FOUNDATION

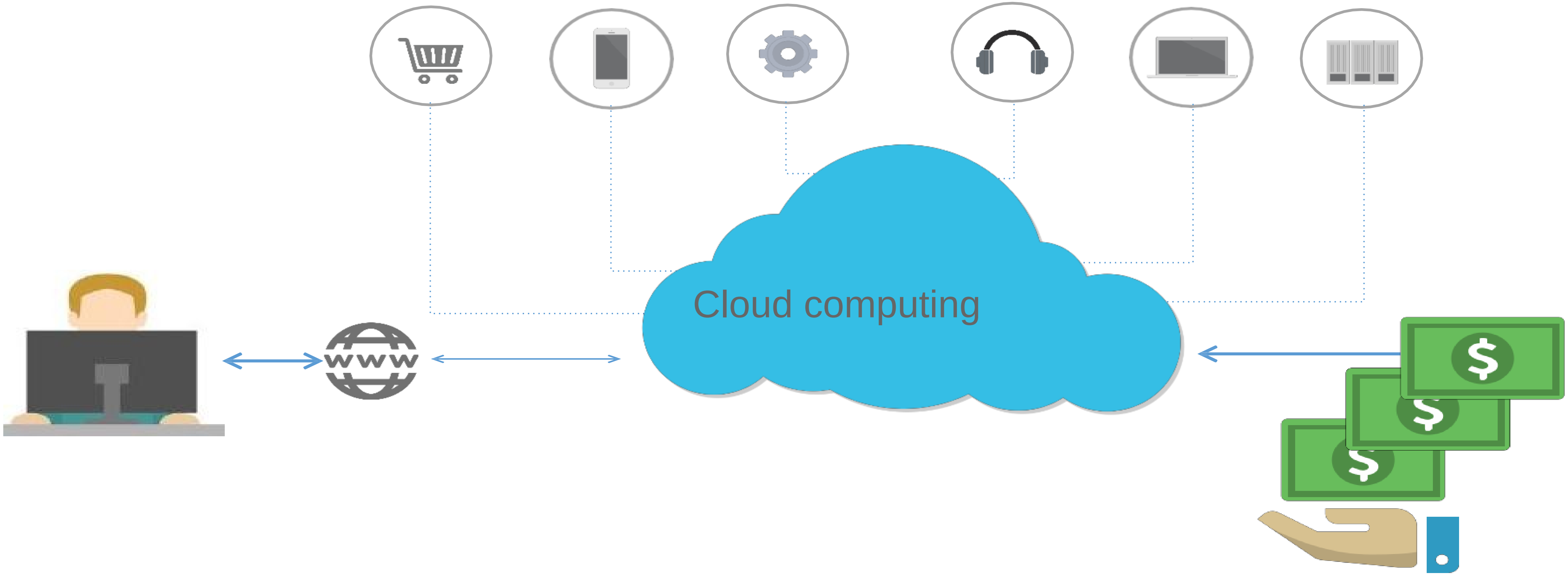
Cloud Basics

CLOUD COMPUTING OVERVIEW

Cloud computing refers to on-demand provisioning of IT resources and applications through the Internet.

Cloud computing facilitates:

- Quick access to cost-efficient and flexible IT resources
- Accessing servers, databases, storage media, and a variety of application services on the World Wide Web



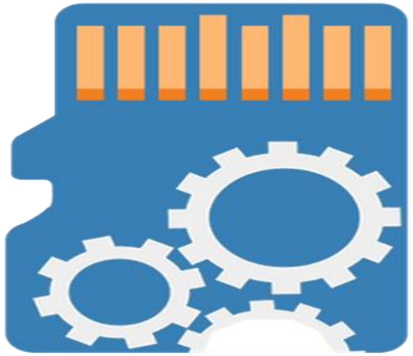
Cloud computing

Pay as
you go

Cloud computing supports the following:

Servers

Databases



Storage media

Services on World Wide Web

CLOUD COMPUTING

- Cloud computing providers such as Amazon Web Services possess and maintain the hardware required for different services.

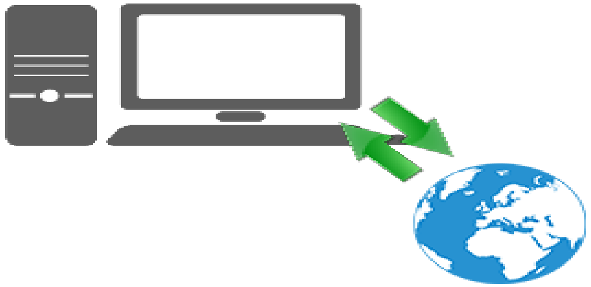
With Cloud computing, you don't need:

Hardware investments

With Cloud computing, you only need to:

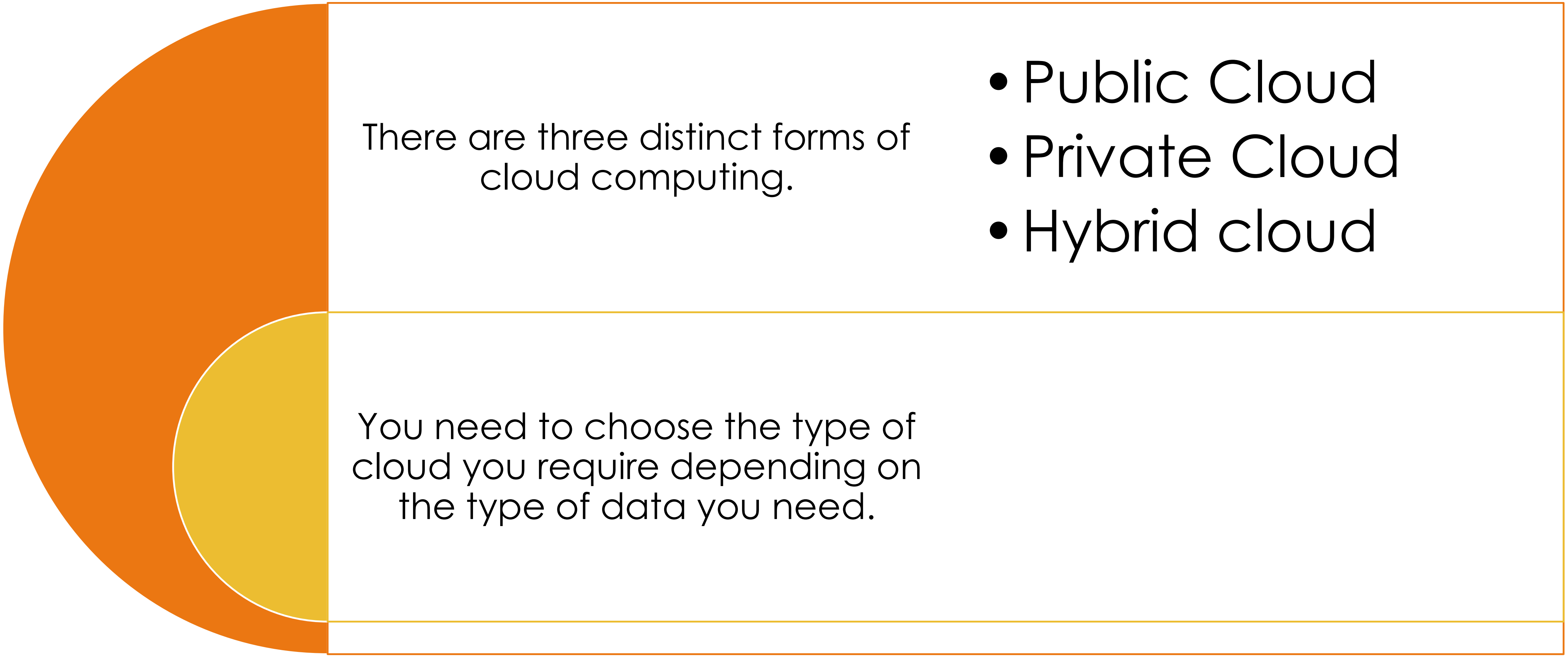
Determine type and size of resources

Empower the latest business project or idea



Identify files to be stored as backup on the Internet

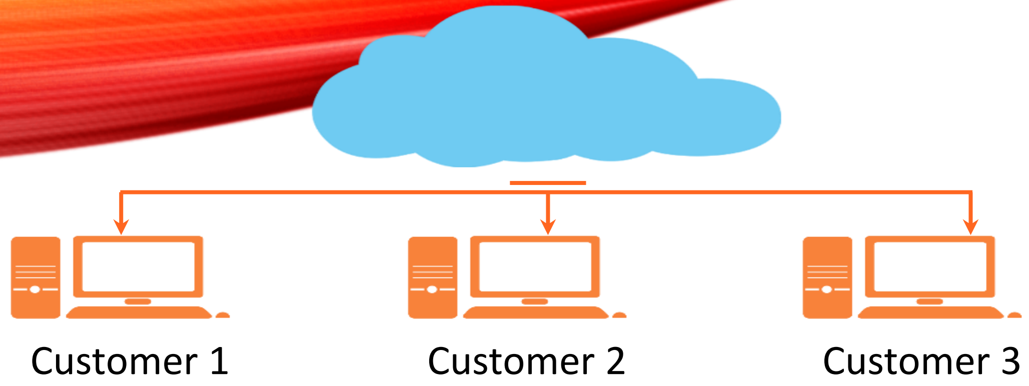
FORMS OF CLOUD COMPUTING



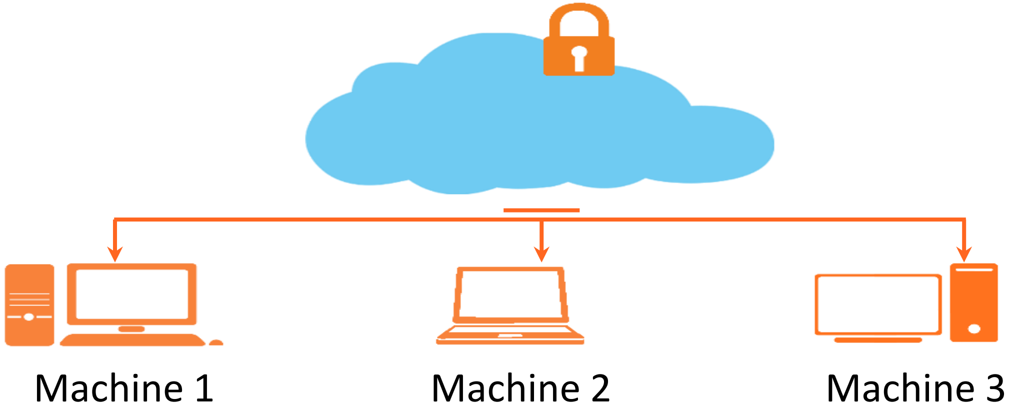
There are three distinct forms of cloud computing.

