ETHIOPIAN IT PARK
የኢትዮጵያ ኢንፎርሜሽን ቴክኖሎጂ ፓርክ

CITCOT

ZERGAW CLOUD

IT Infrastructure Administration

Day 15: Cloud

Ephrem Teshale(PhD)
Tadios Abebe

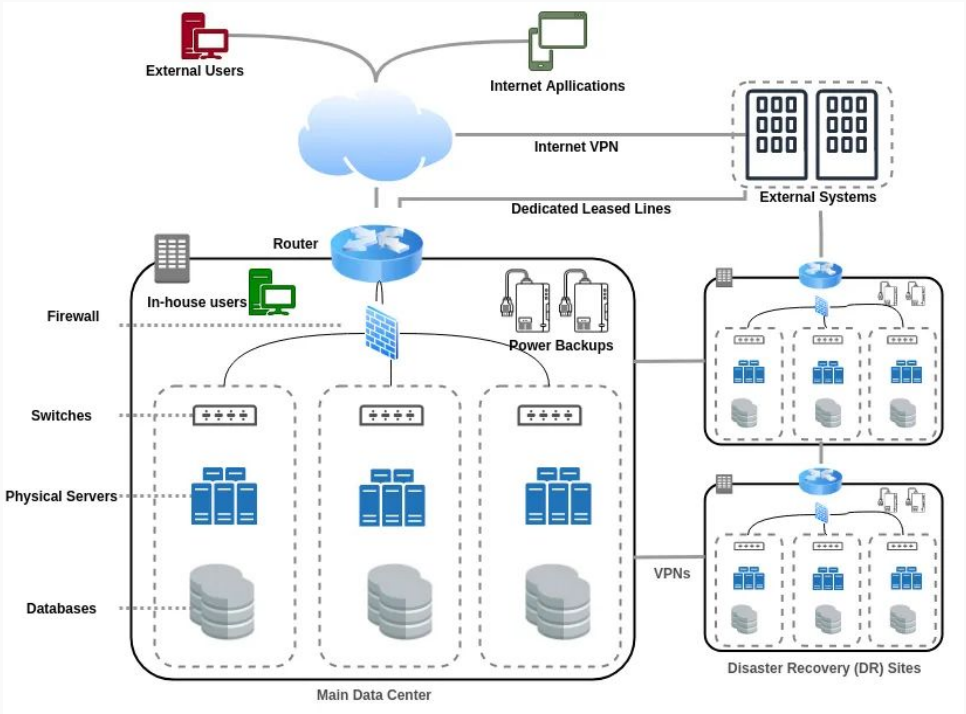
Aug 7, 2025

Cloud

Day 15
Training Outline

On-prem/Colo Architecture
Pros and Cons of On-prem/Colo
The Cloud
Why Cloud
When to go for Cloud
Cloud Service Models
Cloud Deployment Models
Cloud Architectures
Shared Responsibility Model
Cloud Resource Hierarchy
Advanced Cloud Services
Advantages of the Cloud
Disadvantages of the Cloud
Components of Cloud
Infrastructure as a Code
Configuration Management

On-prem/Colo Architecture



- Compute hw/sw
- Network hw/sw
- Storage hw/sw
- Power
- Environment conditioning
- Connectivity
  - branches
  - telecom operator
  - Internet
- Security

Pros and Cons of On-prem/Colo

- Pros:
  - Having a **direct control** of your system and ensuring data security
  - You can **customize** your infrastructure according to your need
  - To guarantee **low latency** and high performance **requirement** for internal systems
  - Lack of outside connectivity due to poor availability of internet
- Cons:
  - High CapEx
  - High OpEx
  - Scalability and resilience come at a huge price
  - Lacks accessibility from outside of the premise

infrastructure is huge distraction from the core business

ETHIOPIAN IT PARK

CITCOT

The Cloud

What is cloud?

a term used to describe a distributed network of servers, each with a unique function. The cloud is not a physical entity, but instead is a vast network of remote servers geographically spread which are hooked together and meant to operate as a single ecosystem.

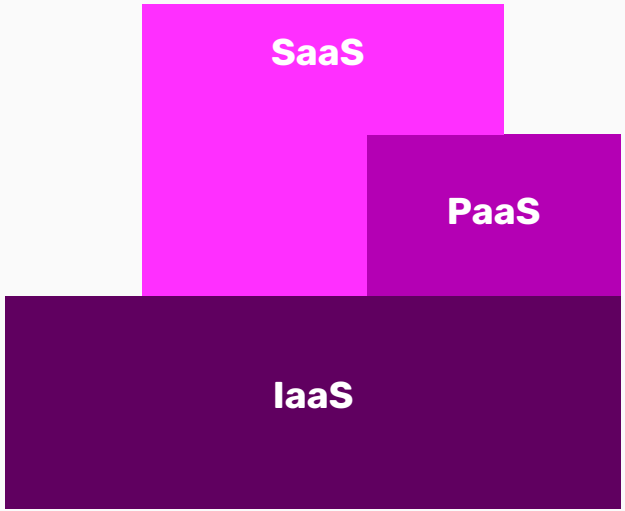
Why Cloud?

- businesses desire to **focus** on core business objective
- **scale** quickly to adapt changing workload requirements
- **cost-effective** alternative to traditional IT infrastructure
- **flexibility** and **agility** to innovate and deploy **new services rapidly**
- global reach
- reliability and resiliency
- security and compliance

When to go for Cloud?

- When you have a **variable** workload
- When you have limited IT resource (physical and human)
- When you need disaster recovery and business continuity
- When you need global reach and accessibility
- When you need cost efficiency

Cloud Service Models



- Derivatives
  - Call Center as a service
  - CaaS: Container as a service
  - DaaS: Desktop as a Service
  - Security as a Service
  - FaaS: Function as a Service
  - STaaS: Storage as a Service
  - AIaaS: Artificial Intelligence as a Service
  - …

Infrastructure as a Service (IaaS)

- on-demand access to cloud-hosted **physical and virtual servers, storage and networking**
- customers can provision, configure and use in much the same way as they use on-premises hardware
- cloud service provider hosts, manages and maintains the hardware and computing resources in its own data centers.
- it's like **renting an empty apartment** - you get the space and utilities, but **you furnish and maintain** it yourself
- offers the **most control and flexibility** but also **requires the most technical expertise** from the customer.

International

Local

Platform as a Service (PaaS)

- builds upon IaaS, providing a platform for **developing**, **deploying**, and **managing** applications
- it is like a pre-built workbench with tools and utilities
- offers features like **databases**, **development frameworks**, and **content management systems**
- a balance between **control** and **ease of use**
- ideal **for developers** → focus on building applications without worrying about server management

Software as a Service (SaaS)

- ready-to-use software applications delivered over the internet.
- using a web application like Gmail or Dropbox
- cloud provider **manages everything** - the infrastructure, platform, and the software itself
- customer **simply accesses** the application through a web browser or mobile app
- it's like subscribing to a meal delivery service - you get prepared meals delivered to your door without any cooking required
- offers the **least control** but is the **easiest to use**, ideal for everyday applications and software that doesn't require heavy customization

