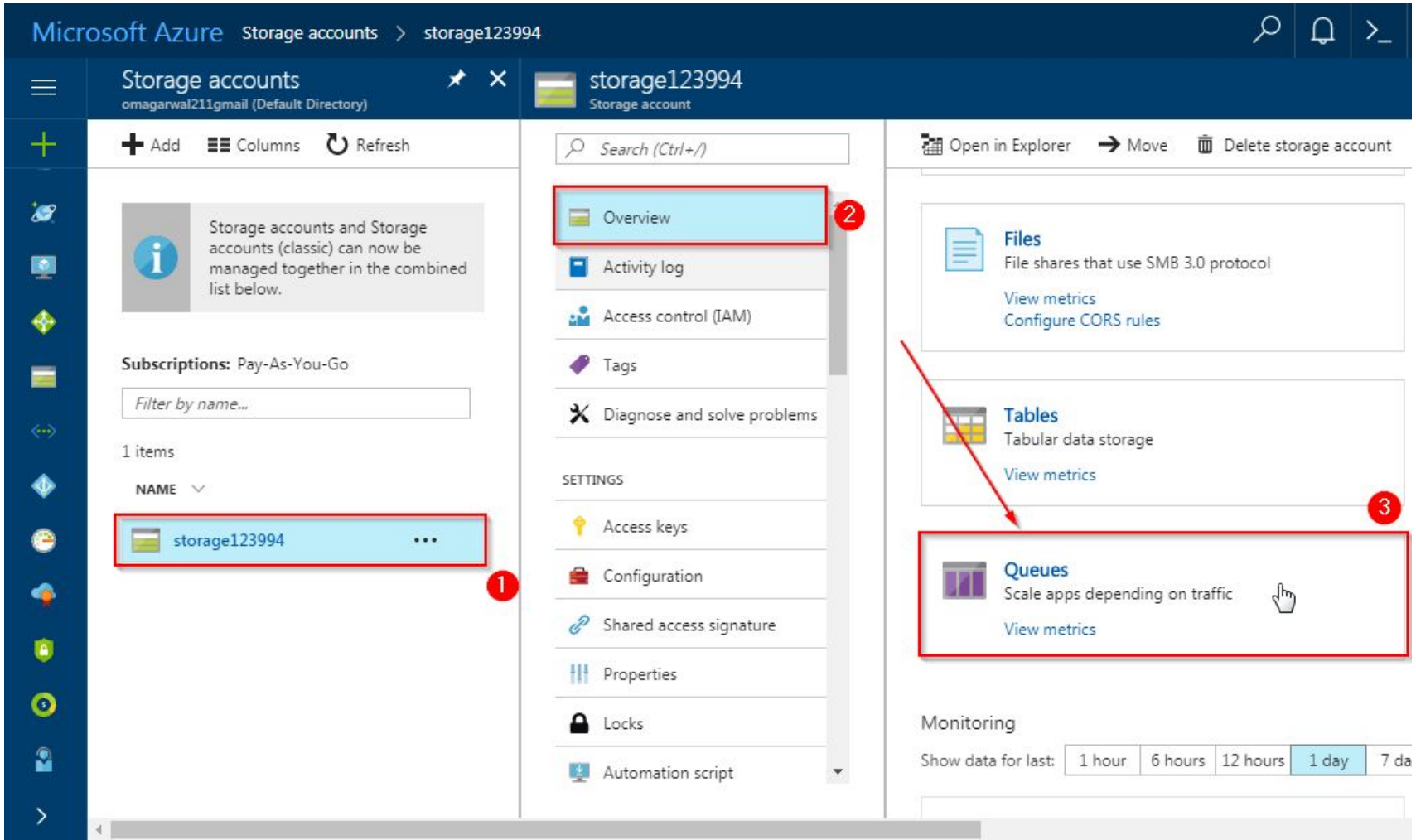
- Step 3: Click on container, **to create a new container**.
  - First, enter the name of the container
  - This should be unique to all the containers that you will be creating in this particular account.
  - Assign public access level to it.
  - Blobs are nothing but files.


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.

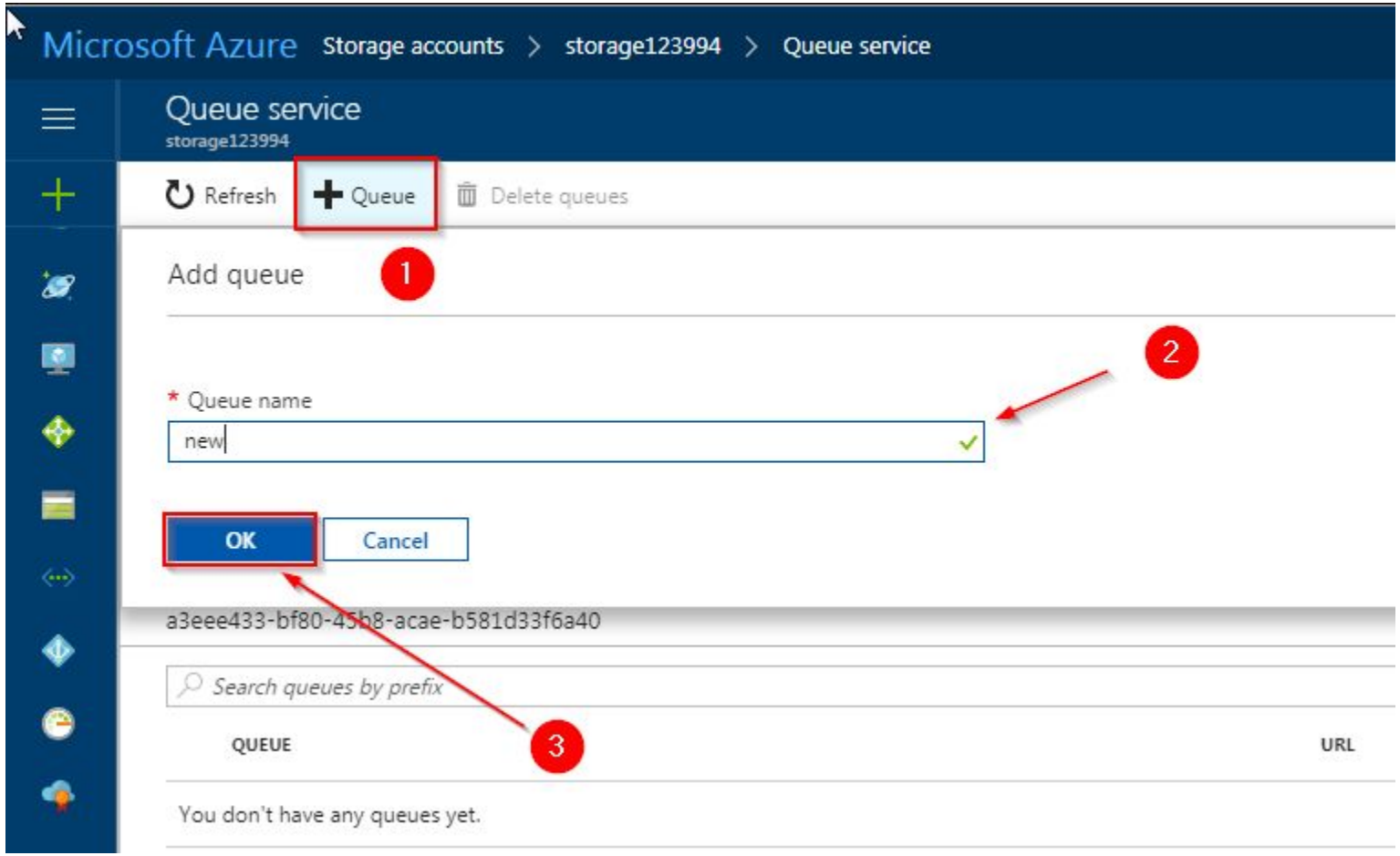
- Step 4: Specify the connection string of your storage account in your website's code.
- A connection string authenticates your code to interact with the specified storage account and its services.