- Public Cloud
- Private Cloud
- Hybrid cloud

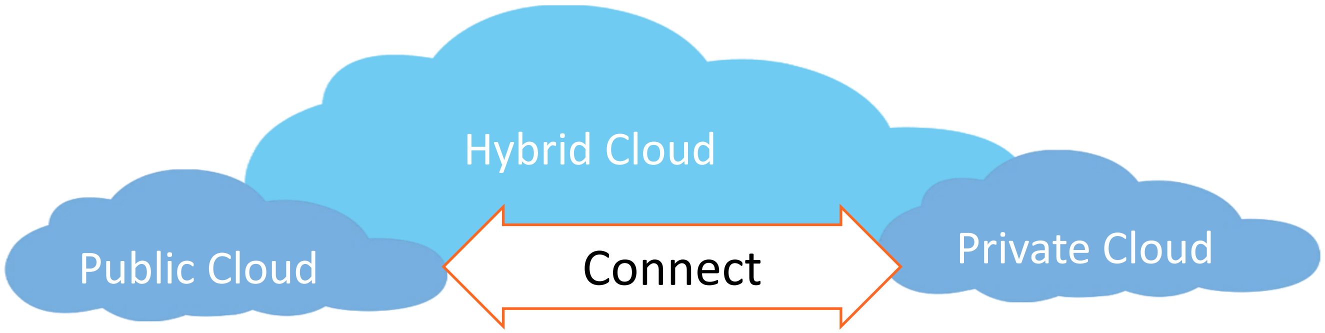
You need to choose the type of cloud you require depending on the type of data you need.



Public Cloud



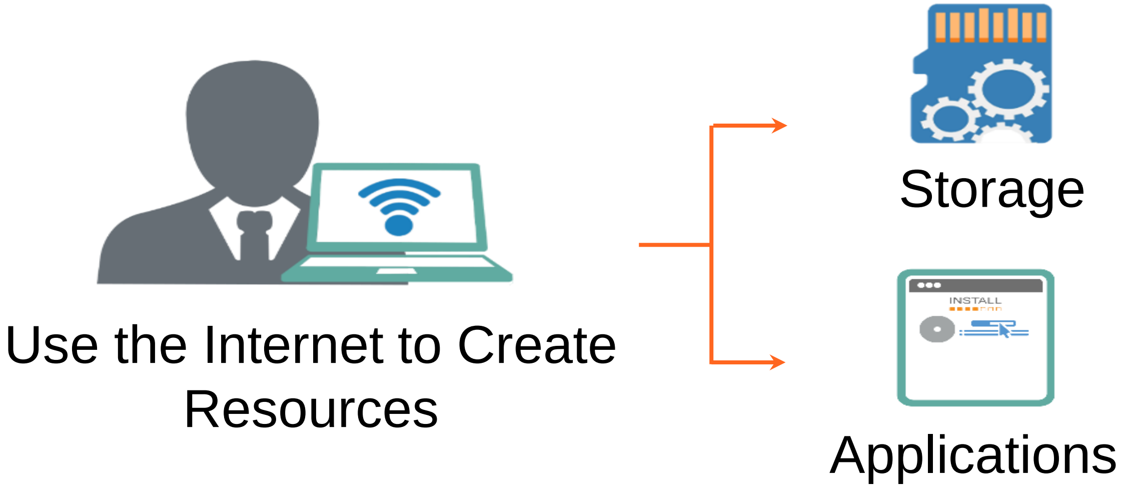
Private Cloud



Hybrid Cloud

PUBLIC CLOUD

- Public cloud service providers use the Internet to create resources, such as storage and applications available to the public.



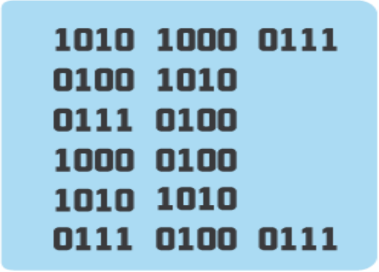
Use the Internet to Create Resources

Storage

Applications

A PUBLIC CLOUD IS IDEAL FOR:

Sharing Non-Secured Applications

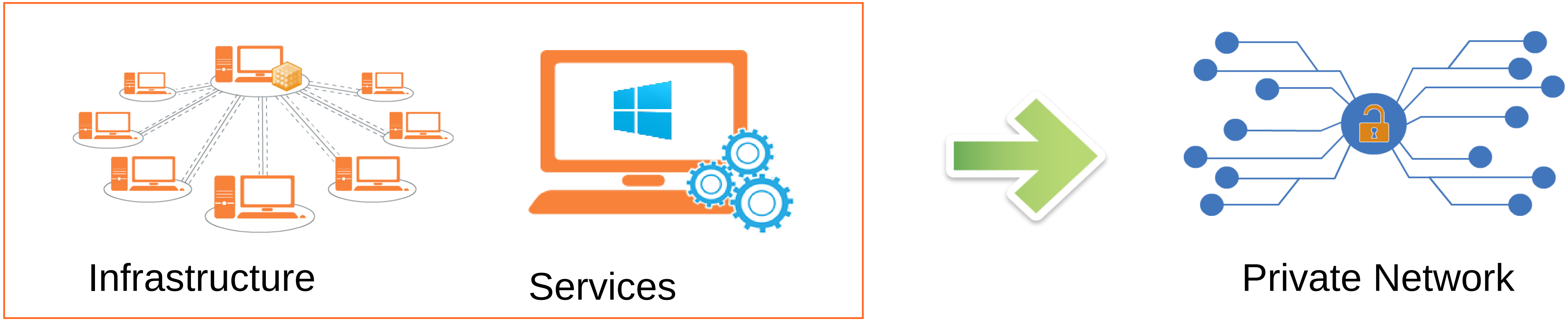


Testing Application Code

Examples of public cloud: Windows Azure Services Platform, Amazon Elastic Compute Cloud or EC2, Sun Cloud, and IBM's Blue Cloud

PRIVATE CLOUD

- A private cloud has an architecture similar to a data center, and a company owns it to ensure provisioning, monitoring, automation, and scalability.

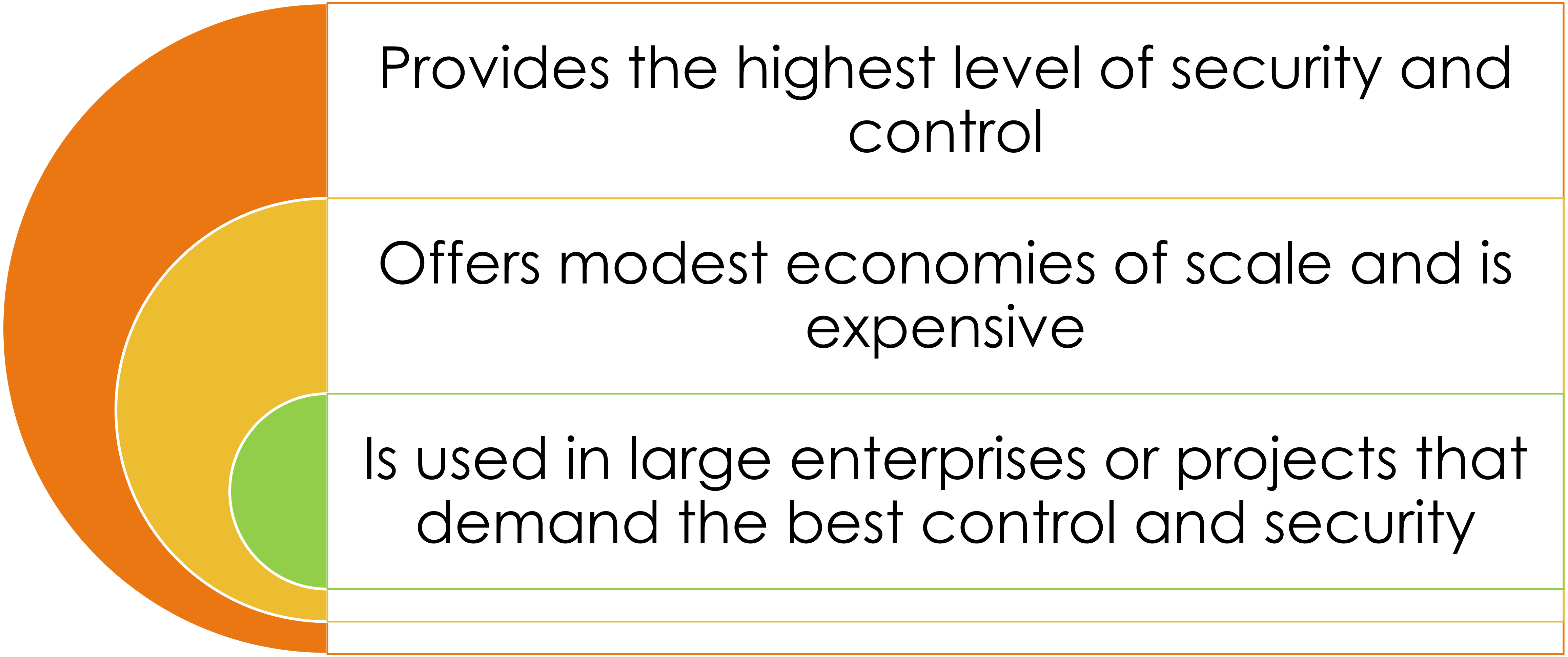


Infrastructure

Services

Private Network

FEATURES OF PRIVATE CLOUD



Provides the highest level of security and control

Offers modest economies of scale and is expensive

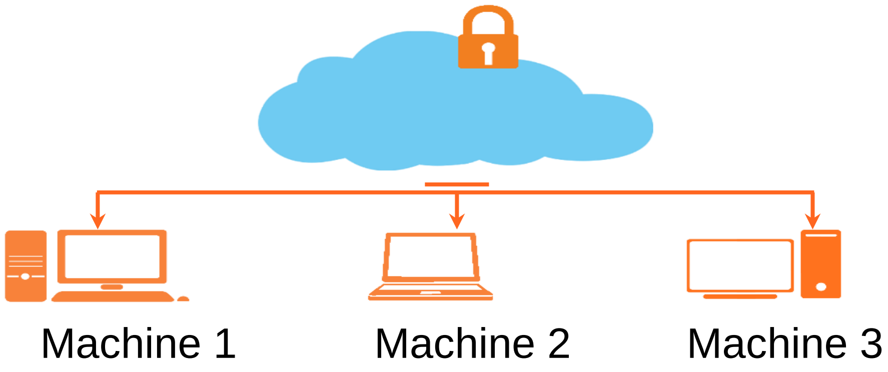
Is used in large enterprises or projects that demand the best control and security