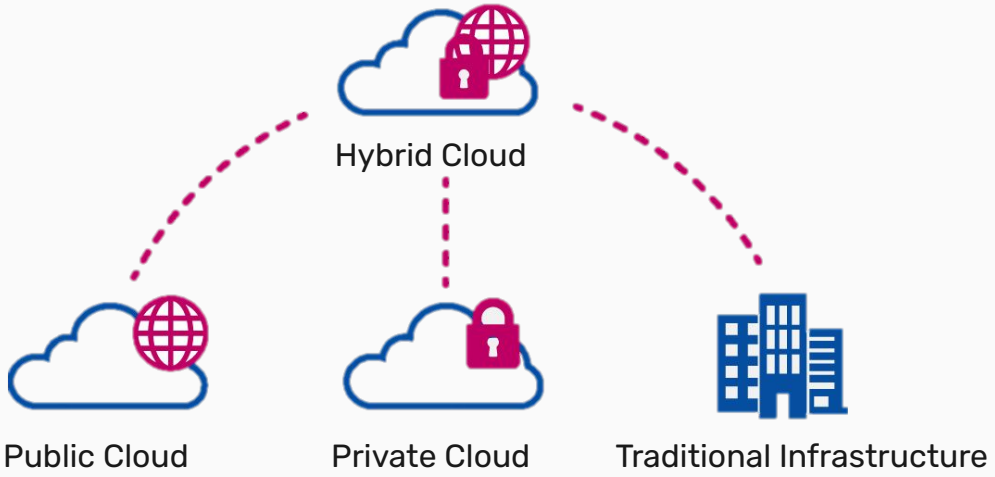
Cloud Deployment Models

Private Cloud

Public Cloud

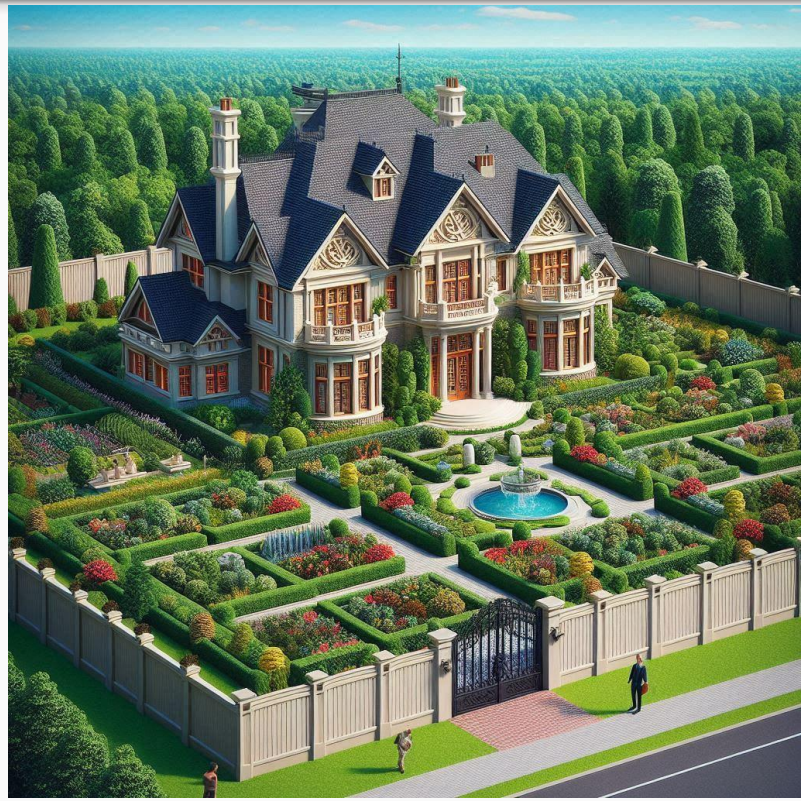
Hybrid Cloud

Multi Cloud



Private Cloud

- dedicated for the use of a single organization
- all computing resources (servers, storage, networking) are owned and operated by the organization itself
- on-premises in their data center or in a dedicated off-site facility
- offers a **high degree of security, control, and customization**
- **expensive** to set up and maintain
- requiring significant investment in hardware, software, and IT expertise
- combines many benefits of cloud computing—including elasticity, scalability and ease of service delivery—with the access control, security and resource customization of on-premises infrastructure



Public Cloud

- owned and operated by a third-party cloud service provider (CSP)
- think of it like a public park
- resources are shared among multiple organizations (tenants) who access them over the internet
- offer **scalability**, **cost-effectiveness**, and a **wide range** of services on-demand
- only pay for what you use
- concerns may arise as data resides on a shared infrastructure, and control over the environment is limited

Google Cloud aws DigitalOcean telecloud Azure ZERGAW CLOUD

ETHIOPIAN IT PARK

CITCOT

ZERGAW CLOUD

Hybrid Cloud

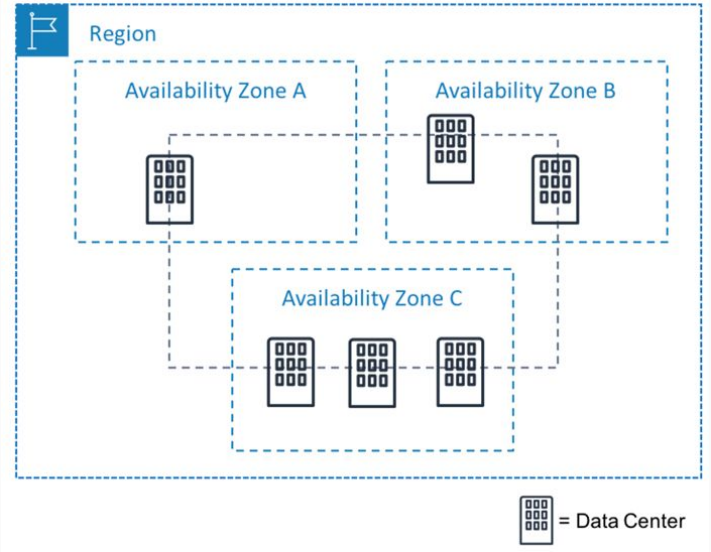
- combines aspects of both private and public clouds
- a connected community with private residences and public parks
- critical data and applications in a private cloud for enhanced security and **low latency**, while using a public cloud for non-critical workloads that require scalability or flexibility
- offer a balance between security, control, and scalability, allowing organizations to leverage the strengths of both private and public cloud environments.

Multi Cloud

- involves using multiple cloud services from different providers concurrently
- can be as simple as email SaaS from one vendor and image editing SaaS from another
- different from a hybrid cloud, which typically combines a private cloud with a single public cloud
- Think of it like visiting multiple parks across different cities
- allows organizations to leverage the best features and pricing options from various cloud providers for different needs
- can enhance flexibility, avoid vendor lock-in, and potentially optimize costs
- managing and integrating multiple cloud environments can become complex

Cloud Architectures

Generally cloud has the following building blocks



- Region
- Zone
- Failure Domain

- The above building blocks help ensure high availability, redundancy, and disaster recovery.
- Different public cloud providers have similar yet distinct ways of organizing their infrastructure.