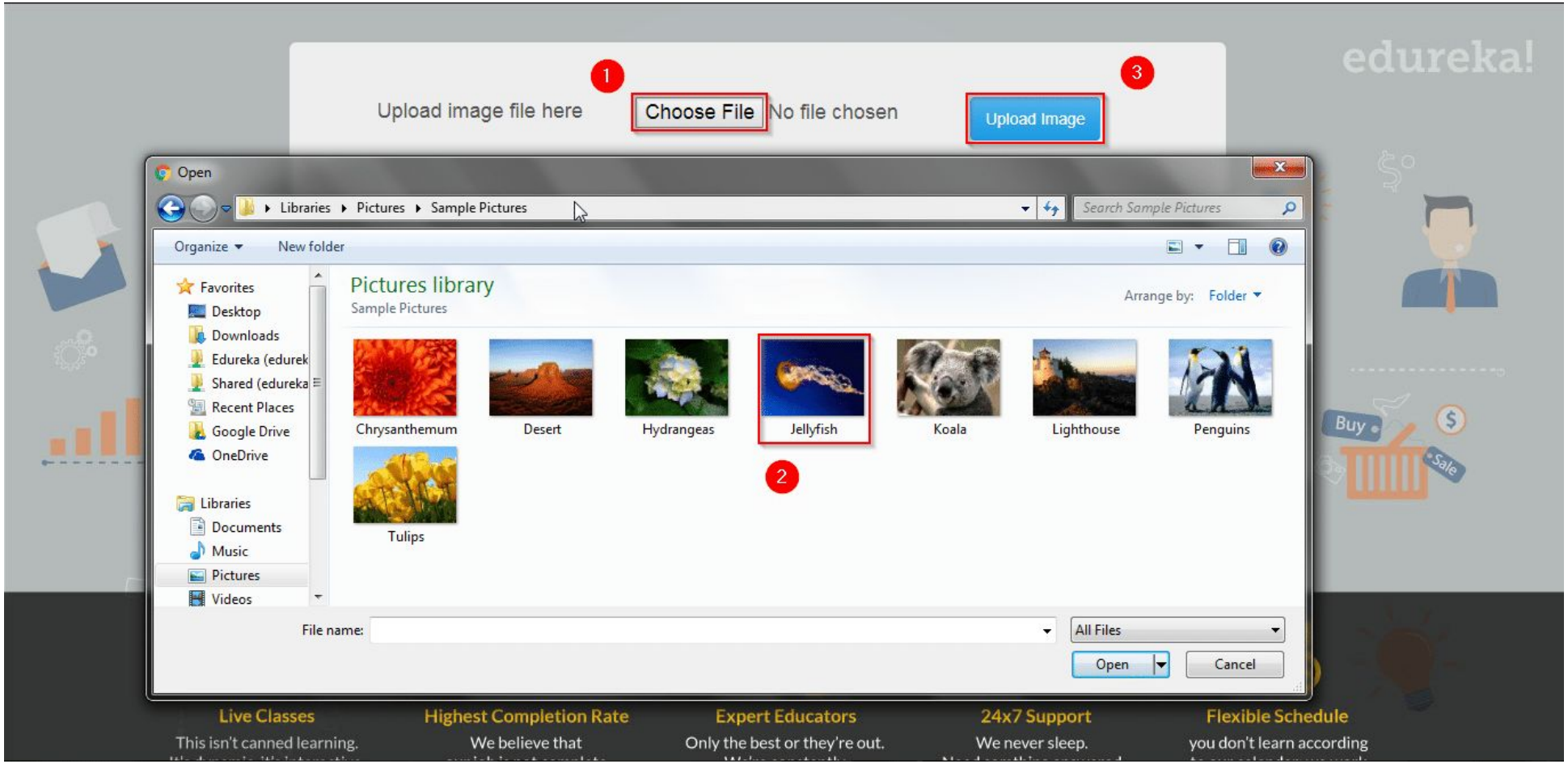
- Step 5: Let's start with the queue now.
- On your storage accounts overview page select queues

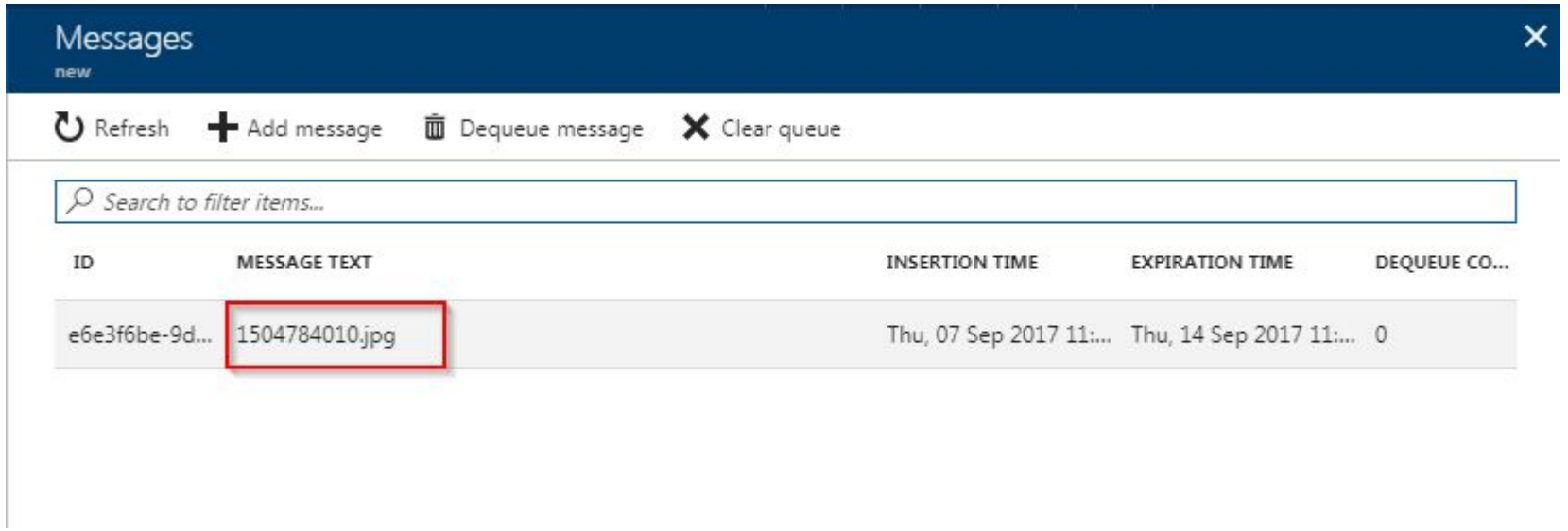


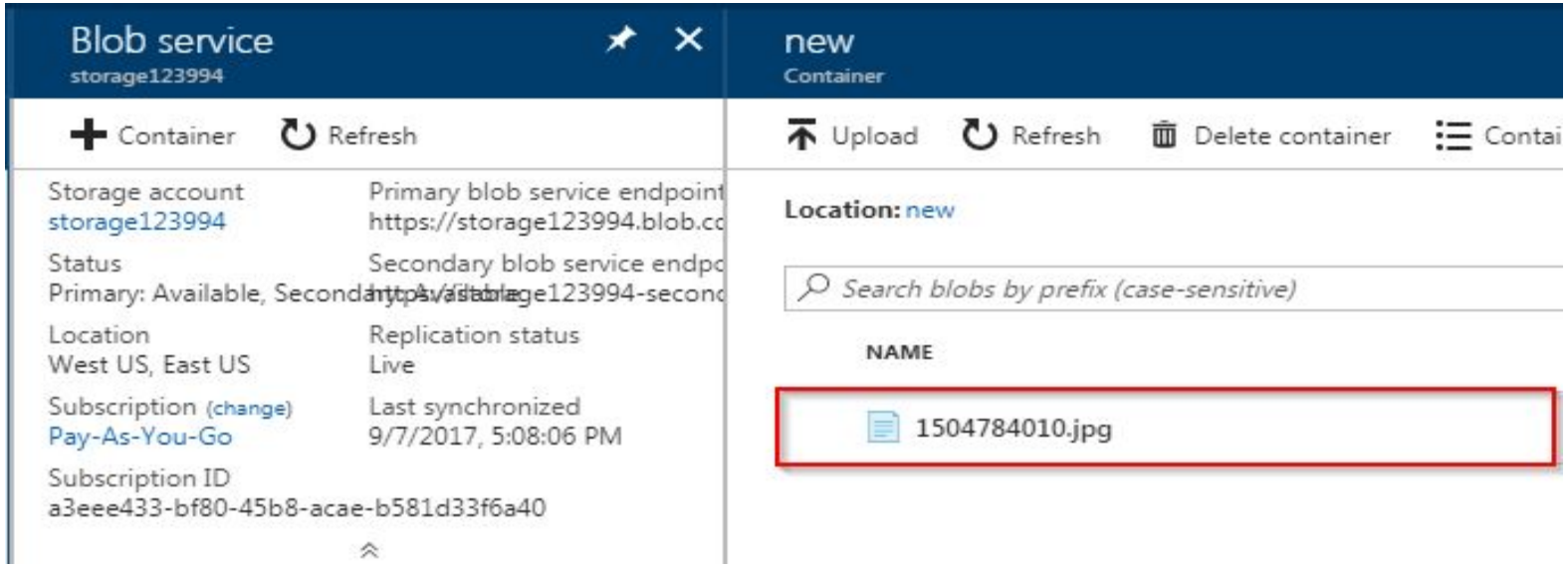
- Step 6: Next, we'll be creating a queue. To do that, Click on Add Queue, give a relevant name to the queue and click on