HYBRID CLOUD

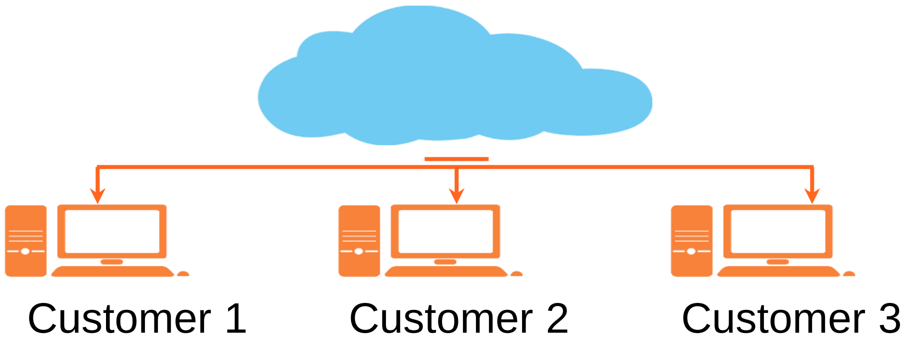
Hybrid cloud is a combination of public and private cloud services offered by multiple providers.

Hybrid cloud caters to different market verticals. While a public cloud allows you to interact with clients, a private cloud helps in securing their data.

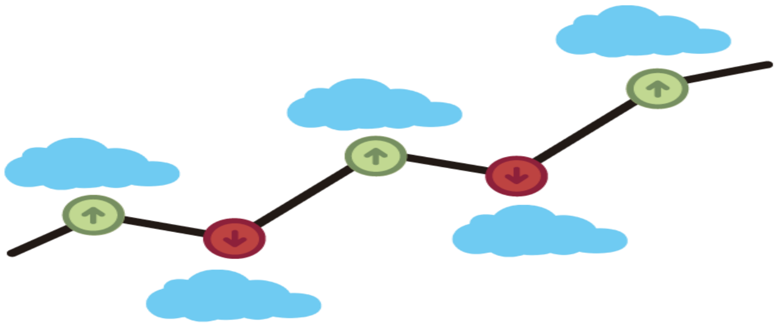
USAGE: HYBRID CLOUD



Retains control



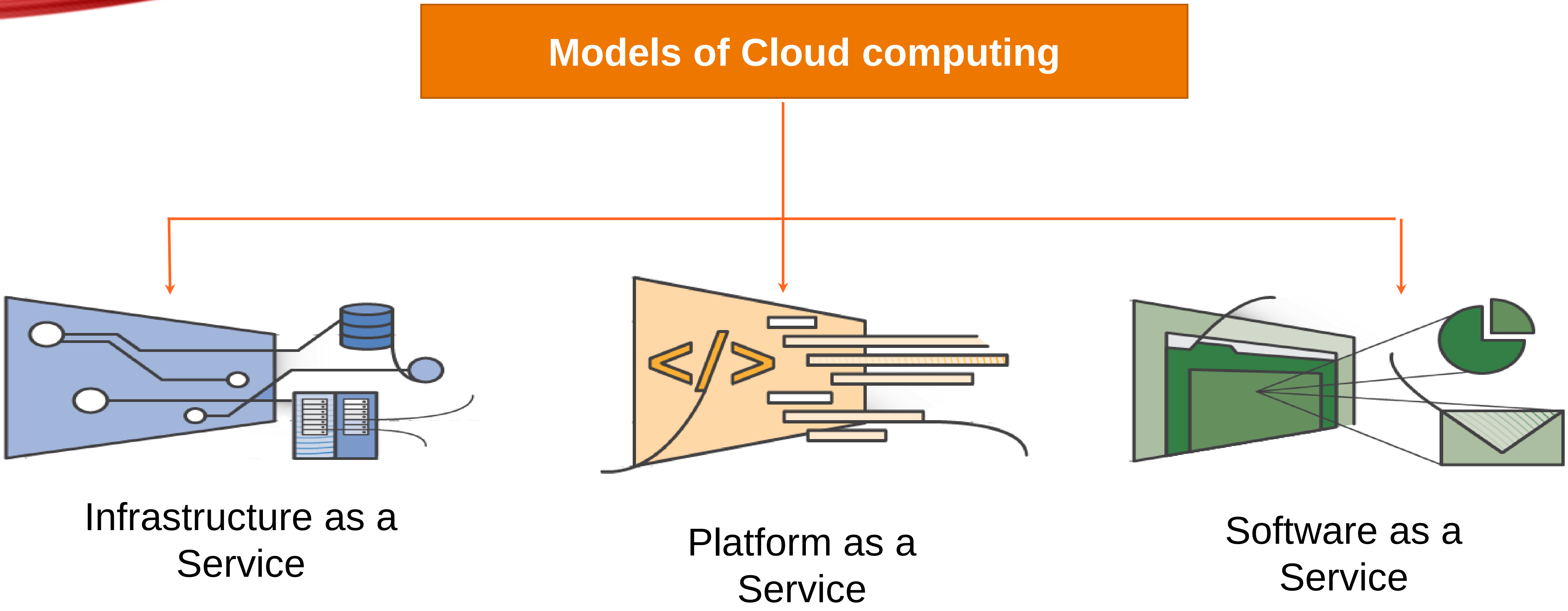
Migrates data
to the public cloud



Tracks different
applications in each cloud

MODELS OF CLOUD COMPUTING

- The three models of cloud computing are Infrastructure as a Service, Platform as a Service, and Software as a Service.
- Each of these models comes with different levels of management, control, and flexibility.



Models of Cloud computing

Infrastructure as a Service

Platform as a Service

Software as a Service

IAAS

- **Infrastructure as a Service (IaaS)** is a cloud computing model that provides virtualized computing resources over the internet.
- It allows businesses to rent IT infrastructure—servers, storage, networking, and other resources—from a cloud provider on a pay-as-you-go basis.

HOW IAAS WORKS?

Virtualization:

- IaaS providers use virtualization technologies to deliver computing resources to customers.
- This includes virtual machines (VMs) that can run different operating systems and applications.

Resource Pooling:

- Resources are pooled together and allocated dynamically to meet customer demands.
- This ensures optimal use of hardware and increases flexibility.

HOW IAAS WORKS?

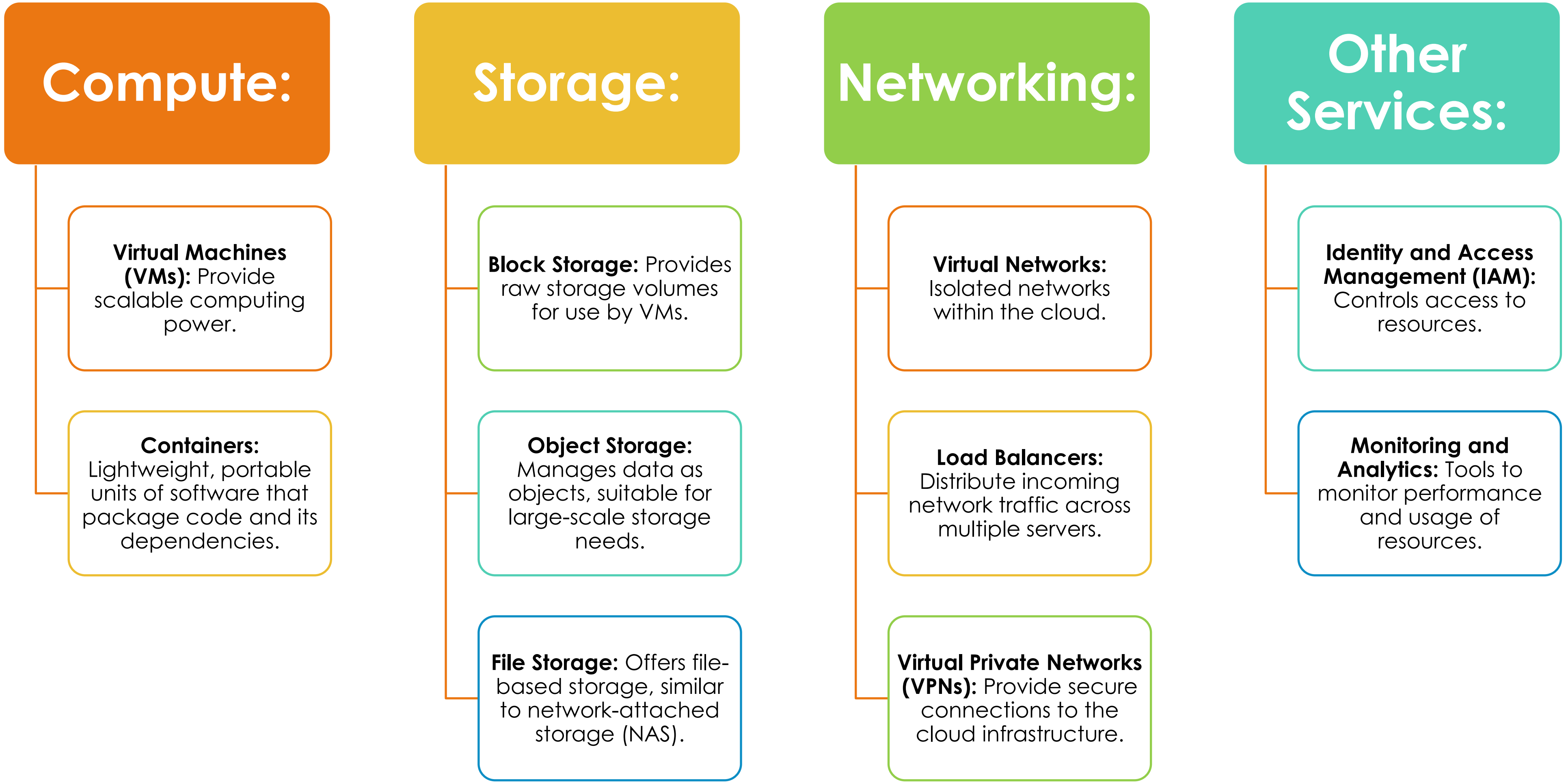
Self-Service and Automation:

- Customers can provision and manage resources through a web-based dashboard or API, allowing for automated scaling and management.