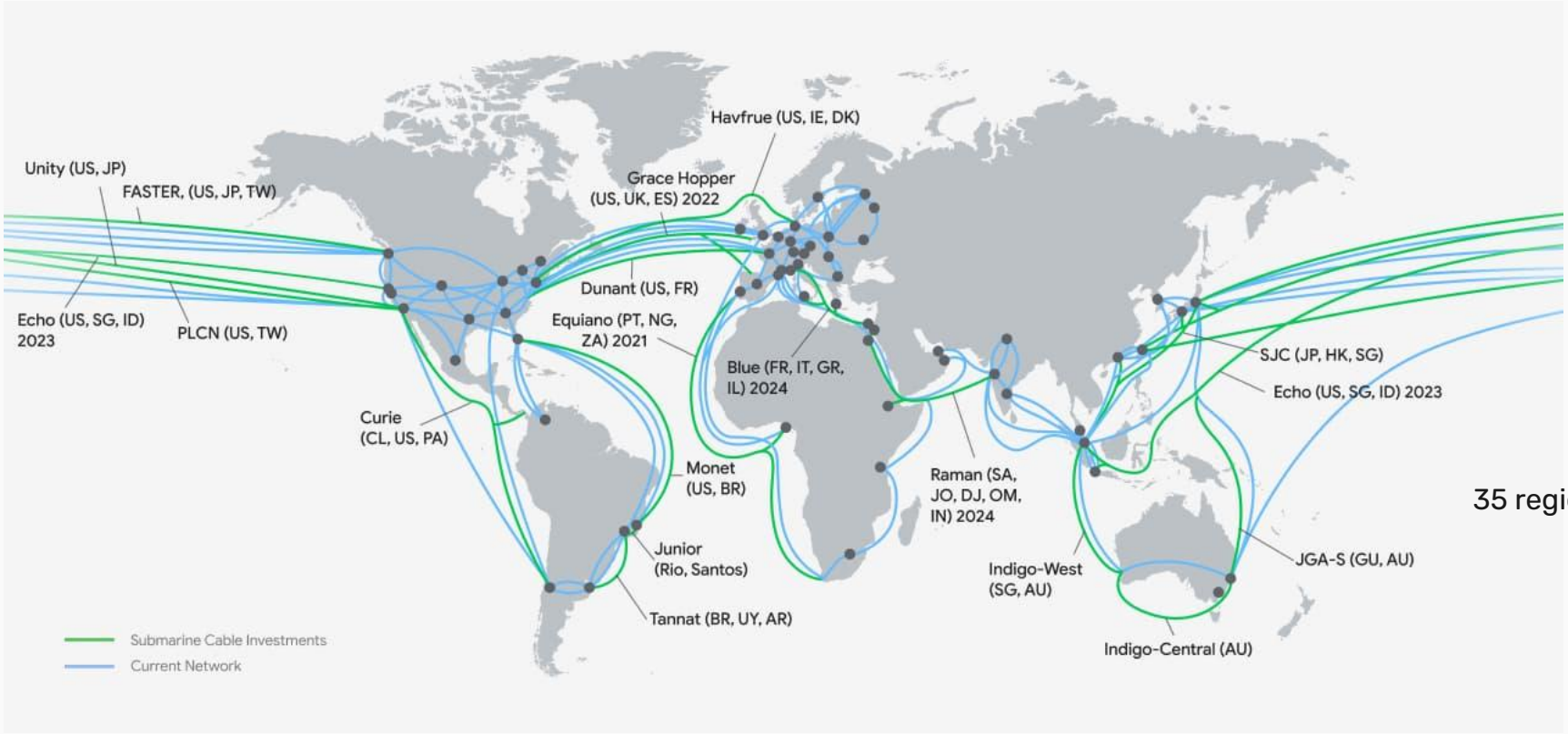
A Cloud Region

- specific geographical location where a cloud service provider (CSP) maintains data centers and infrastructure
- these data centers house the physical servers, storage devices, and networking equipment that power the cloud services offered by the CSP
- each region generally operates independently of others, providing geographical separation to ensure fault tolerance and compliance with data residency requirements
- they help in deploying applications closer to users for reduced latency
- they help companies in complying with data sovereignty laws
- they provide geographical redundancy

AWS Regions

26 regions

GCP Regions



35 regions

Azure Regions

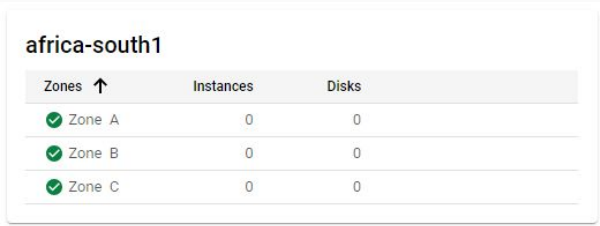
* Three Azure Government region locations undisclosed

64 regions

A Cloud Availability Zone

- a specific location **within a cloud region** offered by a CSP
- each zone has its own independent power, cooling, and network infrastructure
- physically separate from each other but connected by high-speed, low-latency networks
- this isolation helps ensure that if one zone experiences an outage due to power failure, natural disaster, or hardware malfunction, the other zones remain operational
- this redundancy enhances service availability and fault tolerance
- May have different nomenclatures between CSPs:
  - GCP, AWS: Availability Zones
  - Oracle: Availability Domain
  - Azure: Data centers ...
- help in balancing workloads and distributing resources to avoid single points of failure.

GCP Availability zones in Africa



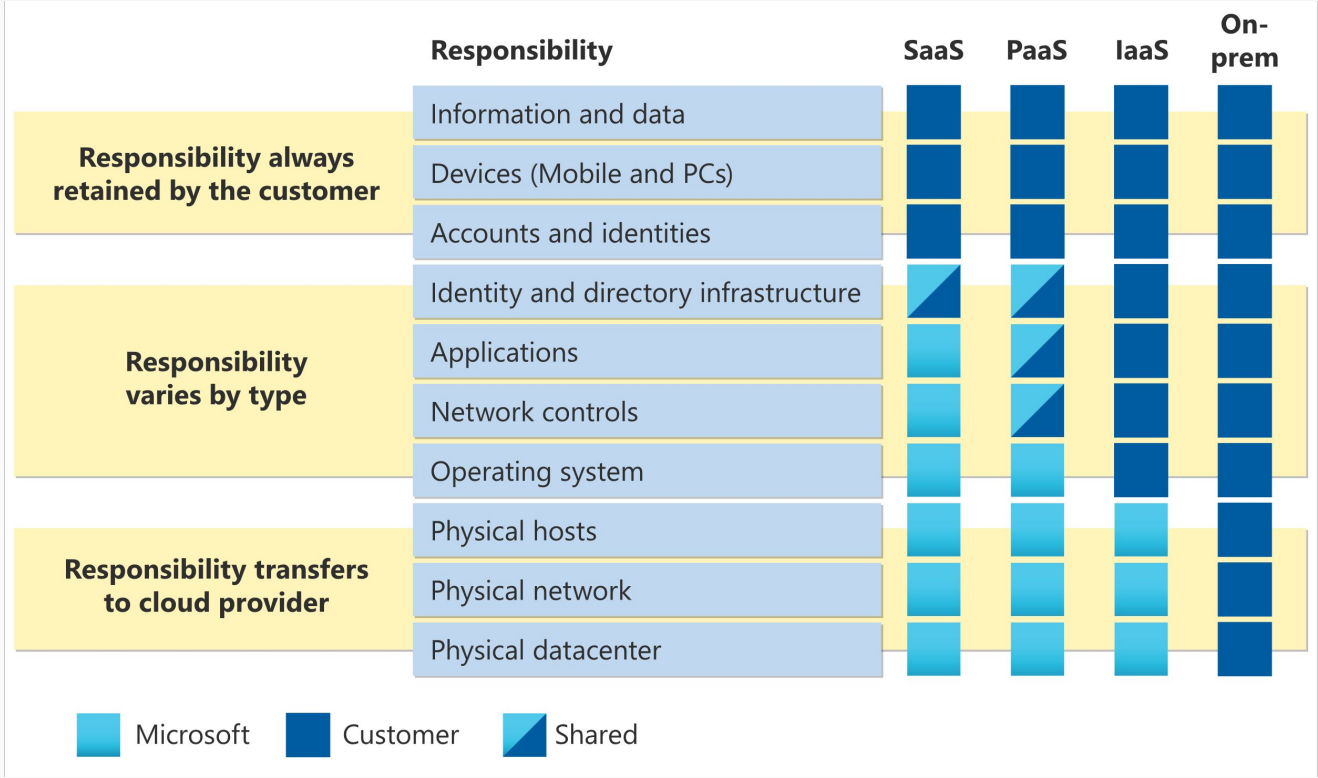
A Cloud Failure Domain

- a group of resources within a cloud environment that are susceptible to a single point of failure
- if that point of failure occurs, all the resources within the domain become unavailable
- a domain of **shared vulnerability** that encompasses resources that share common dependency or infrastructure element
- Examples:
    - A single server: In a non-cloud environment, a physical server itself can be a failure domain. If the server fails, all the applications and data hosted on it become unavailable
    - A cloud instance: In a cloud environment, a virtual machine instance can be a failure domain. If the instance fails due to hardware issues or software problems, the application running on it becomes unavailable
- An availability zone (AZ): While designed for redundancy, an entire availability zone can be considered a failure domain in some scenarios. A major power outage or natural disaster affecting the entire zone could disrupt all resources located within it
- can sometimes span multiple zones but usually refers to entities like racks, network segments, or even whole data centers
- They are used for designing applications to tolerate failures
- They ensure redundancy by replicating critical services across different failure domains
- useful in planning disaster recovery strategies