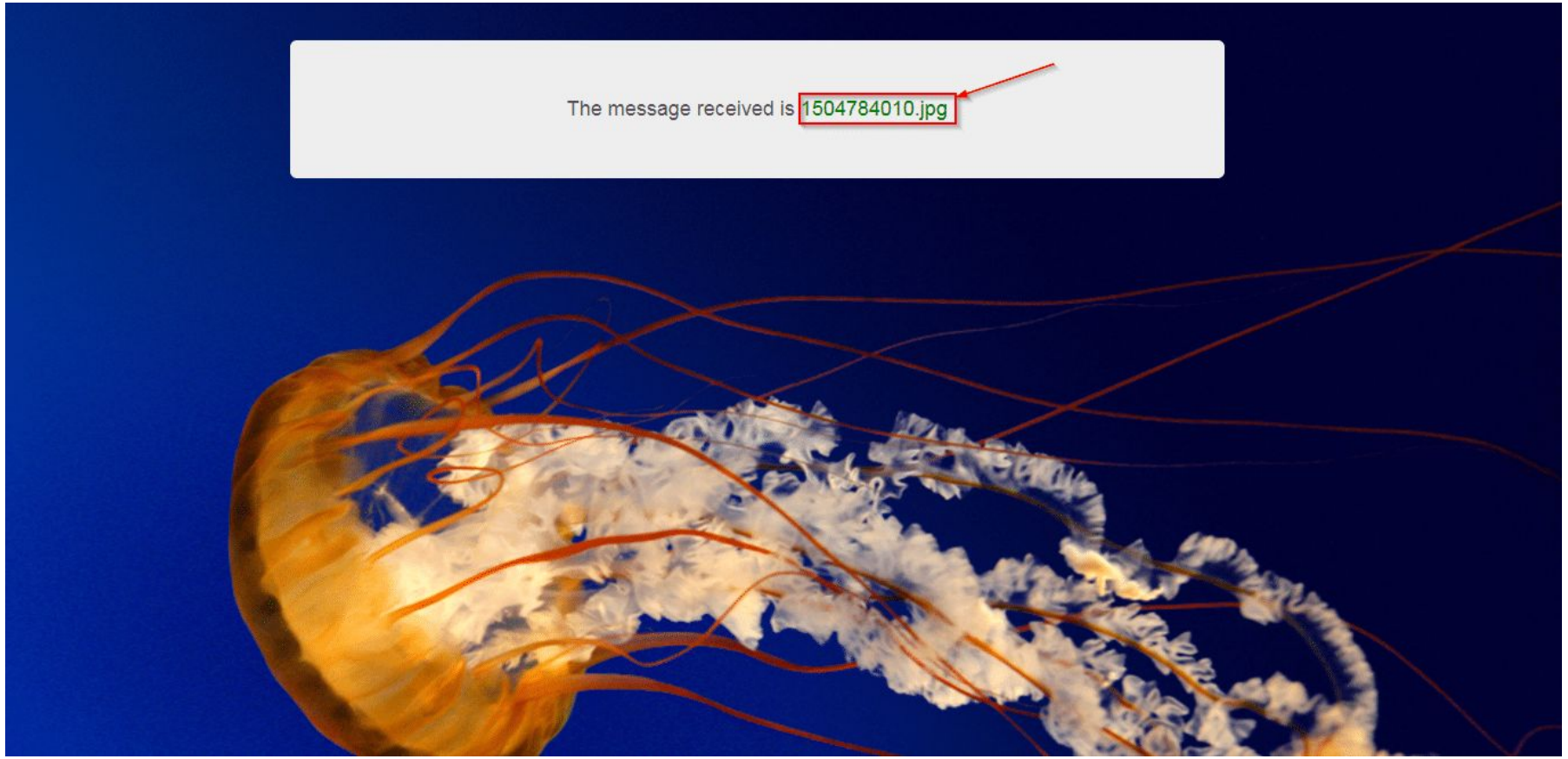
- Step 7: This is the website we have made, select the file that you want to upload, and click on upload.




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

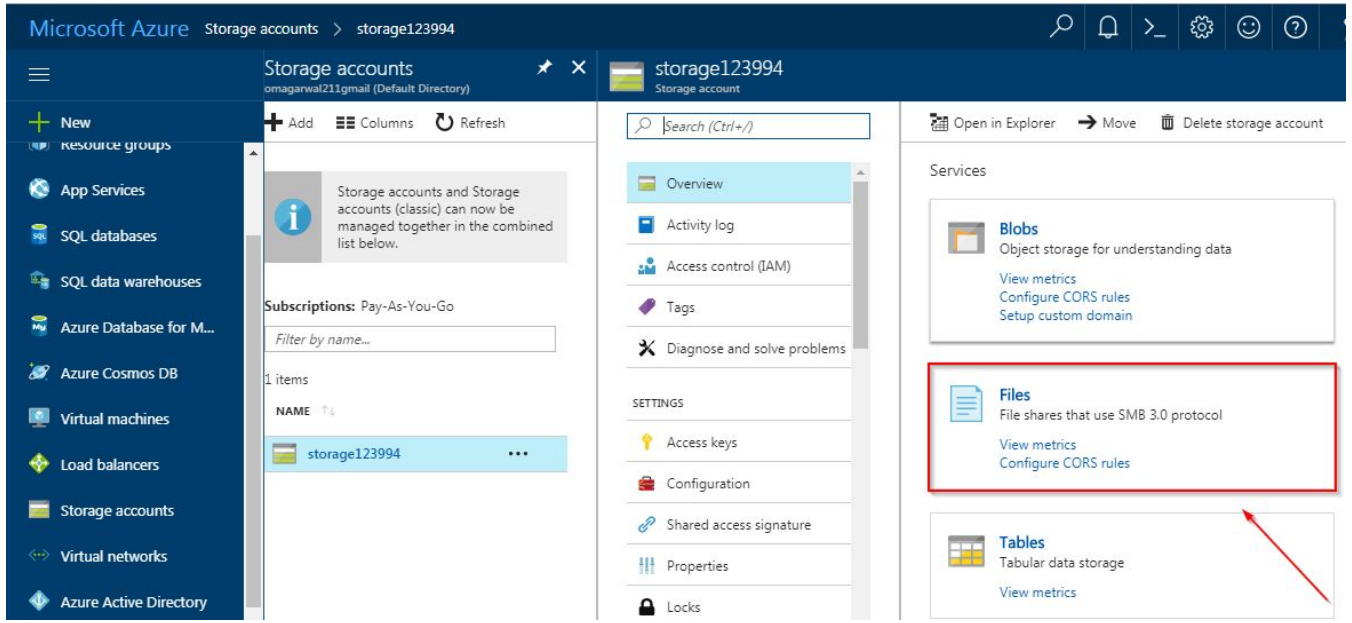

Blob service
storage123994
Container
Refresh
Storage account
storage123994
Primary blob service endpoint
https://storage123994.blob.co
Status
Primary: Available, Secondary:
Secondary blob service endpo
Location
West US, East US
Replication status
Live
Subscription (change)
Pay-As-You-Go
Last synchronized
9/7/2017, 5:08:06 PM
Subscription ID
a3eee433-bf80-45b8-acae-b581d33f6a40
new
Container
Upload
Refresh
Delete container
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