Billing:

- IaaS follows a pay-as-you-go pricing model, charging customers based on their usage of resources like compute power, storage, and bandwidth.

COMPONENTS



Compute:

- **Virtual Machines (VMs):** Provide scalable computing power.
- **Containers:** Lightweight, portable units of software that package code and its dependencies.

Storage:

- **Block Storage:** Provides raw storage volumes for use by VMs.
- **Object Storage:** Manages data as objects, suitable for large-scale storage needs.
- **File Storage:** Offers file-based storage, similar to network-attached storage (NAS).

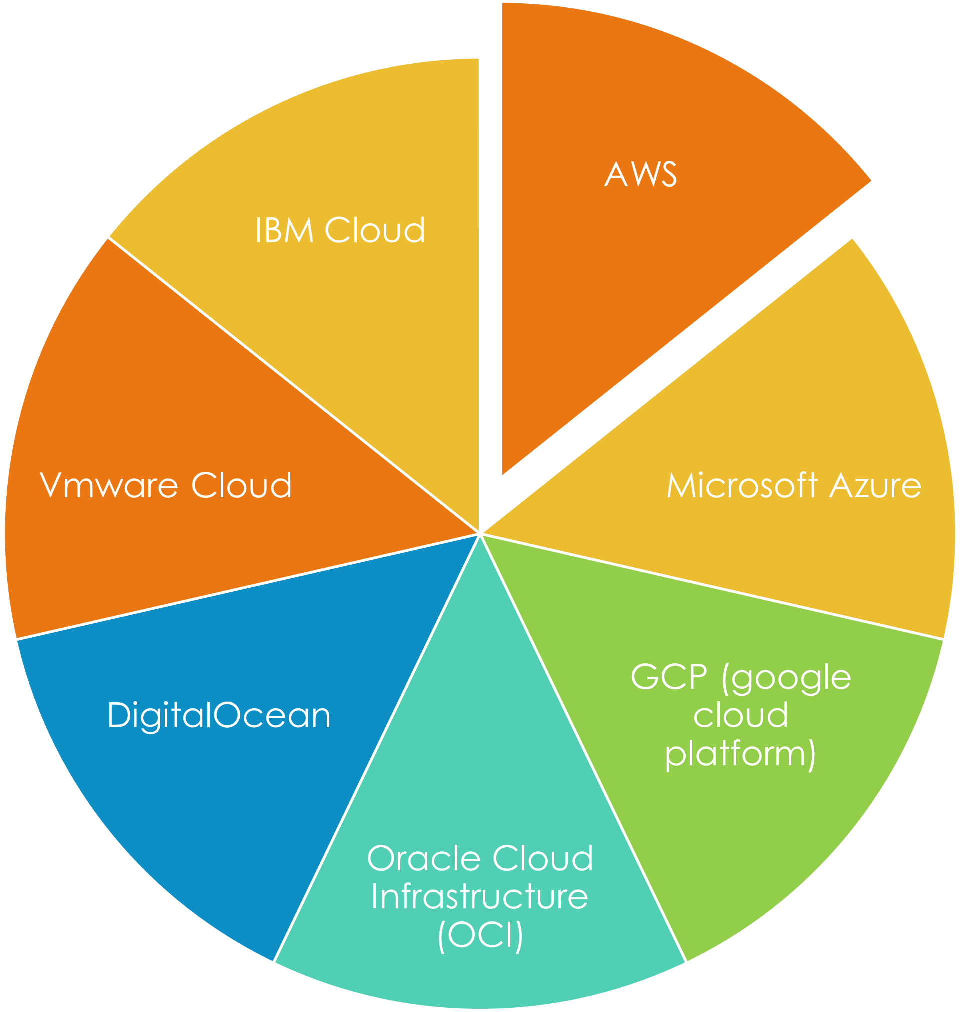
Networking:

- **Virtual Networks:** Isolated networks within the cloud.
- **Load Balancers:** Distribute incoming network traffic across multiple servers.
- **Virtual Private Networks (VPNs):** Provide secure connections to the cloud infrastructure.

Other Services:

- **Identity and Access Management (IAM):** Controls access to resources.
- **Monitoring and Analytics:** Tools to monitor performance and usage of resources.

IAAS EXAMPLES


AWS
Microsoft Azure
GCP (google cloud platform)
Oracle Cloud Infrastructure (OCI)
DigitalOcean
Vmware Cloud
IBM Cloud

PAAS

- **Platform as a Service (PaaS)** is a cloud computing model that provides a platform allowing customers to develop, run, and manage applications without dealing with the complexity of building and maintaining the underlying infrastructure.

HOW PASS WORKS

Application Development Environment:

- PaaS provides a complete development and deployment environment in the cloud, with resources that enable you to deliver everything from simple cloud-based apps to sophisticated, cloud-enabled enterprise applications.

Integrated Development Tools:

- It includes tools for development, database management, business analytics, and more, often integrated into a single environment.

Scalability and Maintenance:

- PaaS handles the scalability and maintenance of the platform, allowing developers to focus on writing code and developing applications.

Collaboration:

- Multiple users can collaborate on projects from different locations.

COMPONENTS OF PAAS

Development Tools:

- **Integrated Development Environments (IDEs):** Tools like Visual Studio, Eclipse, and others that are available in the cloud.
- **Version Control:** Systems like Git that help manage code versions and collaboration.

Middleware:

- Software that connects different applications or services.

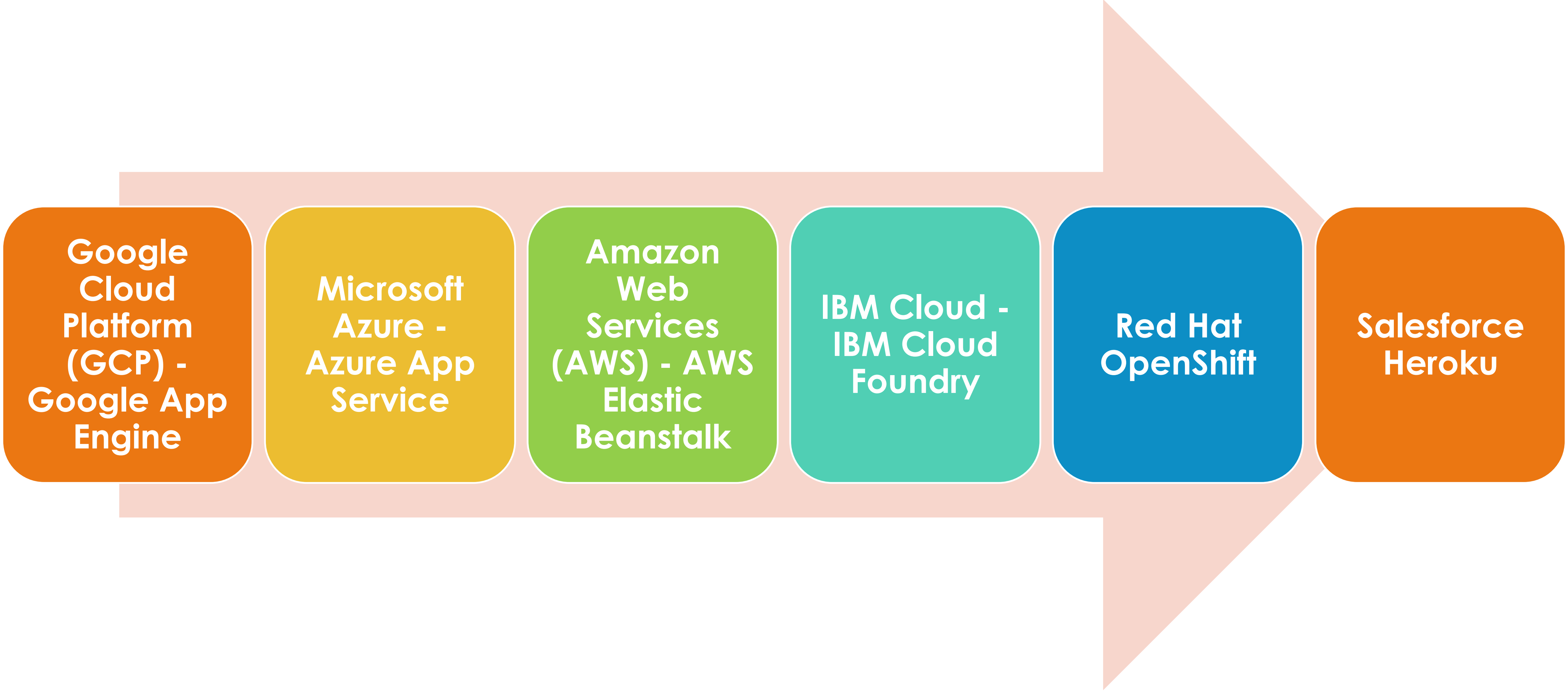
Database Management:

- **Relational Databases:** Such as MySQL, PostgreSQL.
- **NoSQL Databases:** Such as MongoDB, Cassandra.