Shared responsibility model in the cloud

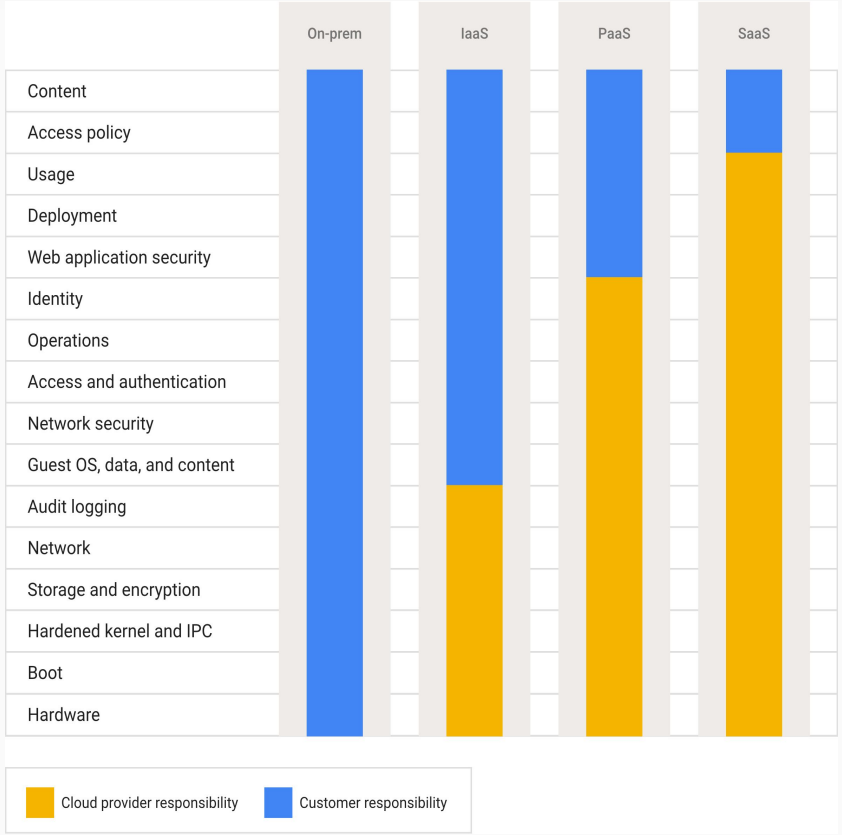
Outlines the division of security and compliance responsibilities between the cloud service provider (CSP) and the customer. This model ensures that both parties understand their obligations to secure the infrastructure and data within the cloud environment.