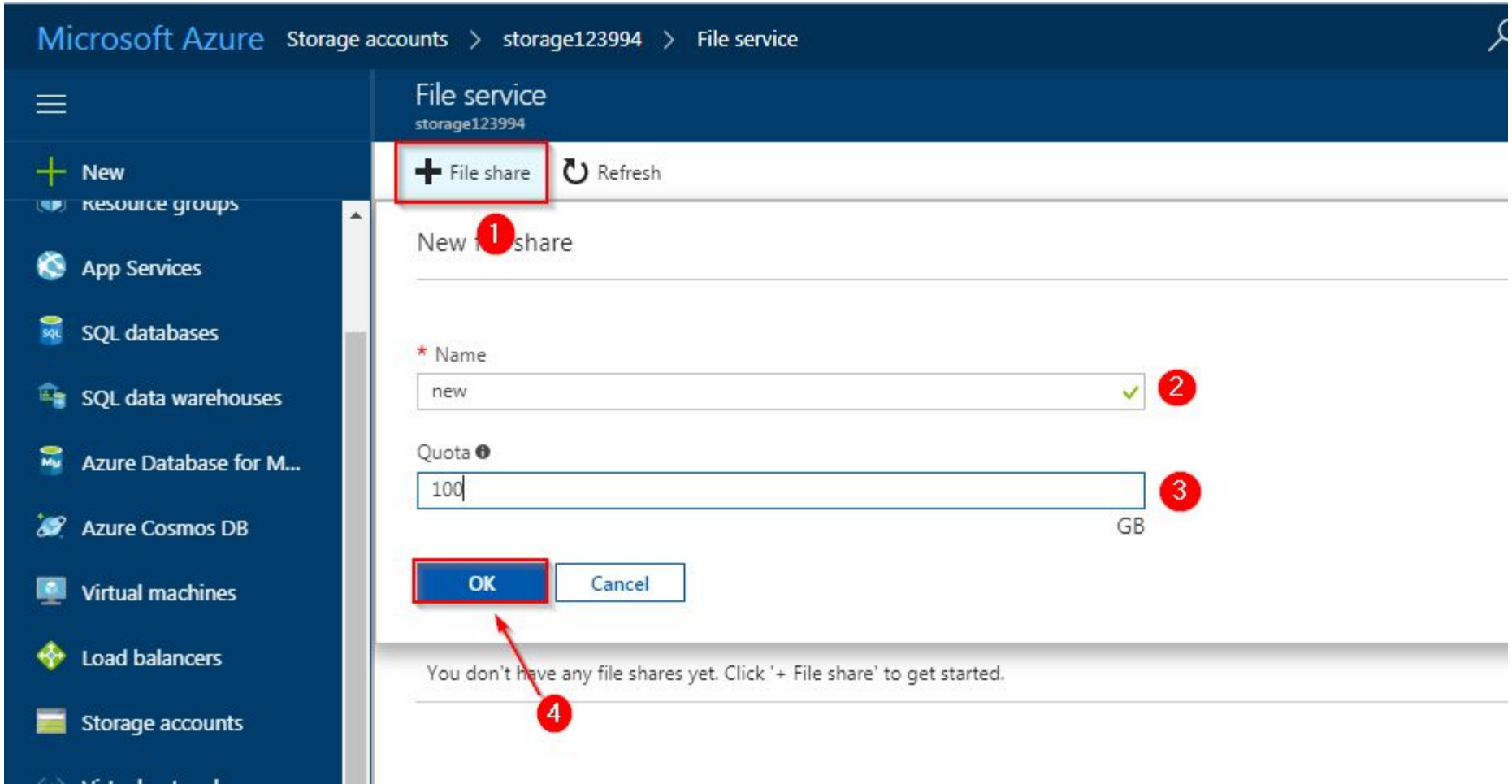


https://www.edureka.co/blog/azure-storage-tutorial/

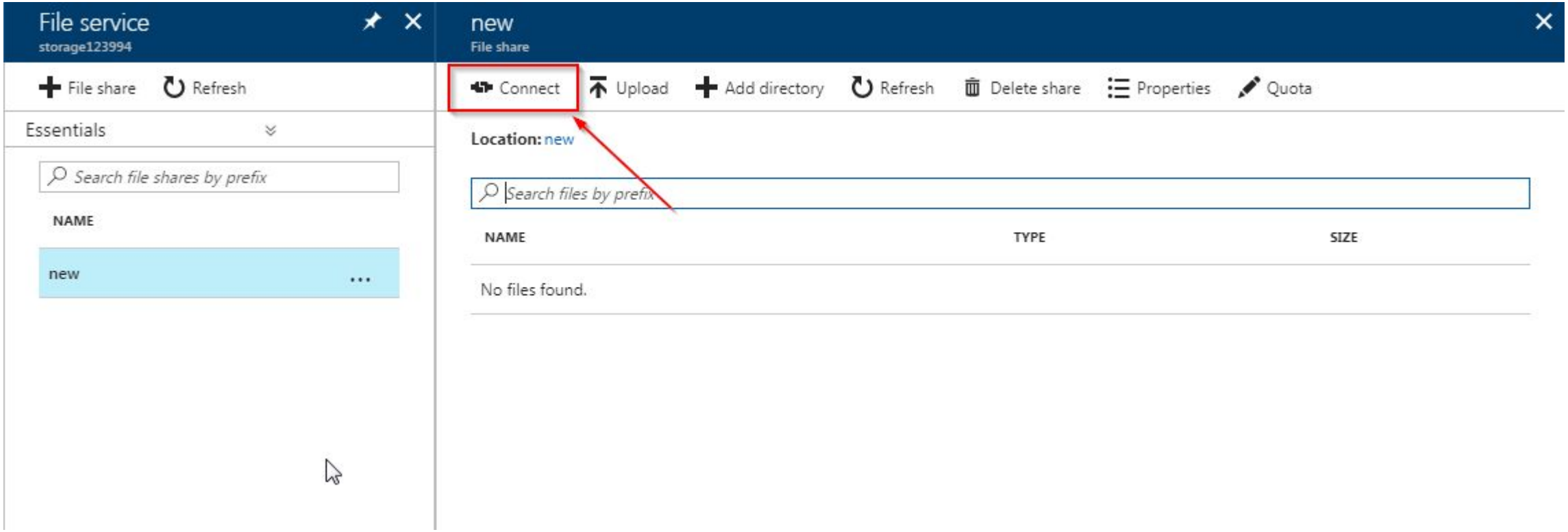
- Part 2: The file service in azure.
- The File Service in Azure uses the SMB (Server Message Block)3.0 protocol for file transfers, this service can be attached to your windows OS as if it was an external drive.
- Step 1: Go to your storage account overview page, and select the file service

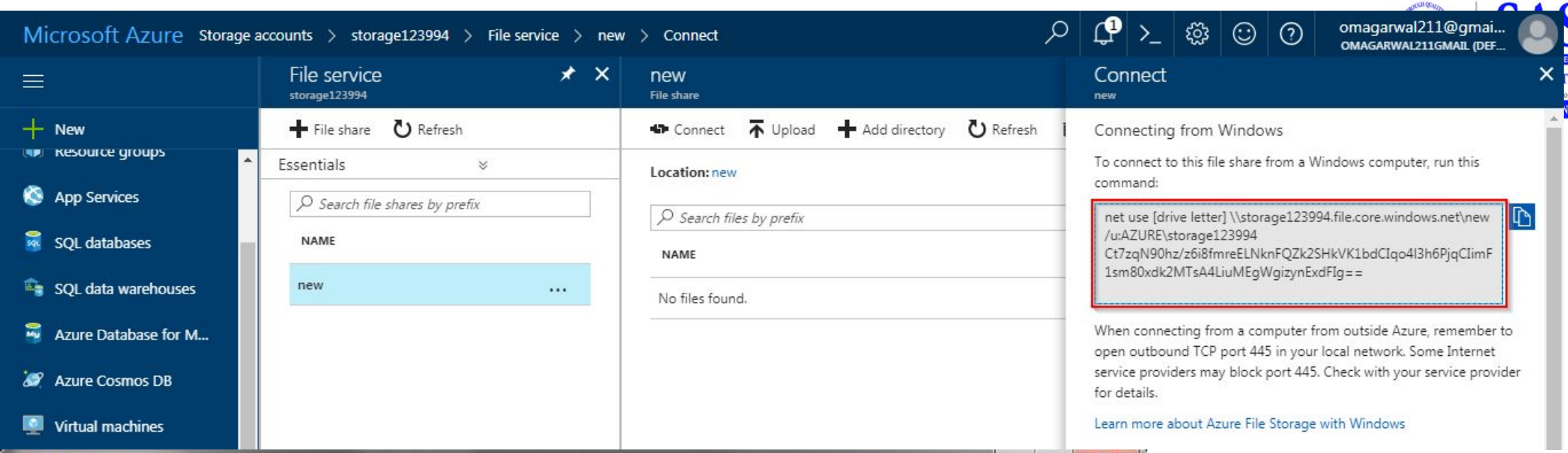


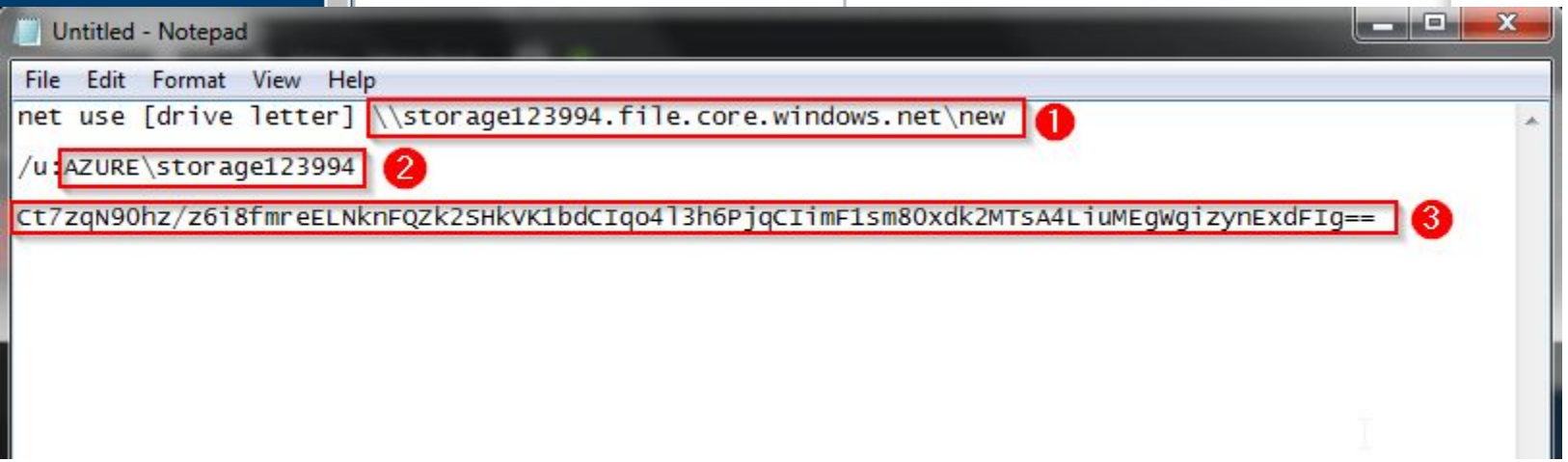
- Step 2: On the next page, enter the name of your file instance, and desired size of your instance. Finally, click on OK.