Application Hosting:

- **Web Servers:** Platforms to host applications.
- **Application Containers:** Technologies like Docker for containerization.

PAAS PROVIDERS



Google Cloud Platform (GCP) - Google App Engine

Microsoft Azure - Azure App Service

Amazon Web Services (AWS) - AWS Elastic Beanstalk

IBM Cloud - IBM Cloud Foundry

Red Hat OpenShift

Salesforce Heroku

SAAS

- **Software as a Service (SaaS)** is a cloud computing model where software applications are delivered over the internet.
- Instead of installing and maintaining software, users can access it via the internet, freeing themselves from complex software and hardware management.

HOW SAAS WORKS

Access via Internet:

- Users access the software through a web browser or a thin client over the internet.

Subscription-Based:

- Typically, SaaS applications are subscription-based, with costs dependent on usage levels or features.

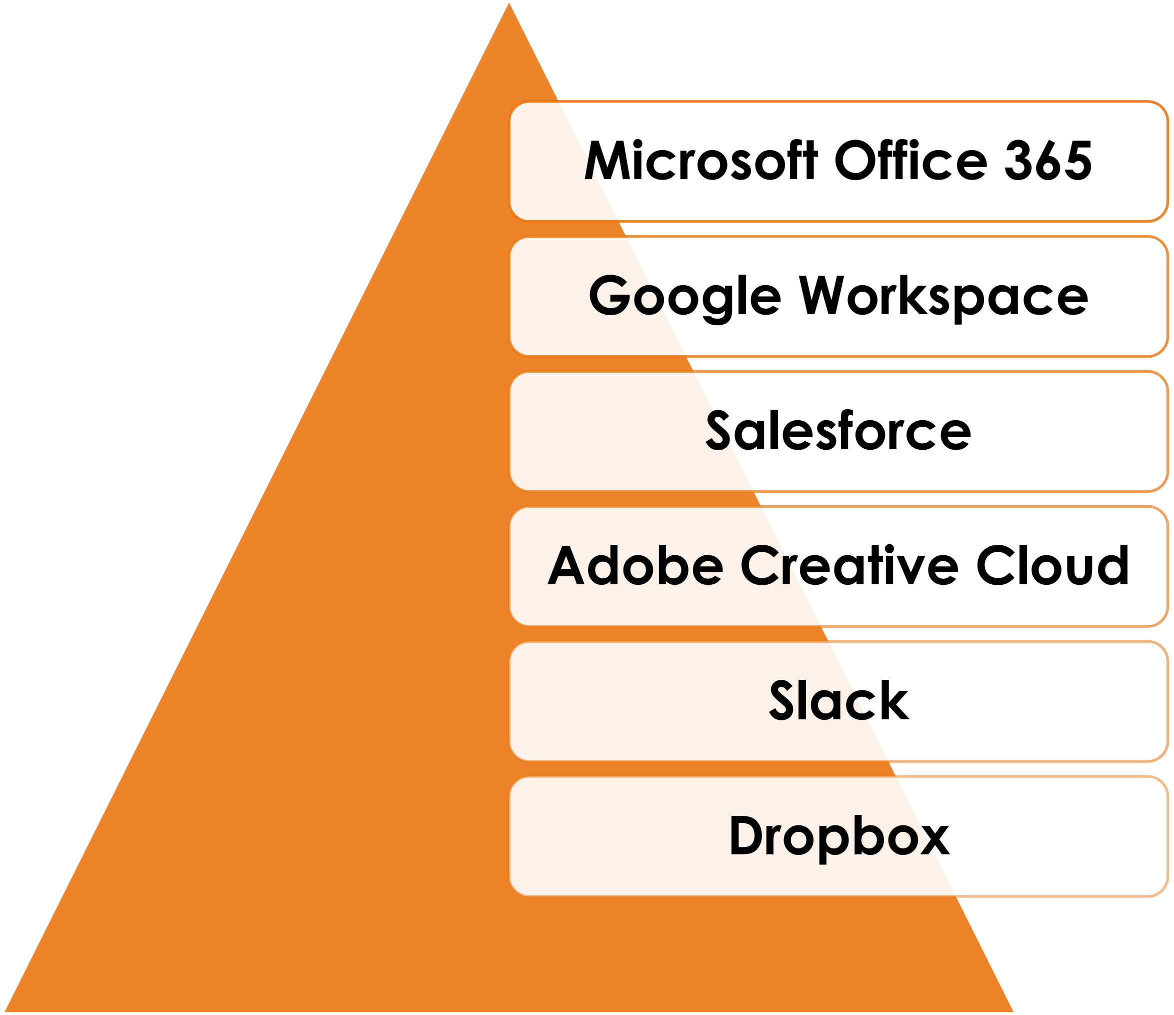
Managed Infrastructure:

- The service provider manages the infrastructure, ensuring the software is always up-to-date and available.

Multi-Tenancy:

- A single instance of the software serves multiple customers, with data kept separate and secure.

SAAS PROVIDERS



Microsoft Office 365

Google Workspace

Salesforce

Adobe Creative Cloud

Slack

Dropbox

CAAS

- **Containers as a Service (CaaS)** is a cloud computing service model that allows users to manage and deploy containerized applications and clusters.
- CaaS provides a framework for running and managing containers, including Docker or Kubernetes, on cloud infrastructure.

HOW CAAS WORKS

Containerization:

- Applications are packaged in containers, which encapsulate the code, dependencies, and runtime environment.

Orchestration:

- CaaS platforms often use orchestration tools like Kubernetes to automate the deployment, scaling, and management of containerized applications.

Resource Management:

- Users can deploy and manage containers on a cluster of virtual or physical machines.

Self-Service Interface:

- Typically, CaaS platforms provide a web-based dashboard or API for managing containers and resources.

CAAS PROVIDERS

- Google Kubernetes Engine (GKE)
- Amazon Elastic Kubernetes Service (EKS)
- Microsoft Azure Kubernetes Service (AKS)
- IBM Cloud Kubernetes Service
- Red Hat OpenShift
- Docker Cloud

FAAS

- **Functions as a Service (FaaS)** is a cloud computing model that allows developers to build, run, and manage application functionalities without the complexity of maintaining the infrastructure typically associated with developing and launching an app.
- This model abstracts away the server management, allowing developers to focus solely on writing code.

HOW FAAS WORKS

Event-Driven:

- FaaS is typically event-driven, meaning functions are triggered by events such as HTTP requests, database changes, or message queue events.

Stateless Functions:

- Each function execution is independent and does not rely on previous executions, making them stateless.

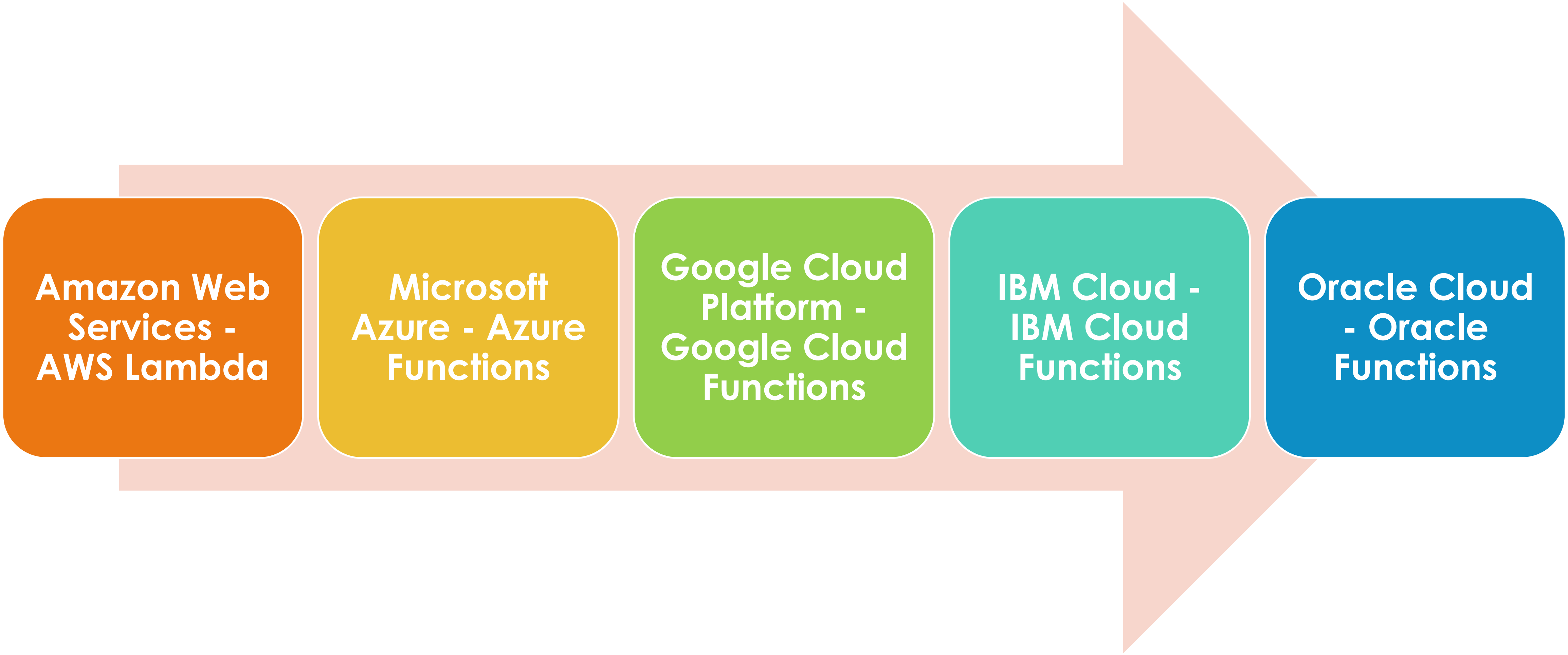
Automatic Scaling:

- Functions automatically scale up and down based on the number of incoming requests, with no need for manual intervention.

Pay-As-You-Go:

- Users are billed only for the compute time consumed by the function execution, not for idle time.