Shared responsibility Model in the Cloud: Azure



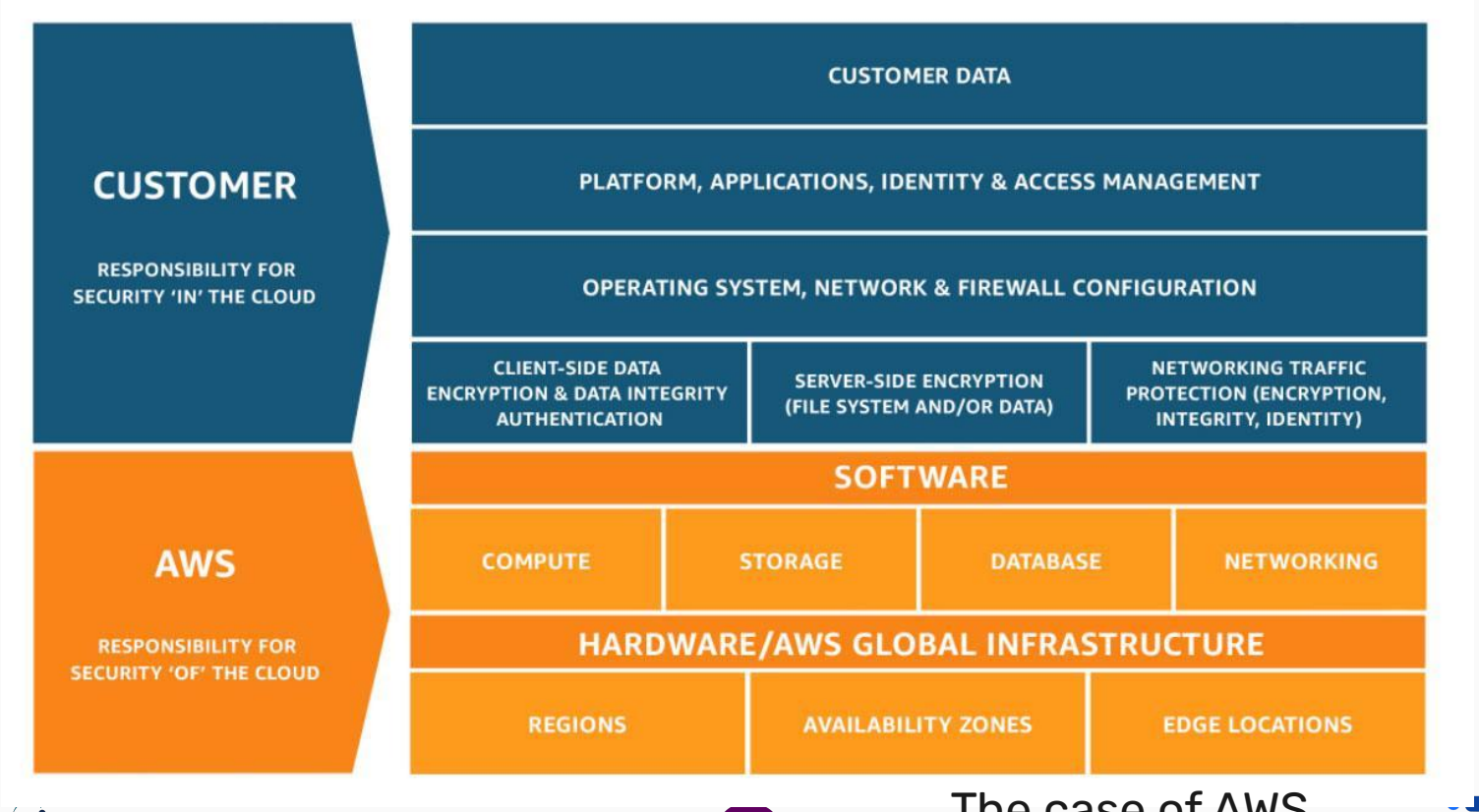
Azure

Shared responsibility Model in the Cloud: GCP



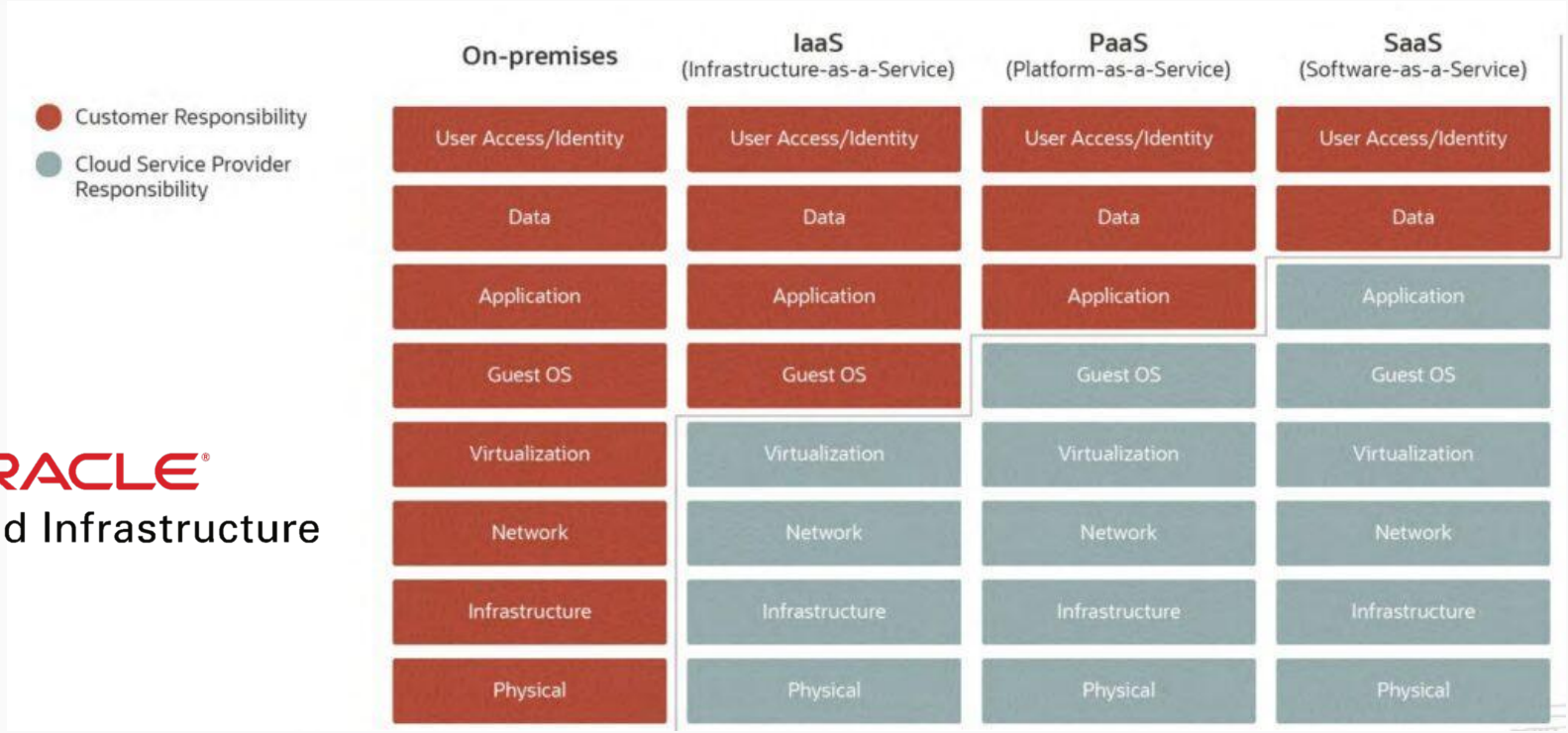
Google Cloud

Shared responsibility Model in the Cloud: AWS



The case of AWS

Shared responsibility Model in the Cloud: OCI



ORACLE®
Cloud Infrastructure

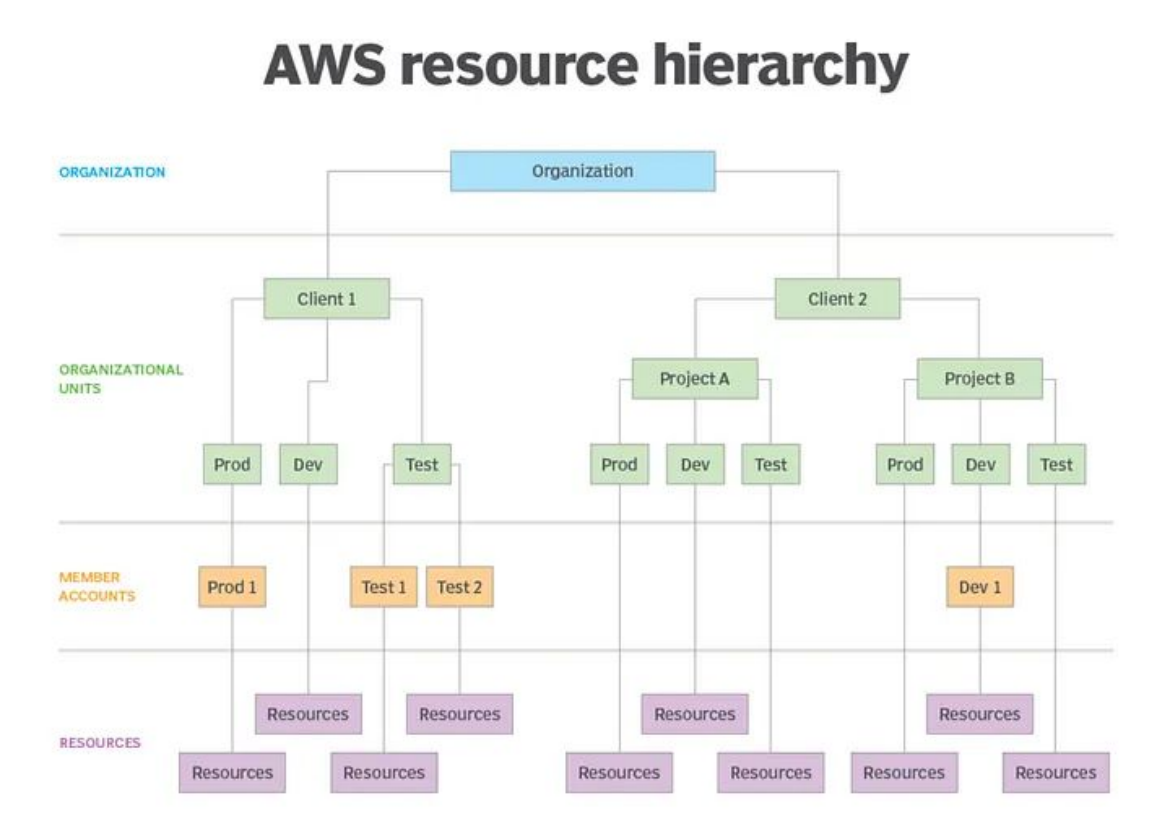
Shared responsibility Model in the Cloud: Zergaw Cloud

On-Premises	IaaS	PaaS	SaaS
User Access	User Access	User Access	User Access
Data	Data	Data	Data
Application	Application	Application	Application
Guest OS	Guest OS	Guest OS	Guest OS
Virtualization	Virtualization	Virtualization	Virtualization
Network	Network	Network	Network
Infrastructure	Infrastructure	Infrastructure	Infrastructure
Physical	Physical	Physical	Physical

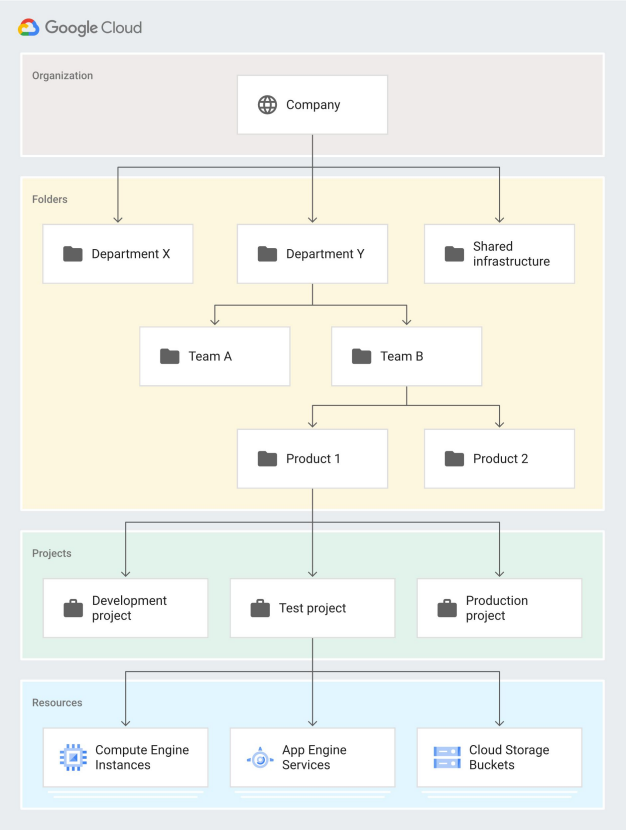
Cloud Resource Hierarchy

- a structured organization system for managing all the resources, permissions, and policies
- establishes a clear chain of command and access control, making it easier to manage, secure, and optimize cloud environment
- a tree-like structure, with a root node at the top and child nodes branching out below
- each level represents a specific type of resource or grouping
- the specific levels and terminology may vary slightly between cloud providers, in general
    - Root Node (Organization): the topmost level, representing the entire cloud account or organization within the cloud provider's platform
    - Folders: optional levels that allow further categorization and grouping of resources based on project type, department, or any other relevant criteria. Folders can contain other folders or projects.
    - Projects: the core building blocks of cloud environment. They typically represent a specific application, service, or workload being deployed
    - Resources like virtual machines, storage buckets, databases, and network configurations all belong to specific projects.
    - Resources: These are the individual components that make up the cloud environment. Examples include virtual machines, storage buckets, databases, network configurations, and other cloud services
- Benefits of Cloud Resource Hierarchy:
    - Improved Organization
    - Enhanced Security: assigning access control policies at different levels of the hierarchy
    - Optimization: Grouping resources by project allows you to track and manage costs associated with each project
    - Simplified Billing with resources associated with specific projects