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

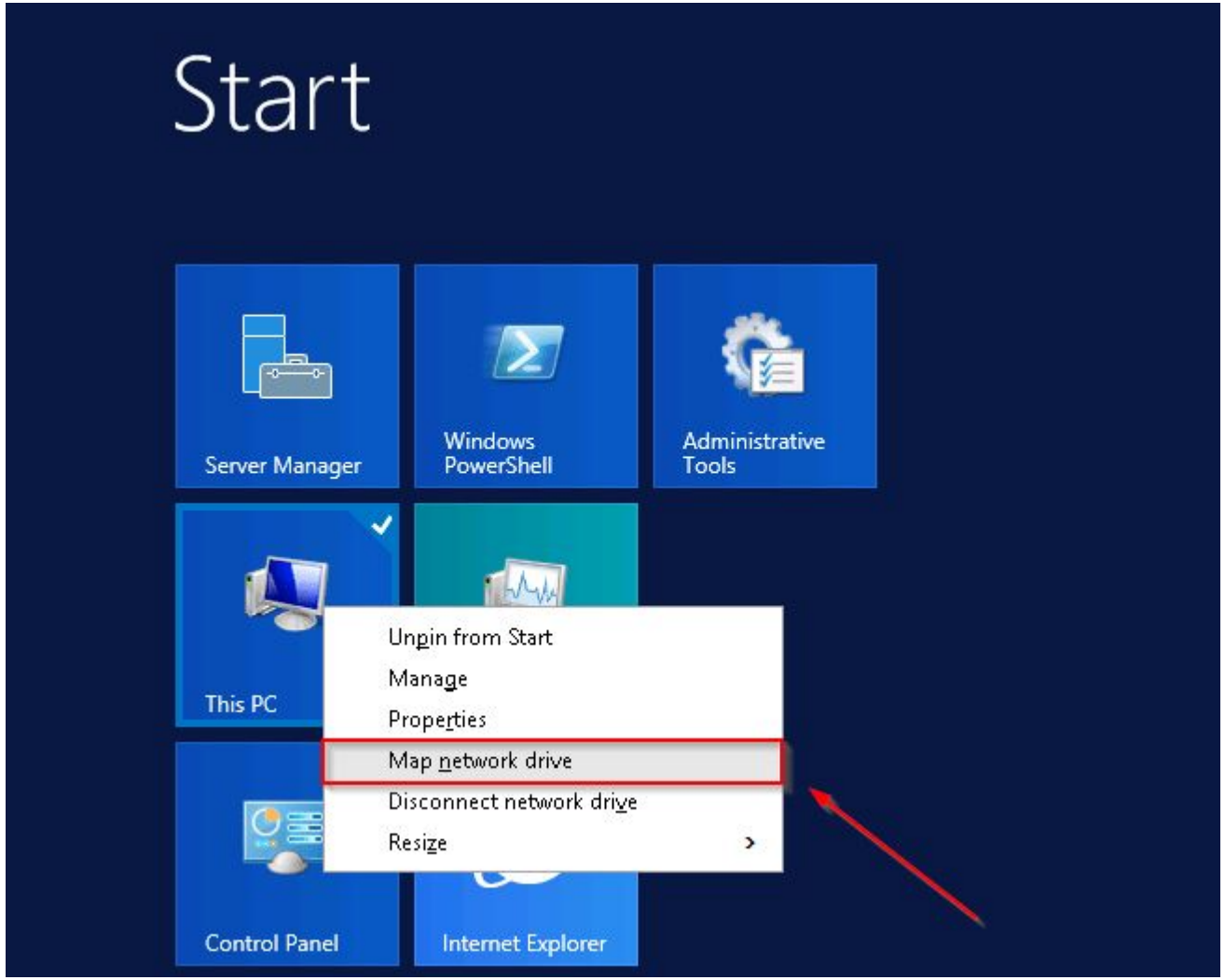




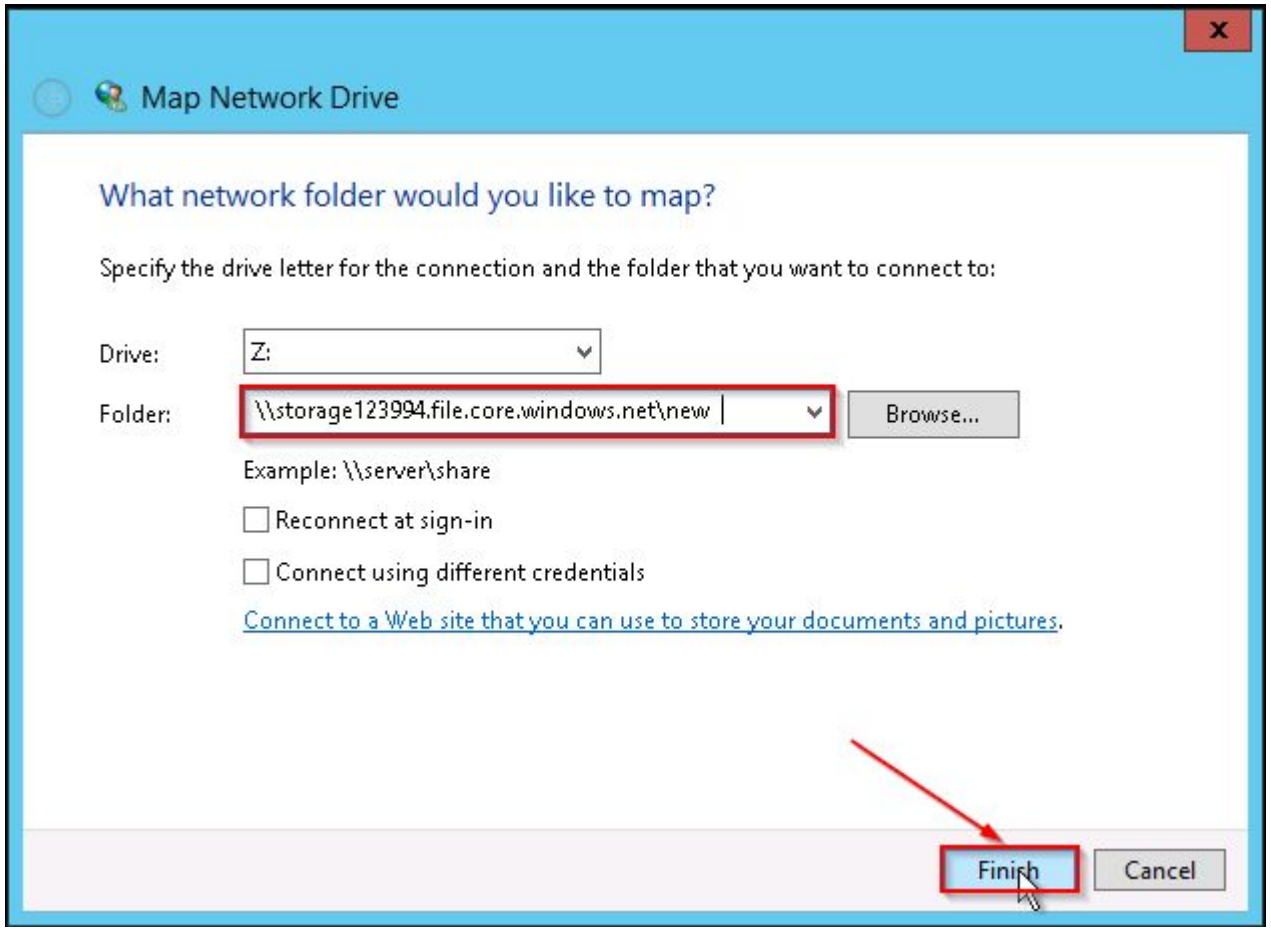


- The first point is the address column
- The second point is the user name
- The third point is your password

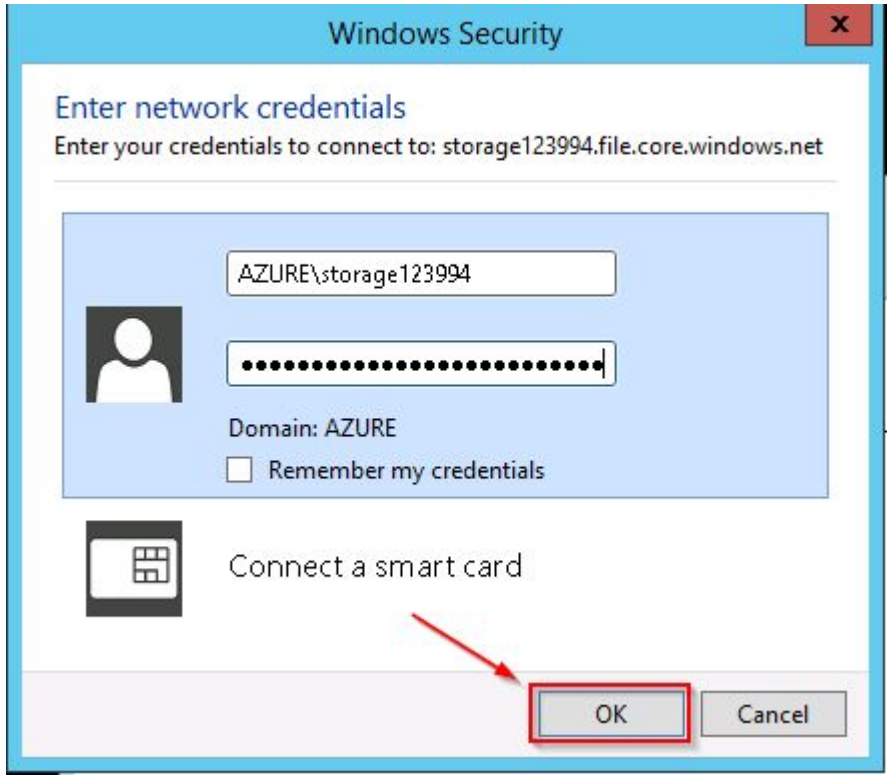
- Step 4: Right Click on your my computer icon, on your desktop and click on Map Network Drive.