FAAS PROVIDERS



Amazon Web Services - AWS Lambda

Microsoft Azure - Azure Functions

Google Cloud Platform - Google Cloud Functions

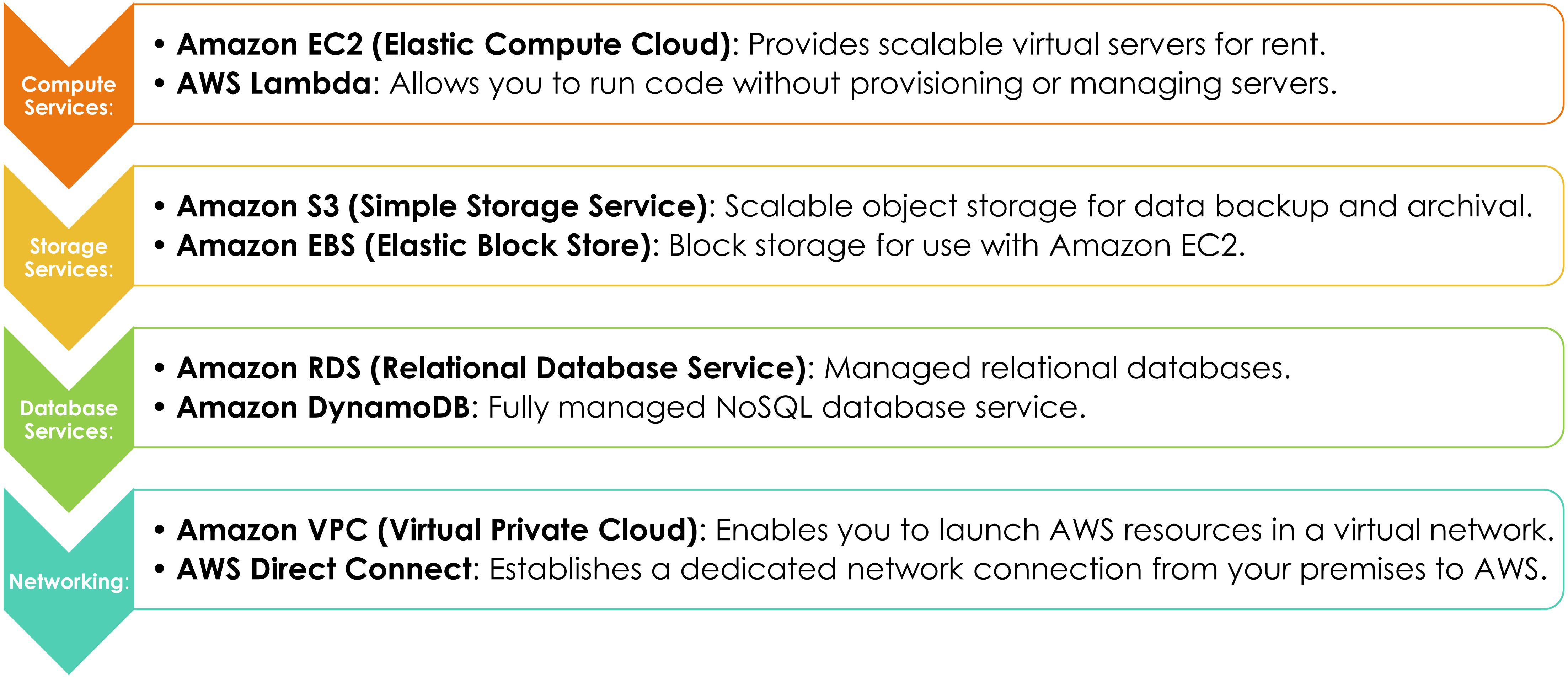
IBM Cloud - IBM Cloud Functions

Oracle Cloud - Oracle Functions

AWS (AMAZON WEB SERVICES)

- AWS is a comprehensive and widely adopted cloud platform, offering over 200 fully featured services from data centers globally.
- It provides a variety of services including computing power, storage, and databases, along with other functionalities like machine learning, artificial intelligence, and Internet of Things (IoT), among others.

KEY FEATURES OF AWS



Compute Services:

- **Amazon EC2 (Elastic Compute Cloud)**: Provides scalable virtual servers for rent.
- **AWS Lambda**: Allows you to run code without provisioning or managing servers.

Storage Services:

- **Amazon S3 (Simple Storage Service)**: Scalable object storage for data backup and archival.
- **Amazon EBS (Elastic Block Store)**: Block storage for use with Amazon EC2.

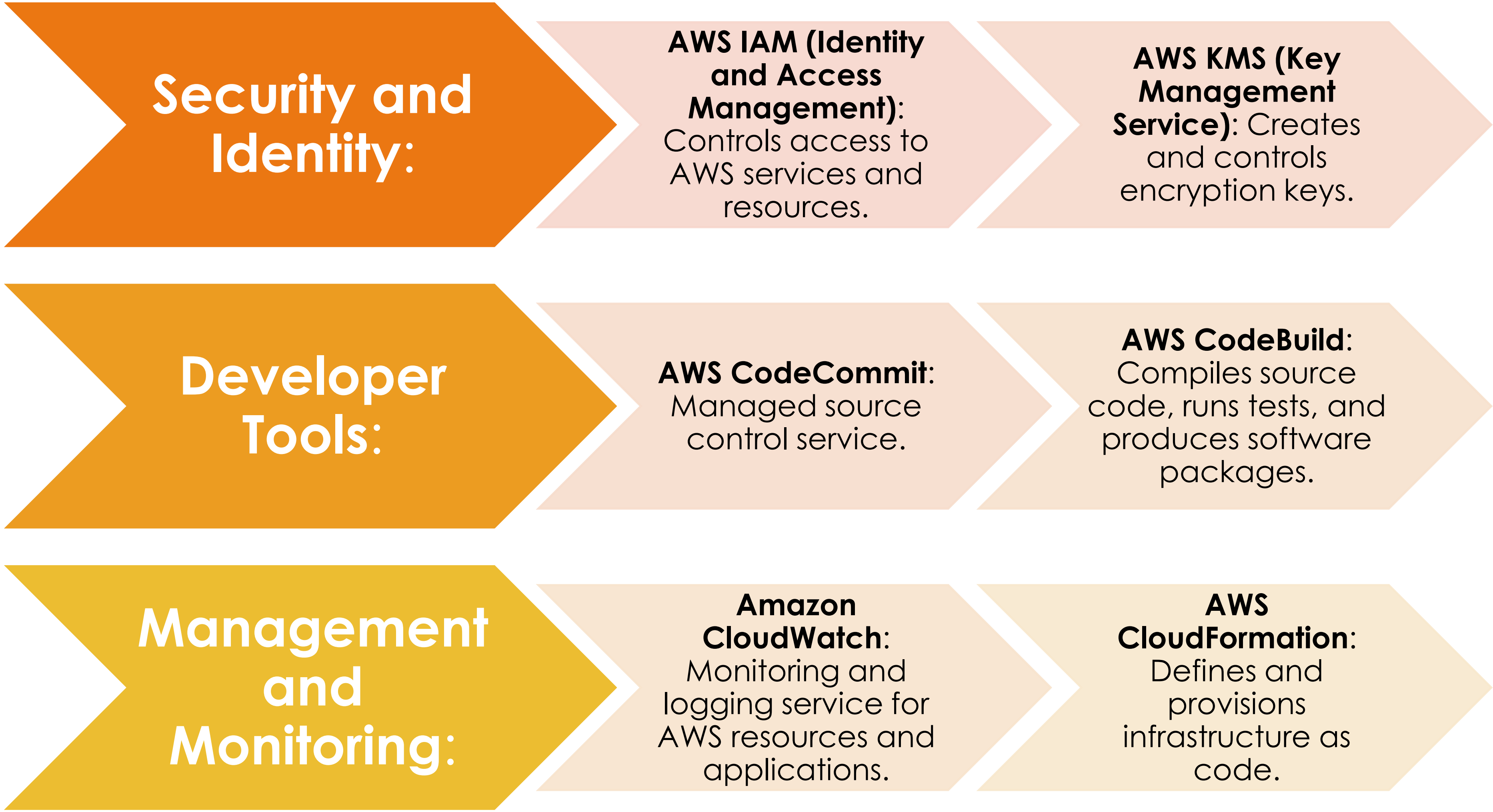
Database Services:

- **Amazon RDS (Relational Database Service)**: Managed relational databases.
- **Amazon DynamoDB**: Fully managed NoSQL database service.

Networking:

- **Amazon VPC (Virtual Private Cloud)**: Enables you to launch AWS resources in a virtual network.
- **AWS Direct Connect**: Establishes a dedicated network connection from your premises to AWS.

KEY FEATURES OF AWS



Security and Identity:

AWS IAM (Identity and Access Management): Controls access to AWS services and resources.

AWS KMS (Key Management Service): Creates and controls encryption keys.

Developer Tools:

AWS CodeCommit: Managed source control service.

AWS CodeBuild: Compiles source code, runs tests, and produces software packages.

Management and Monitoring:

Amazon CloudWatch: Monitoring and logging service for AWS resources and applications.

AWS CloudFormation: Defines and provisions infrastructure as code.