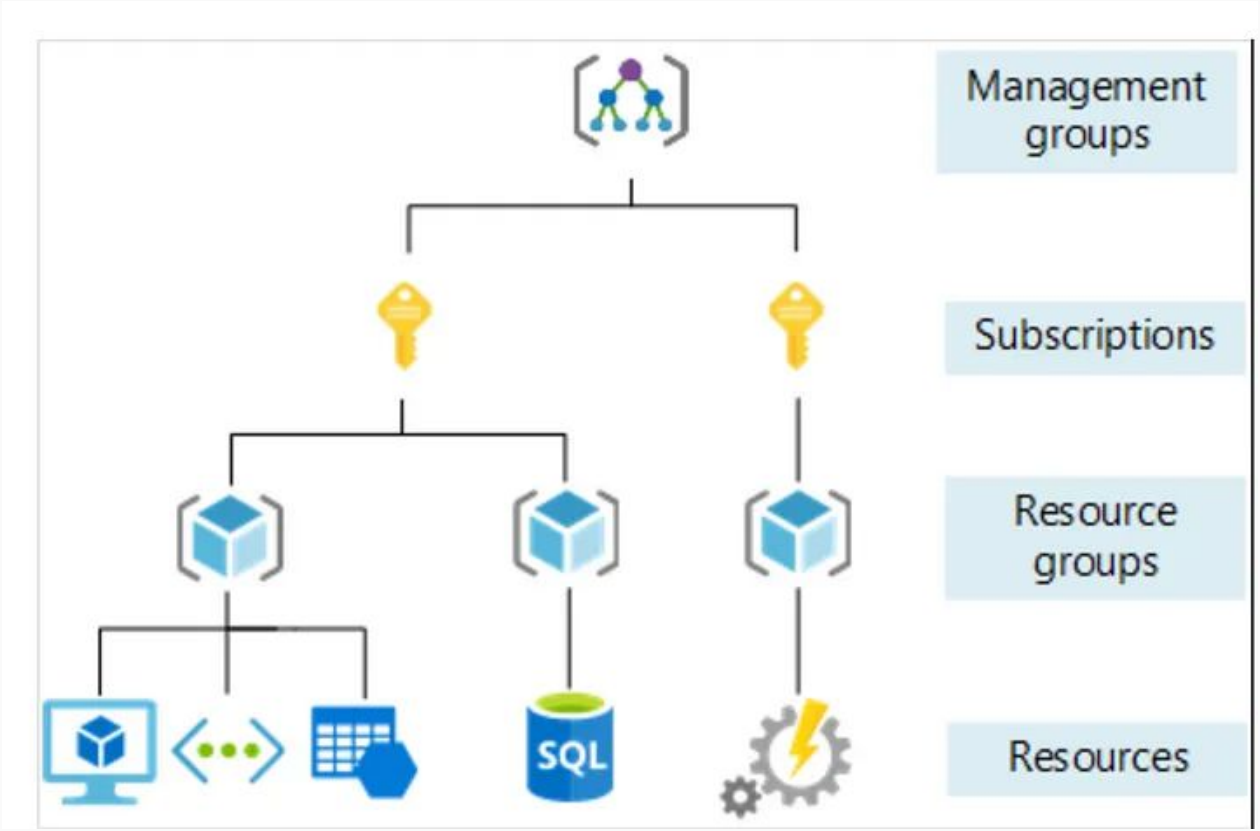
Cloud resource hierarchy[AWS]



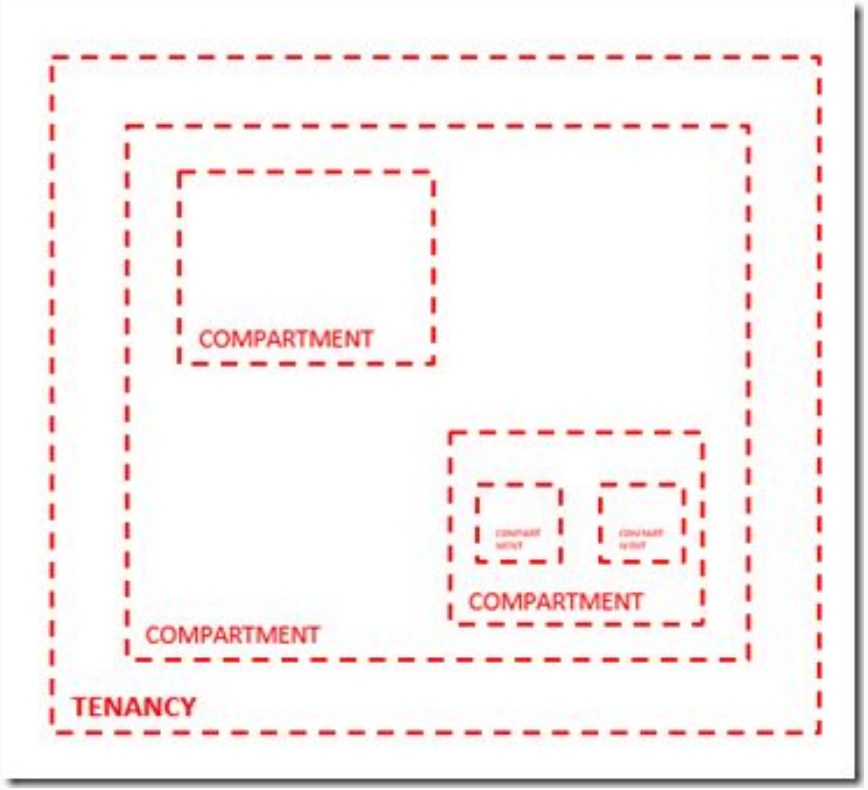
Cloud resource hierarchy (GCP)



Cloud resource hierarchy (Azure)

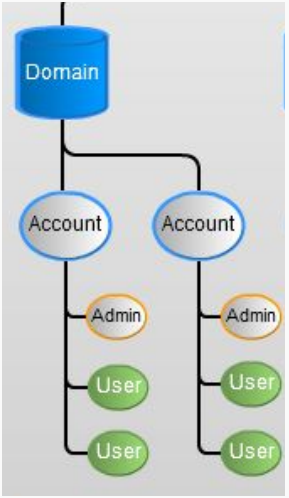


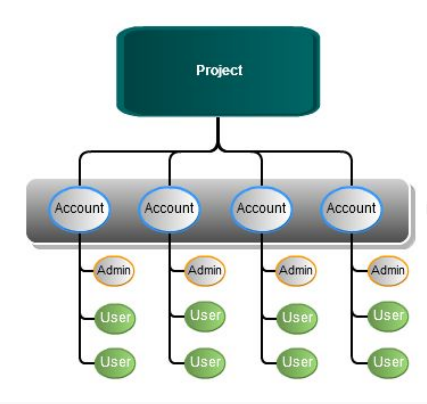
Cloud resource hierarchy: OCI



Cloud resource hierarchy: Zergaw Cloud

- Domains are the equivalent of an organizational unit
- Domains (generally) don't own resources, but they can impose resource limits upon all accounts held within them
- Domains can house projects and accounts, but domains don't really own any instances, volumes, or other resources on their own
- A domain is basically a logical container for other things which can own
- Accounts own resources → If you delete an account all resources associated with it will be removed as well
- Usage is also tracked at an account level hence reporting is available for resources used at an account level.
- Projects are similar to accounts but unique in one special aspect: they can share control of resources amongst multiple accounts
- The resources themselves are owned by the project and are allowed to be manipulated by multiple accounts within the same domain
- Users are the base organizational unit and come in two flavors, User, and Admin.
- The User is enabled to operate resources within an account as defined by the account admin.