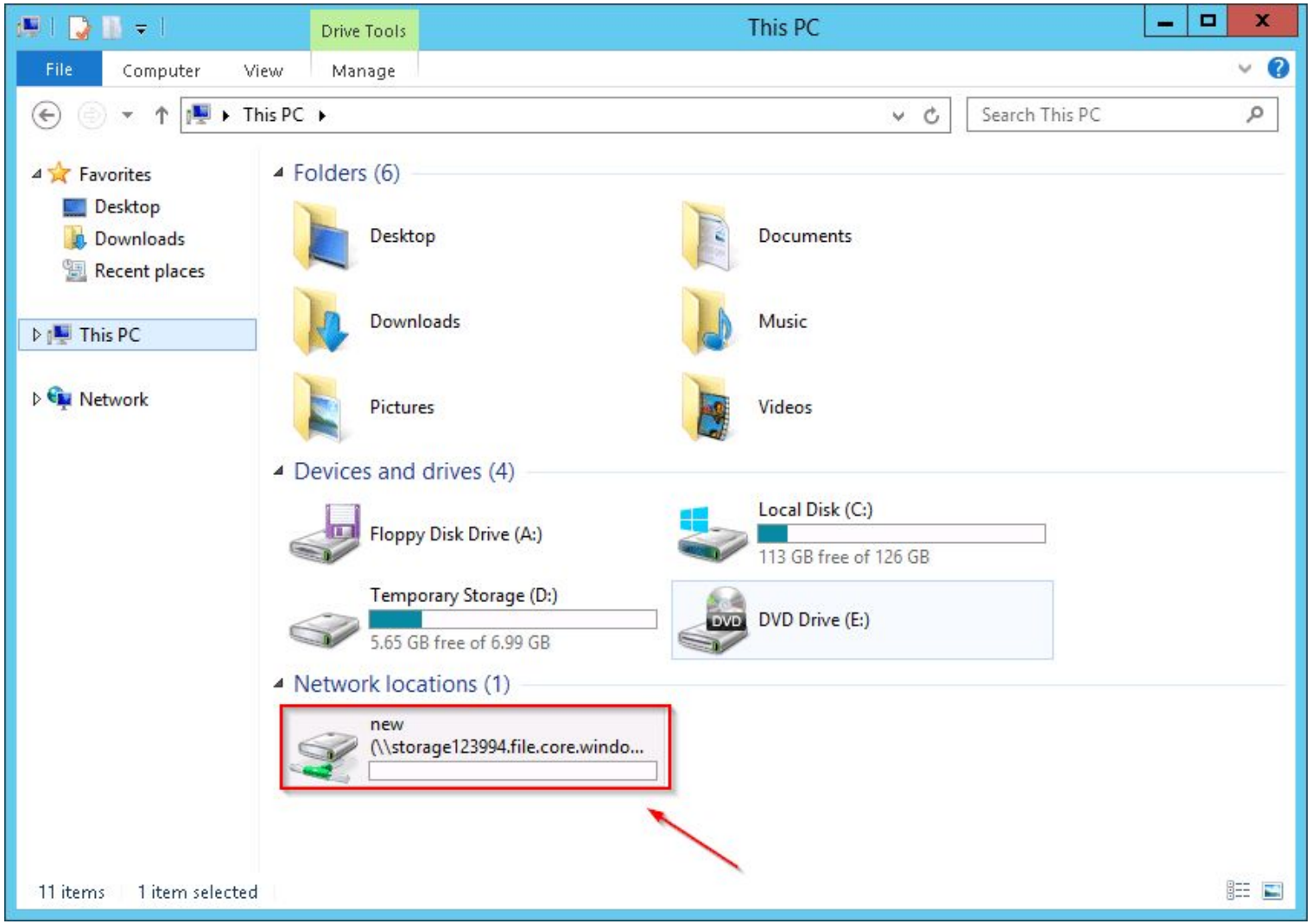
- Step 5: Enter the first point that you copied from your notepad in the folder text box and click on finish.



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

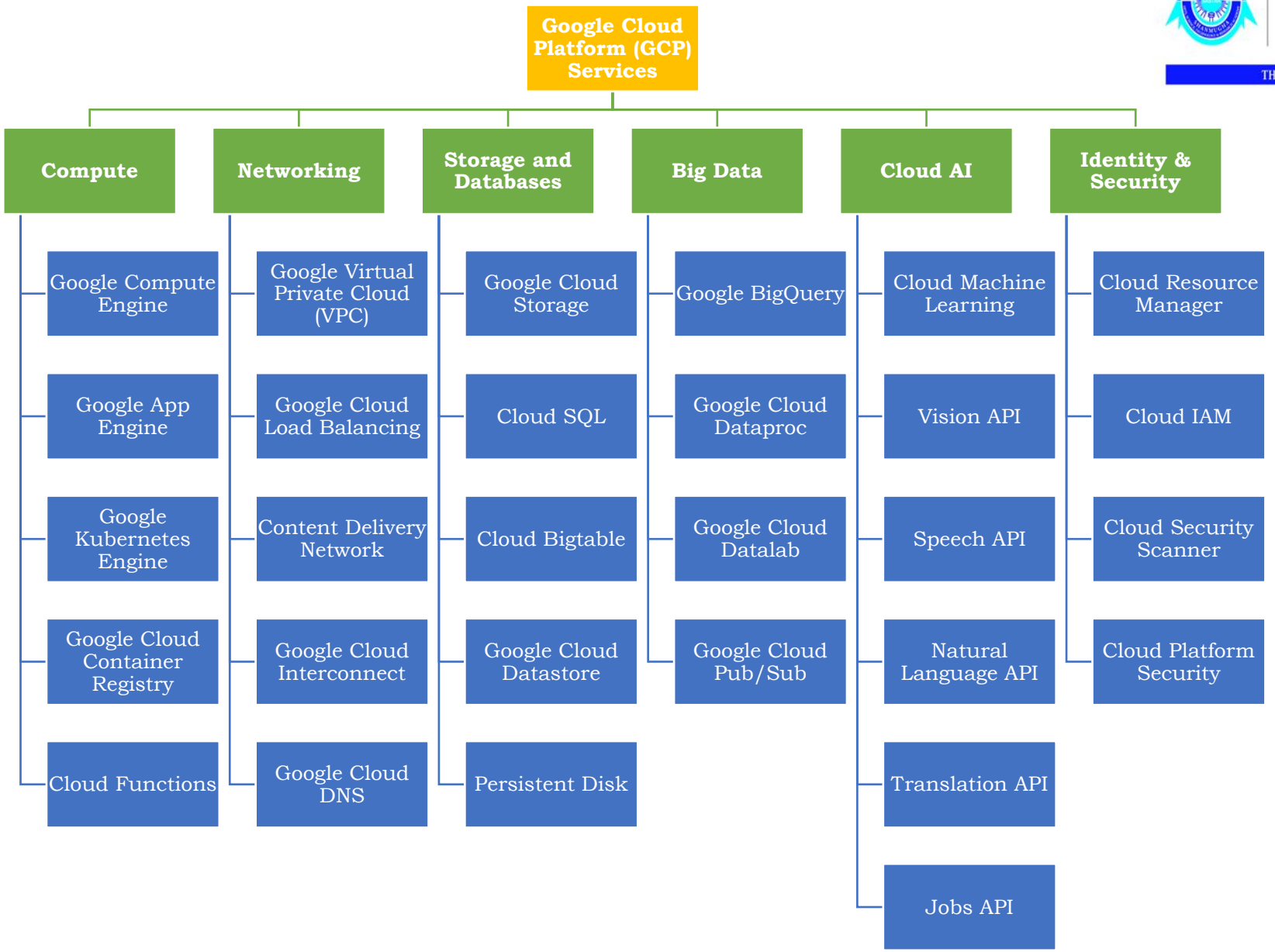


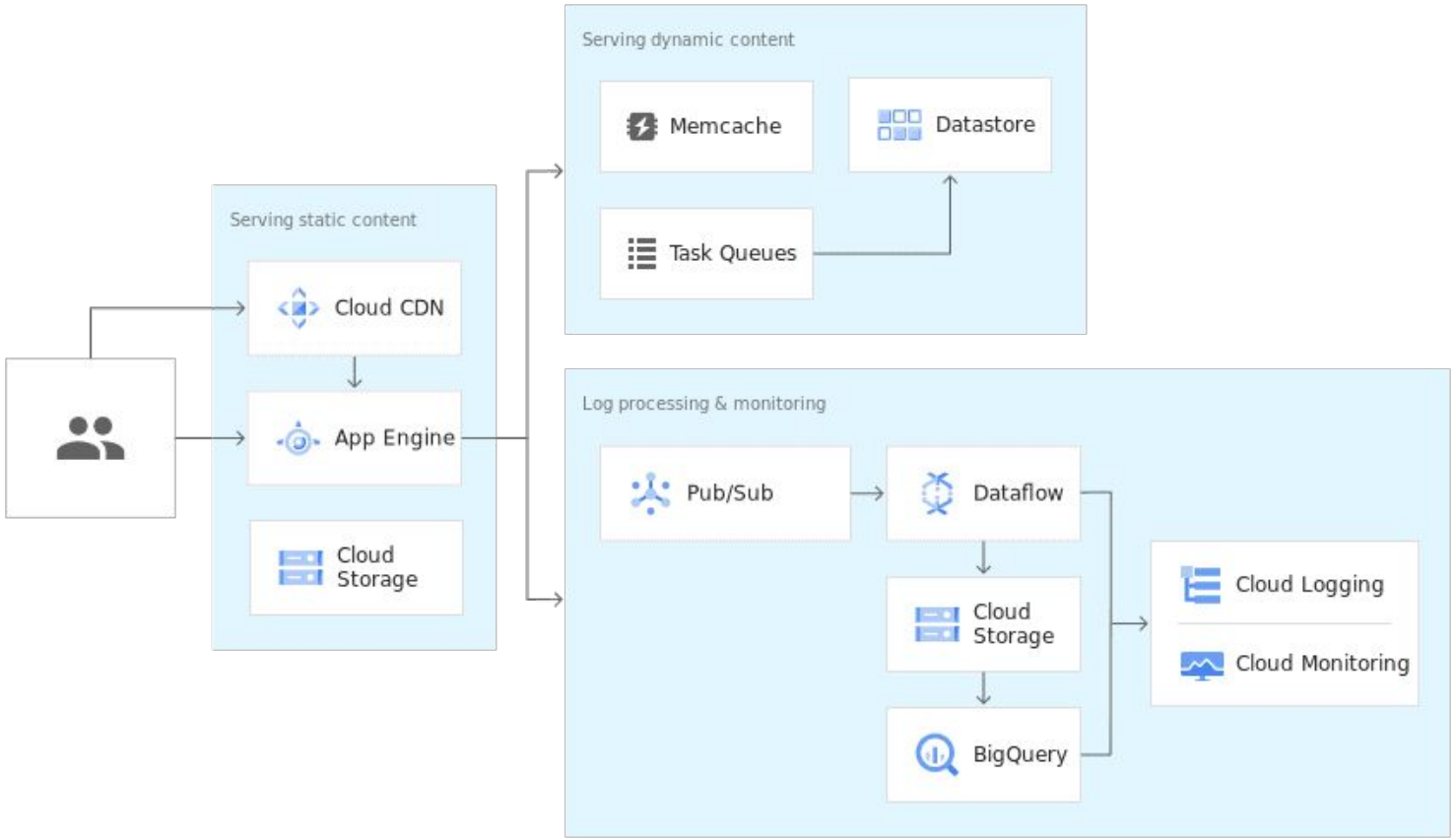
- Step 7: Congratulations! Your azure storage drive is ready. You can now use it, like any other drive on your computer