OVERVIEW OF EC2

- Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud
- Features:
    - Virtual computing environments, known as **instances**
    - Preconfigured templates for your instances, known as **Amazon Machine Images (AMIs)**
    - Various configurations of CPU, memory, storage, and networking capacity for your instances, known as **instance types**

FEATURES OF EC2

- Secure login information for your instances using **key pairs**
- Storage volume for temporary data that's deleted when you stop or terminate your instance, known as instance **store volumes**
- Persistent storage volumes for your data using Amazon Elastic Block Store (Amazon EBS), known as **Amazon EBS volumes**
- Multiple physical locations for your resources, such as instances and Amazon EBS volumes, known as **Regions and Availability Zones**

SOME MORE FEATURES

- A firewall that enables you to specify the protocols, ports, and source IP ranges that can reach your instances using **security groups**
- Static IPv4 addresses for dynamic cloud computing, known as **Elastic IP addresses**
- Metadata, known as tags, that you can create and assign to your **Amazon EC2 resources**
- Virtual networks you can create that are logically isolated from the rest of the AWS cloud, and that you can optionally connect to your own network, known as **virtual private clouds (VPCs)**

BENEFITS OF EC2

- Elastic
- Flexible
- Reliable
- Inexpensive
- Controlled
- Secure

INSTANCE TYPES

- When you launch an instance, the instance type determines the hardware of the host computer used for your instance. Instances can be of the following types:
  - Burstable performance instances (T3 and T2)
  - General purpose instances (M5, M5a, M5d, T2, and T3)
  - Compute optimized instances (C4, C5, Cn, and Cd)
  - Memory optimized instances (R4, R5, R5a, and R5d)
  - Storage optimized instances (D2, H1, and I3)
  - Windows accelerated computing instances
  - T1 Micro Instances

INSTANCE PURCHASING OPTIONS

- On-Demand Instances
- Reserved Instances
- Scheduled Instances
- Dedicated Hosts
- Dedicated Instances
- Capacity Reservations

PRICING

- Amazon EC2 is free to use
- There are three ways to pay for Amazon EC2 instances:
  - On-Demand
  - Reserved Instances
  - Spot Instances
- You can also pay for Dedicated Hosts which provide you with EC2 instance capacity on physical servers dedicated for your use

ACTIVITY 1

- You are given a project to launch and connect to EC2 Linux instance.
- Connect with EC2 Instance using Amazon CLI

ACTIVITY 2

1. create one AWS EC2 Instance with AWS Linux and set HTTP and AllTCP in Network
2. Connect with AWS CLI using ssh link
   - ssh -i "springboot.pem" ec2-user@ec2-100-26-29-207.compute-1.amazonaws.com
3. install HTTPd Server
   - sudo yum install httpd -y
4. move to the folder /var/www/html: cd /var/www/html
5. Once you are here let's create one HTML file using VI Editor
   - sudo vi index.html (add sample code)
6. Start Http Service
   - sudo service httpd start
7. open your publc Ip and see with http protocol

SECURITY GROUPS

- A security group acts as a virtual firewall that controls the traffic for one or more instances.
- When you launch an instance, you can specify one or more security groups; otherwise, we use the default security group.
- You can add rules to each security group that allows traffic to or from its associated instances.
- You can modify the rules for a security group at any time; new rules are automatically applied to all instances associated with the security group.
- When you decide whether to allow traffic to reach an instance, you evaluate all the rules applied to security groups associated with the instance.

SECURITY GROUP RULES

- By default, security groups allow all outbound traffic.
- Security group rules are always permissive; you can't create rules that deny access.
- Security groups are stateful. If you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules.
- When you associate multiple security groups with an instance, the rules from each security group are effectively aggregated to create one set of rules.

DEFAULT VS CUSTOM SECURITY GROUPS

Default Security Group:

- Your AWS account has a default security group for the default Virtual Private Cloud (VPC) in each region.
- It allows inbound traffic from all instances which are associated with it.
- It allows outbound traffic from all instances.

Custom Security Group:

- You can create your own security group and specify it when you launch your instances. You must provide a name and a description.
- It allows no inbound traffic.
- It allows all outbound traffic.

AMAZON VPC

- Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined.
- This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.
- You can specify an IP address range for the VPC, add subnets, associate security groups, and configure route tables.

EBS (AMAZON ELASTIC BLOCK STORE)

- Amazon EBS provides block level storage volumes to use with EC2 instances.
- EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances.
- EBS volumes are highly available and reliable storage volumes that can be attached to any running instance that is in the same Availability Zone.

FEATURES OF EBS

- EBS volumes are created in a specific Availability Zone and can be attached to any instance in that same zone. To make a volume available outside the zone, you can create a snapshot and restore it to a new volume anywhere in that region.
- Amazon EBS provides the following volume types: General Purpose SSD (gp2), Provisioned IOPS SSD (io1), Throughput Optimized HDD (st1), and Cold HDD (sc1).
- You can create your EBS volumes as encrypted volumes, in order to meet a wide range of data-at-rest encryption requirements for regulated or audited data and applications.

FEATURES OF EBS

- You can create point-in-time snapshots of EBS volumes. Snapshots protect data for long-term durability, and they can be used as the starting point for new EBS volumes.

- Performance metrics, such as bandwidth, throughput, latency, and average queue length are available through the AWS Management Console.

- The metrics, provided by Amazon CloudWatch, allow you to monitor the performance of your volumes to make sure that you are providing enough performance for your applications without paying for resources you don't need.

ACTIVITY

1. You are given a project to create an EBS volume.
2. Attach that volume to your running Instance
3. You are given a project to format and mount an EBS volume on Linux.
4. Detach volume from an instance.
5. Delete a created volume.

MOUNT UNMOUNT VOLUME

1. connect your instance using aws cli
2. check attached volumes.
   - Lsblk
3. let's check if your volume has any data:
   - sudo file -s /dev/xvdf
4. Let's format the volume to ext4 filesystem.
   - sudo mkfs -t ext4 /dev/xvdf
5. let's create some folder and add it to this external volume.
   - sudo mkdir /testdata
6. mount that directory to ext volumn:
   - sudo mount /dev/xvdf /testdata/
7. to check its mounted or not: lsblk
8. for unmount
   - sudo umount /dev/xvdf

EBS SNAPSHOTS

- EBS snapshots are incremental backups; every snapshot only copies the blocks in the volume that change after the last snapshot.
- The only changed blocks are copied (in compressed form) to the S3 in subsequent snapshots.

EBS Volume 1

A		B	C
	D		
		E	

A,B,C,D,E

EBS Snapshot1

A		B	C
	D		
		E	

EBS Volume 2

A		B	C
	D^1		
		E^1	F

A,B,C, D^1,E^1,F

EBS Snapshot2

A		B	C
	D^1		
		E^1	F

EBS Volume 3

A		B	C
	D^1		
		E^2	F^2
			G

A,B,C, D^1,E^2, F^2,G

EBS Snapshot3

A		B	C
	D^1		
		E^2	F^2
			G

CREATING EBS SNAPSHOTS

- A snapshot of an EBS volume can be created and used as a baseline for new volumes or for data backup.
- Snapshots are incremental when periodic snapshots of a volume are made.
- Snapshots occur asynchronously.
- An in-progress snapshot is not affected by ongoing reads and writes to the EBS volume.

CREATING EBS SNAPSHOTS

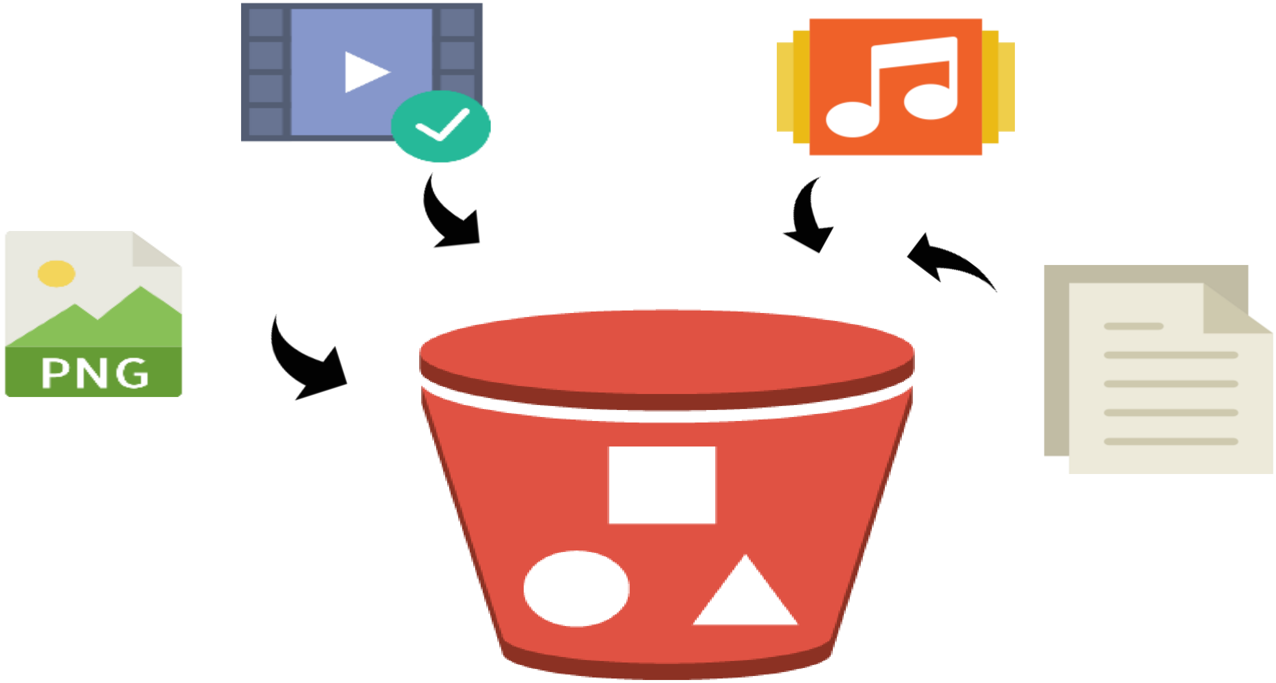
- Snapshots only capture data that has been written to the Amazon EBS volume when the snapshot command is issued.
- Your snapshot will be complete if you can pause any file writes to the volume long enough to take a snapshot. However, if you can't pause all file writes to the volume, you should unmount the volume from within the instance, issue the snapshot command, and then remount the volume.

ACTIVITY

- You are given a project to create a snapshot.
- View the created snapshot
- You are given a project to initialize a volume restored from a snapshot.

S3 BUCKET

- Amazon S3 provides object storage and is built for storing and recovering data from anywhere over the Internet.



Unlimited Storage

PURPOSE OF S3

- Factors that make a repository expensive and time-consuming are:
- Need to purchase hardware and software components
- Need to hire a team of experts for maintenance
- Lack of scalability based on your requirements
- Requirement for data security

CREATE S3 BUCKET

- You need to create an S3 bucket in one of the AWS regions in order to upload your data.

- When data is added to the bucket, Amazon S3 creates a unique version ID and allocates it to the object.