Cloud resource hierarchy

aws	Google Cloud	Azure	ORACLE Cloud Infrastructure	ZERGAW CLOUD
Organization (Root)	Organization	Root Management Group	Tenancy	Domain
Organization Unit (OU)	Folder	Management Group	Compartments	Sub-Domain
Account	Project	Subscription	Nested Compartments	Account
		Resource Group		
Resource	Resource	Resource	Resource	Resource

Cloud pricing model

refers to the structure and methodology that cloud service providers use to charge for their services. It outlines how customers are billed for using cloud resources such as compute power, storage, and networking.

Cloud pricing model

Pricing Model	AWS	Azure	GCP
Free Tier	12-month free tier for new customers	12-month free tier for new customers	Free tier with $300 credit for new customers
Pay-As-You-Go	Charges based on actual usage	Charges based on actual usage	Charges based on actual usage
Reserved Instance	Up to 75% discont for reserved instance for 1 or 3 years	Up to 72% discount for reserved instance for 1 or 3 years	Up to 57% discount for committed use contracts for 1 or 3 years
Spot Instance	Up to 90% discount but offers spare capacity	Significant savings on unused capacity	(Preemptible VMs)Significant saving but can be terminated anytime

Other Pricing Models that you may here include: Enterprise Agreement, Per-Second Billing and Sustained Use Discount

Advanced Cloud Services

Beyond providing virtual machine, network, and storage resources, cloud providers provide additional add on services such as:

- Serverless Computing
- Container Orchestration
- Database Services
- Big Data and Analytics
- Data Lake and Data Management, and
- AI and Machine Learning to name a few

Serverless Computing

- an execution model that allows developers to build and **run applications without managing servers** or backend infrastructure
- the cloud provider handles server **provisioning, scaling, and maintenance**
- developers **focus solely on writing code**, and **pay only for the resources** actually used by their applications, promoting scalability and cost-efficiency
- while the name suggests no servers are involved, servers are still used behind the scenes, but their management is abstracted away from the developer.

[GCP] Cloud Function: Fully managed environment for deploying and executing event-driven functions without provisioning or managing server resources.
[AWS] AWS Lambda: Event-driven execution, supports multiple programming languages, automatic scaling.
[Azure] Azure Functions: Similar capabilities with event-driven execution, multiple language support, and integration with other Azure services.

Container Orchestration

- refers to the automated management of containerized applications
- involves tools and processes that **deploy**, **scale**, and **manage** the lifecycle of containers across a cluster of servers
- ensures **efficient** resource utilization, **high availability**, and **simplified** application **deployment** for cloud-based environments.

GCP: Google Kubernates Engine(GKE): Fully managed Kubernetes service that provides a production-ready environment for deploying containerized applications.

AWS: AWS Elastic Kubernetes Service(EKS): Managed Kubernetes with integrated IAM, VPC networking, support for EC2 and Fargate.

Azure: Azure Kubernetes Service(AKS): Managed Kubernetes environment with CI/CD integration, monitoring, and scaling.

Zergaw Cloud Kubernetes Service: Managed self-deploy kubernetes environment

Database Services

- database services provide **on-demand, scalable storage and management solutions** for **application data**
- eliminate the need for businesses to **install, configure, and maintain** their own databases
- variety of options, from **relational databases** for structured data to **NoSQL databases** for unstructured data
- typically billed based on usage, promoting cost-efficiency and allowing businesses to **focus on their core functionalities** rather than database administration.

[GCP] Google Cloud Spanner: A fully managed, scalable, relational database service with strong transactional consistency and high availability.

[AWS] AWS Aurora: MySQL and PostgreSQL-compatible relational database with high performance and availability.

[Azure] Azure Cosmos DB: Globally distributed, multi-model database with horizontal scaling and strong consistency options.

[Telecloud] MySQL Service: Quickly deploy and scale RDS for MySQL on the cloud - a fully managed database service.

Big Data and Analytics

- refers to the process of **analyzing massive and complex datasets** (often exceeding traditional processing capabilities) using **cloud-based infrastructure**
- cloud platforms provide the **scalability, elasticity, and processing power** needed to handle these large datasets efficiently
- allows businesses **to extract valuable insights** from their data, **make data-driven decisions**, and **improve operational efficiency**
- Cloud-based big data analytics solutions offer **cost-effective alternatives** to on-premises infrastructure, enabling businesses of all sizes to leverage the power of big data analytics.

[GCP] BigQuery: A fully managed, serverless data warehouse that allows super-fast SQL queries using the processing power of Google's infrastructure.

[AWS] AWS Redshift: Fully managed data warehouse, columnar storage, parallel query execution.

[Azure] Azure Synapse Analytics: Combines big data and data warehousing, with on-demand query capabilities and integration with Power BI.

Data Lake and Data Management

[GCP] Cloud storage: Unified object storage for developers and enterprises, offering a highly durable and available service for storing any amount of data.

[AWS] AWS S3: Object storage with high durability, lifecycle policies, and integrated security features.

[Azure] Azure Blob Storage: Scalable object storage with tiered storage, lifecycle management, and integrated security.

[Telecloud] Object Storage Service: cloud storage that lets you store virtually any volume of unstructured data in any format and access it from anywhere using REST APIs.

[Zergaw Cloud] Buckets: modern and secure object storage service

AI and Machine Learning

[GCP] AI Platform: Provides tools for developing, deploying, and managing machine learning models. Supports TensorFlow, PyTorch, and scikit-learn.

[AWS] AWS SageMaker: Fully managed service to build, train, and deploy ML models. Integrated Jupyter notebooks, built-in algorithms, and model tuning.

[Azure] Azure Machine Learning: End-to-end machine learning platform with automated ML, designer interface, and model management.

Advantages of the Cloud

1. Cost Efficiency
2. Scalability and Flexibility
3. Disaster Recovery and Backup
4. Collaboration and Accessibility
5. Automatic Update and Maintenance

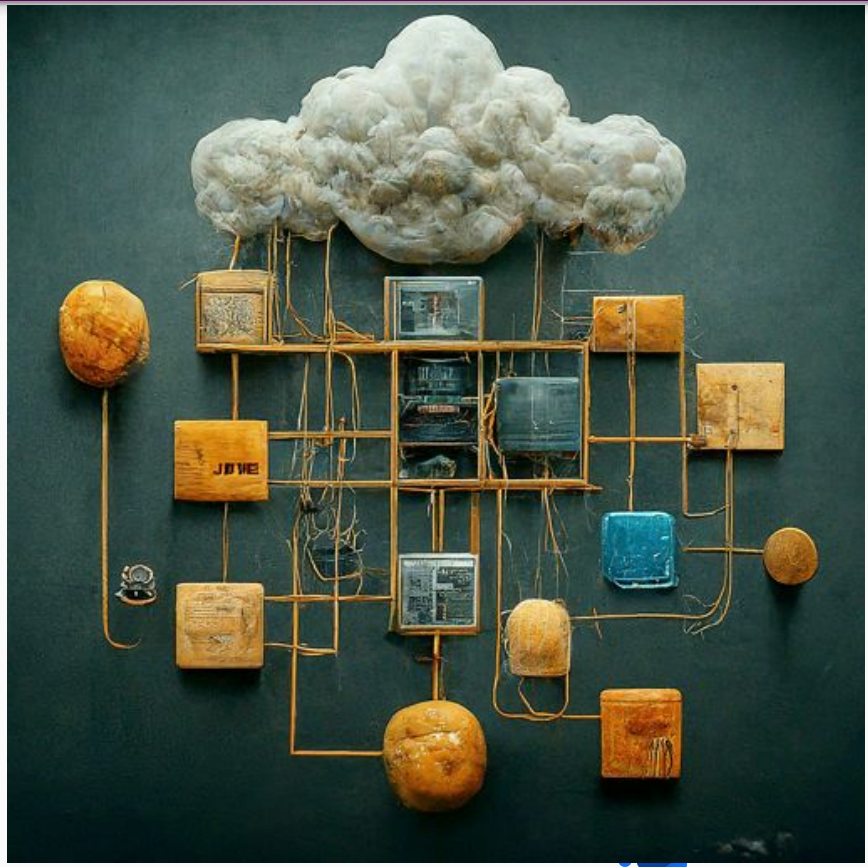
Disadvantages of the Cloud

1. Security and Privacy Concerns
2. Downtime and Reliability
3. Cost Management
4. Limited Control and Flexibility
5. Compliance and Legal Issues

What are the main components that makes up cloud?