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

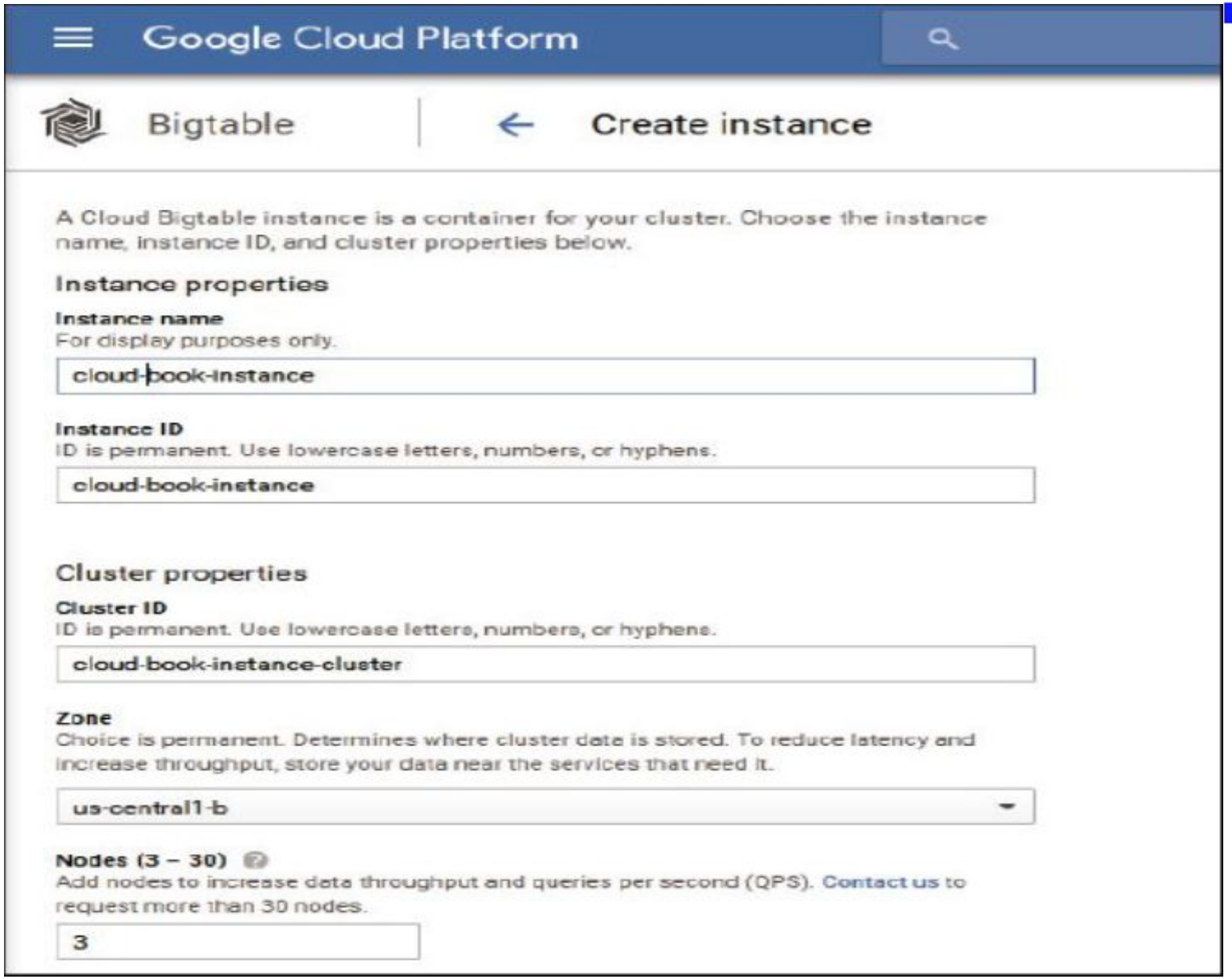


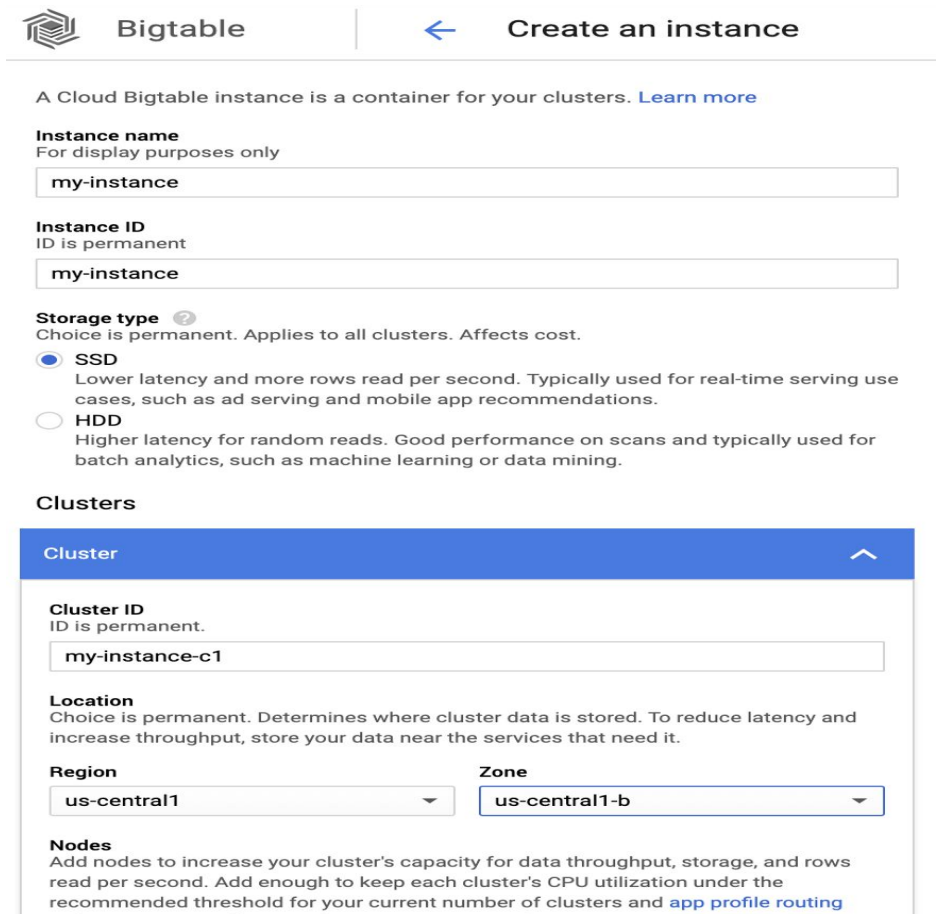
https://cloud.google.com/storage/docs/introduction

```
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()
```

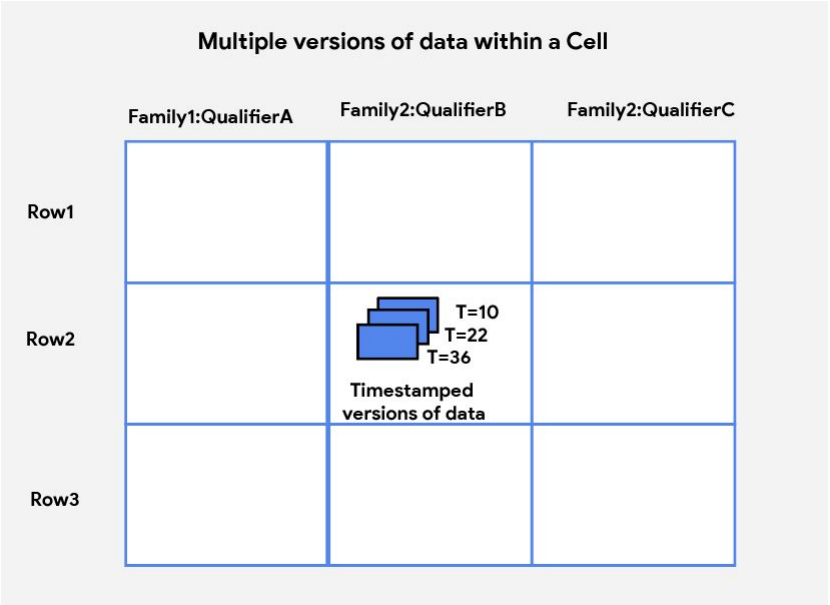
Google Bigtable

- 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
us-central1
Zone
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.