- The name of an S3 bucket cannot be used by another AWS account in any AWS region until the bucket is deleted.

- You should choose a location that is close to you in order to optimize latency, minimize costs, and address regulatory requirements.

RULES FOR BUCKET NAMING

- Bucket names must be between 3 and 63 characters long.
- Bucket names must be a series of one or more labels.
- AWS recommends separating labels with a single period (.).
- Bucket names can contain lowercase letters, numbers, and hyphens.
- Each label must start and end with a lowercase letter or a number.

S3 OBJECTS

- Amazon S3 contains multiple objects with keys and values. An object consists of the following:
  - Key: It is the name assigned to an object
  - Version ID: It is a string generated by Amazon S3 when an object is added to a bucket
  - Value: It is the content that you store in an object
  - Metadata: It is a set of name-value pairs with which you can store information regarding the object
  - Subresource: It is a mechanism used by Amazon S3 to store object-specific additional information
  - Access Control Information: Amazon S3 supports both resource-based access control and user-based access control

S3 OBJECT KEY

- You specify a key name when you create an object.
- It uniquely identifies an object in a bucket.
- The name for a key is a sequence of Unicode characters . Its UTF-8 encoding is maximum 1024 bytes.
- You cannot download an object using the Amazon S3 console if an object key name consists of a single period (.) or two periods (..).

S3 OBJECT METADATA

- There are two kinds of object metadata: system metadata and user-defined metadata.
- Amazon S3 maintains a set of system data for every object in a bucket.
- There are two kinds of system metadata. They can be metadata such as object creation date and other system metadata such as the storage class configured for the object.
- Metadata can be assigned to an object while uploading it. This information can be added as a name-value pair.

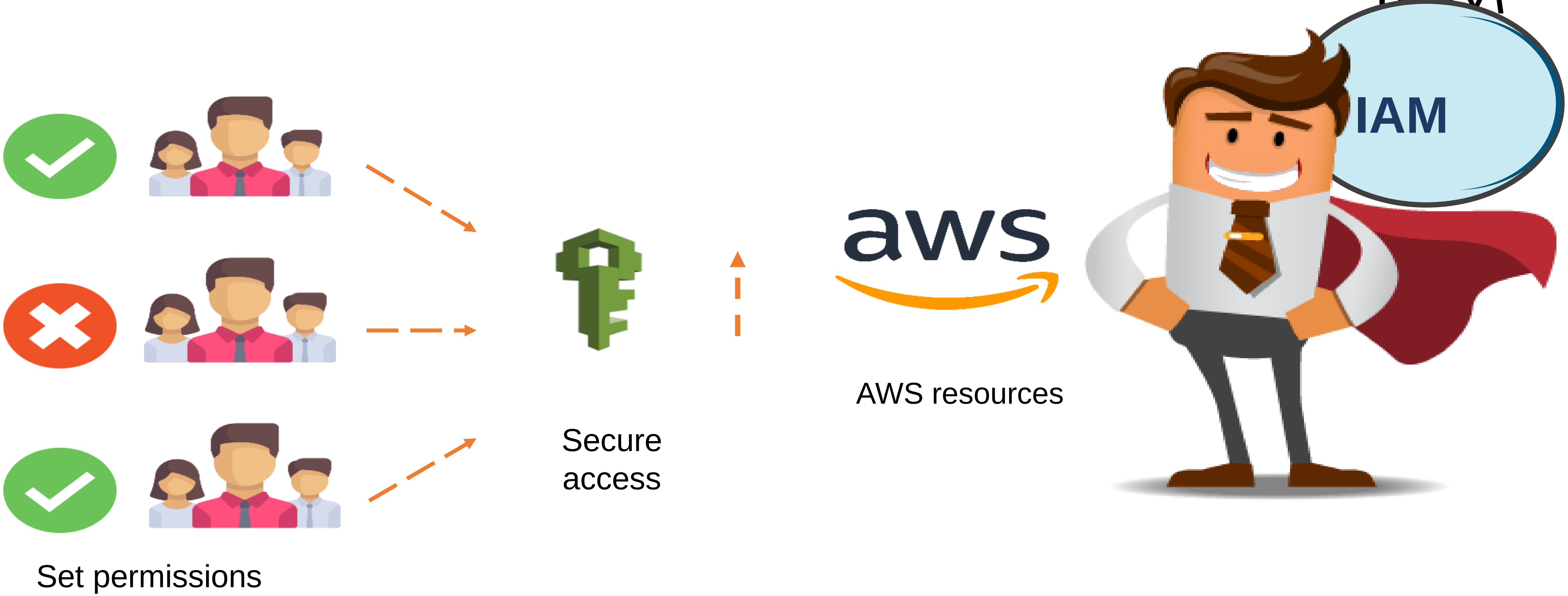
ACTIVITY

- You are given a project to create an S3 bucket.
- Add Some data to your buckets
- See the object created in the buckets
- Empty bucket
- Delete bucket

IDENTITY AND ACCESS MANAGEMENT (IAM)

- AWS Identity and Access Management is a web service for securely controlling access to AWS services.
- It enables you to create and control services for user authentication or limit access to a certain set of users on your AWS resources.

IAM

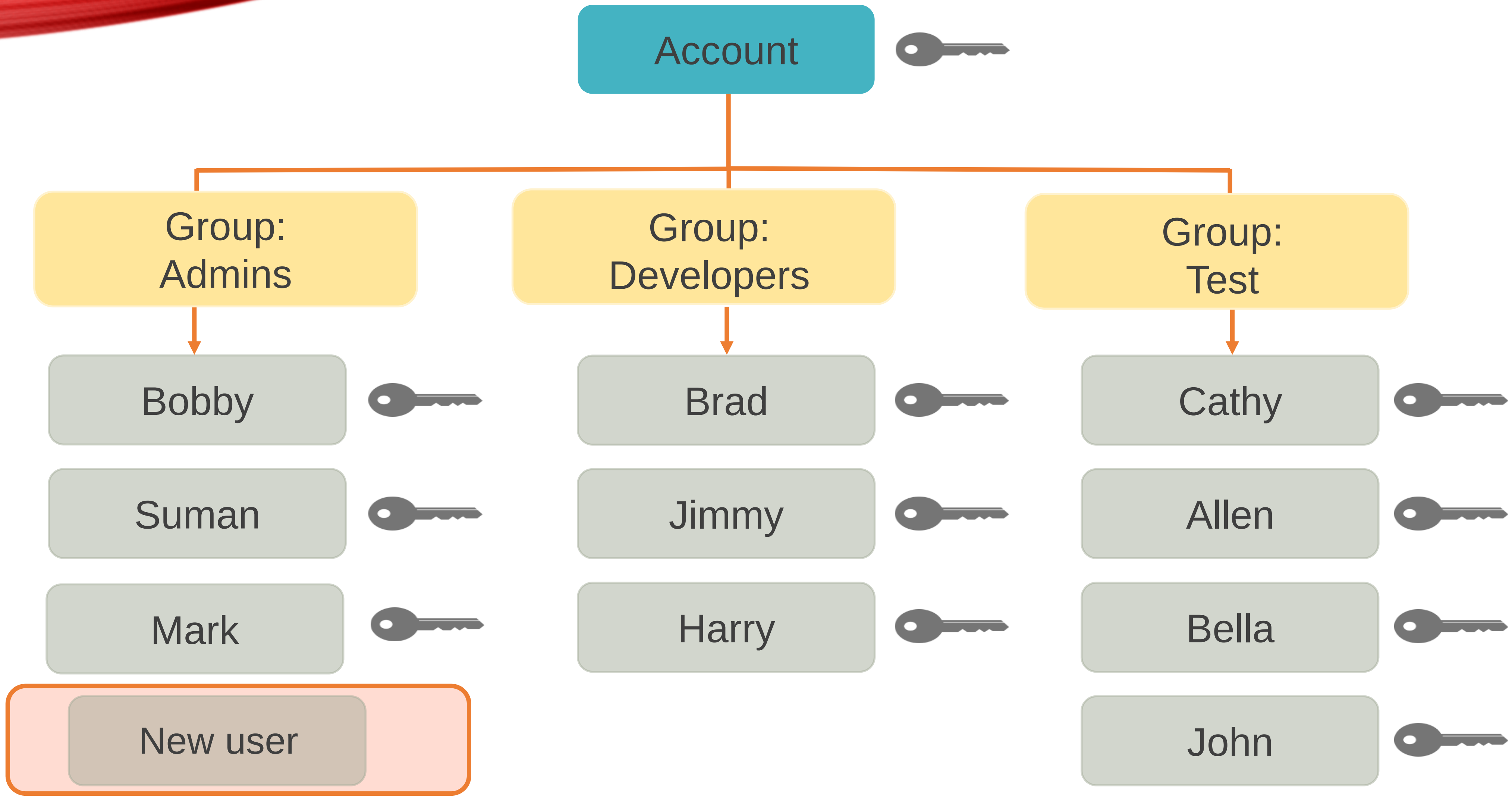


Set permissions

Secure access

AWS resources

IAM GROUPS



Account

Group: Admins

Group: Developers

Group: Test

Bobby

Suman

Mark

New user

Brad

Jimmy

Harry

Cathy

Allen

Bella

John

Three groups

Set permissions

LET'S PUBLISH ONE WEBSITE

- Create Ubuntu EC2 Instance.
- Connect with Instance.
- Install Apache:
  - sudo apt update -y
  - sudo apt install apache2 -y
  - sudo systemctl start apache2
  - sudo systemctl enable apache2
- Once its started You can check your public IP with default apcahe Server Page.

LET'S PUBLISH ONE WEBSITE

- For Cloning repo install git:
    - sudo apt install git –y
- Move to the default Http folder:
    - cd /var/www/html
    - sudo git clone https://github.com/your-username/your-repo.git
- Move the contents of the repo to /var/www/html:
    - sudo mv your-repo/* .
    - sudo rm -rf your-repo
- Set Permissions:
    - sudo chmod -R 755 /var/www/html
    - sudo chown -R www-data:www-data /var/www/html
- Restart Server:
    - sudo systemctl restart apache2
- Access Website:
    - http://your-ec2-public-ip