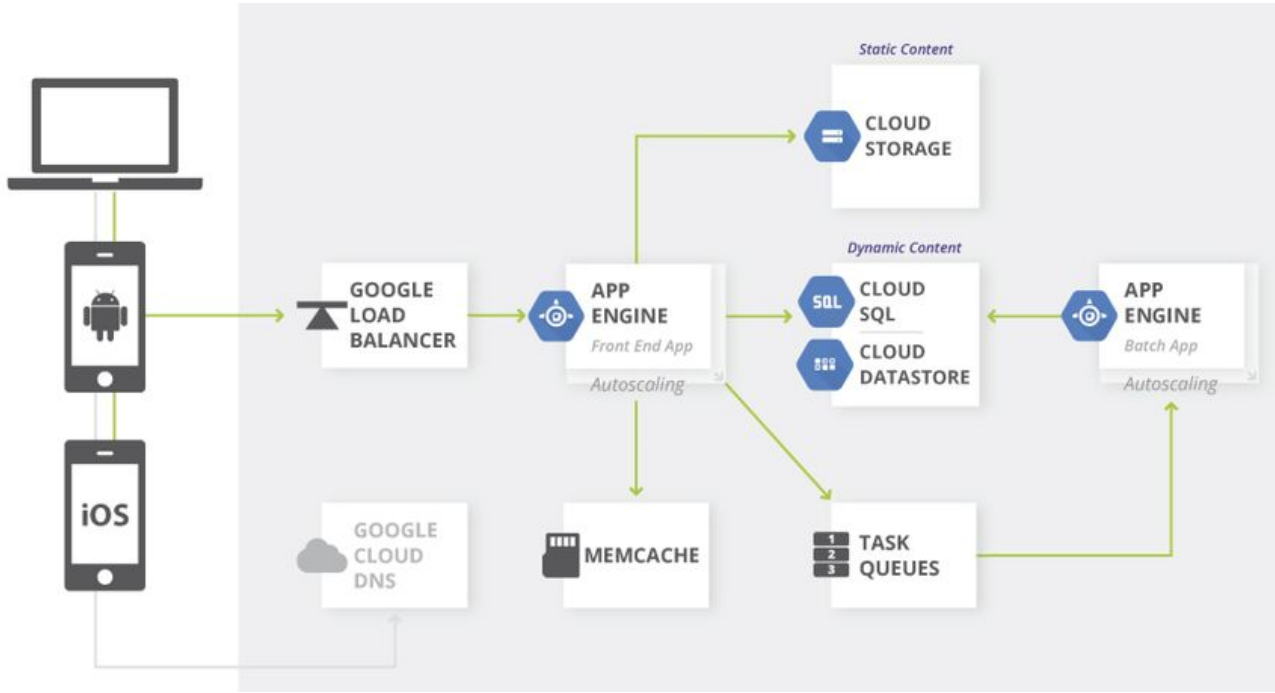
```
entity = datastore.Entity(key=key)
entity['experiment-name'] = 'experiment name'
entity['date'] = 'the date'
entity['description'] = 'the text describing the
experiment'
entity['url'] = 'the url'
clientds.put(entity)
```

```
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)
```

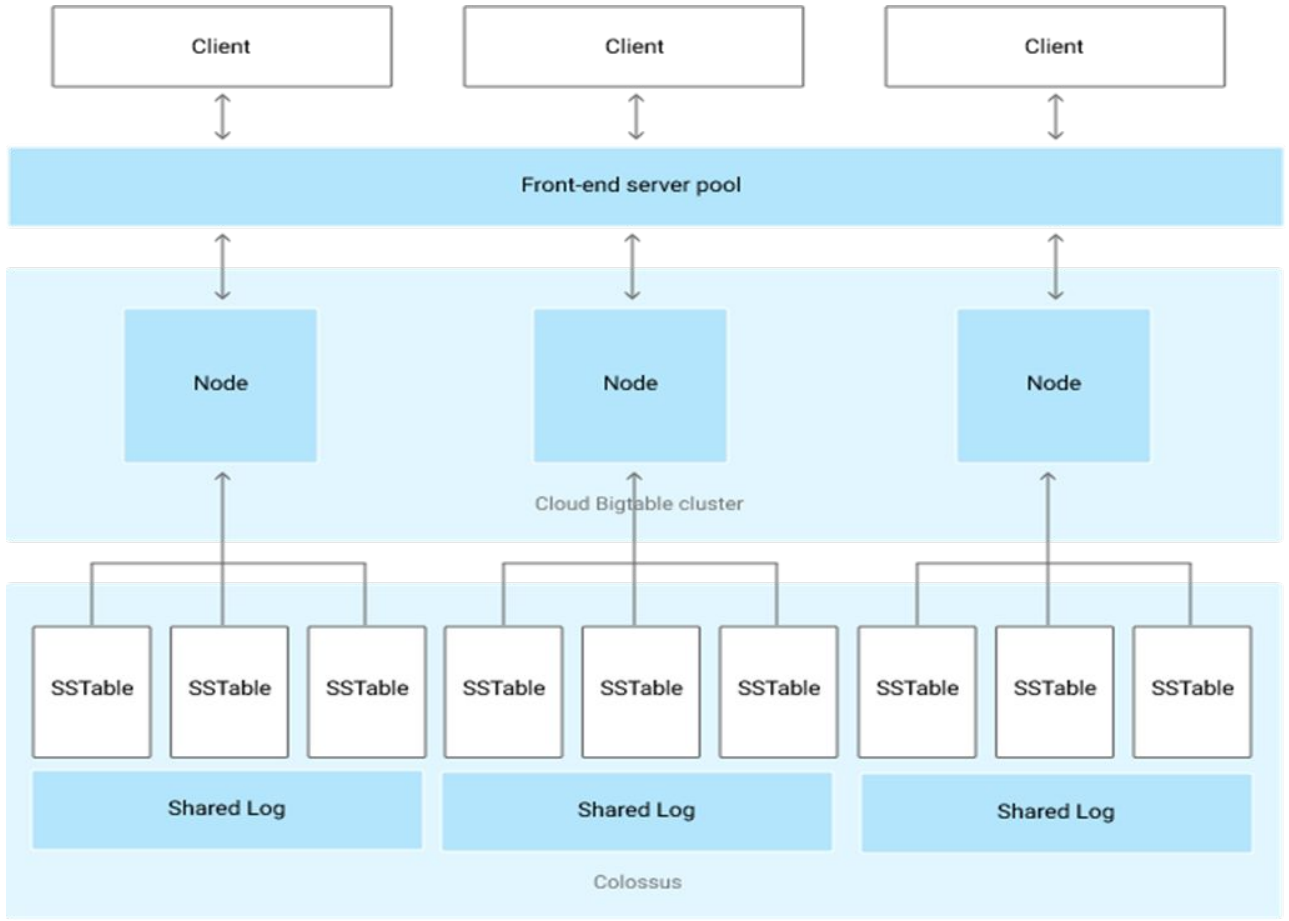
GOOGLE APP ENGINE

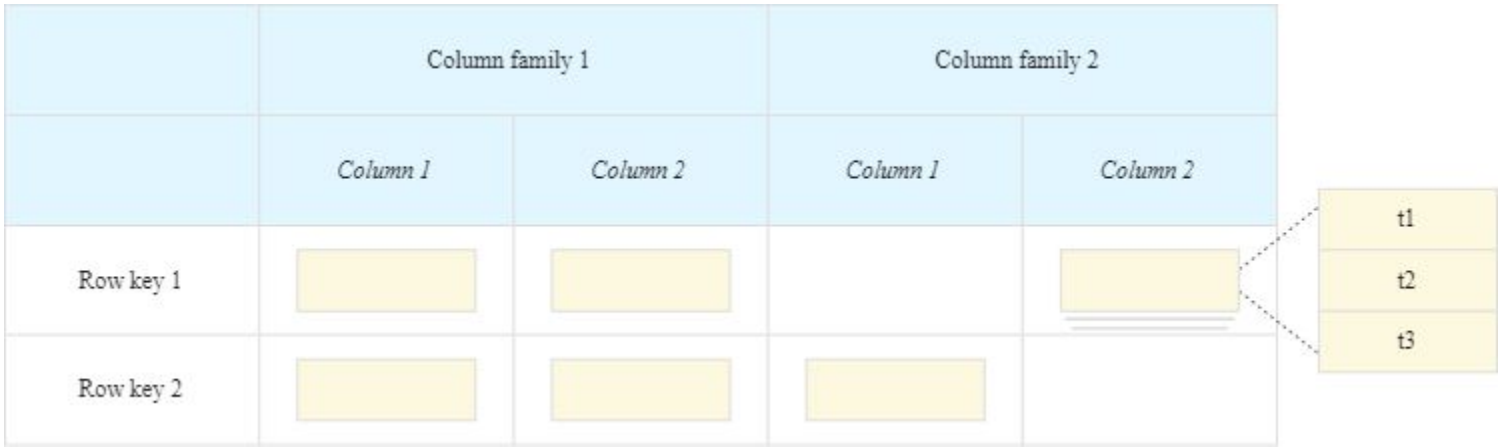
- Google App Engine is a PaaS solution that enables users to host their own applications on the same or similar infrastructure as Google Docs, Google Maps and other popular Google services



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)