Every cloud has some kind of

- Dashboard/Console
- IAM Module
- Image Service
- Compute Module
- Networking Module and
- Storage Module



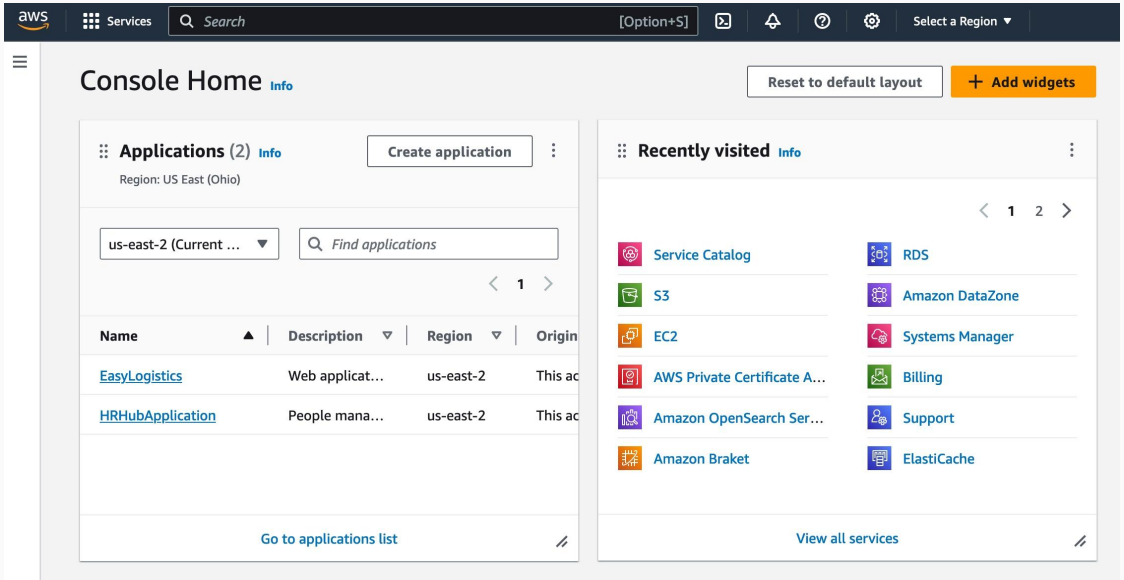
Cloud Console/Dashboard

A cloud dashboard, often referred to as a cloud console, is a web-based user interface provided by cloud service providers that allows users to manage and interact with their cloud resources.

AWS Cloud Console

The AWS Management Console is a web application that comprises a broad collection of service consoles for managing AWS resources

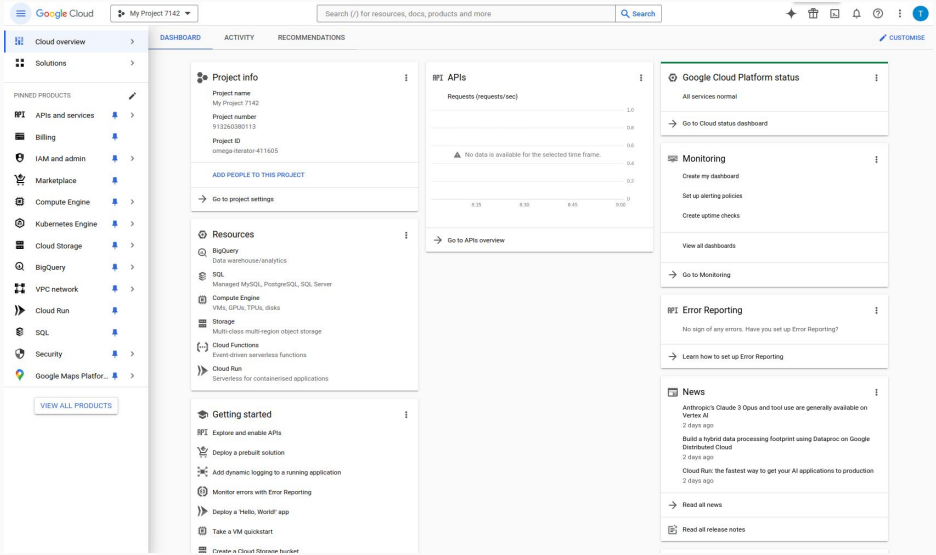
- provides access to each service console
- offers a single place to access the information you need to perform your AWS related tasks



Google Cloud Console

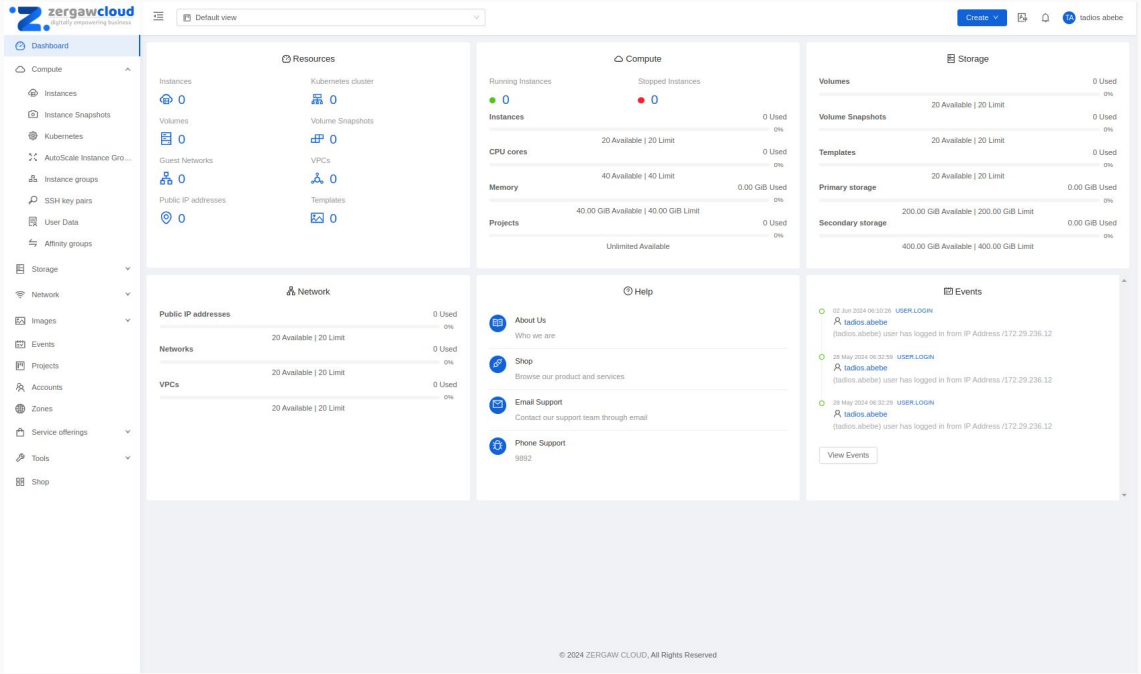
Manage and get insights into everything that powers your cloud application — including web applications, data analysis, virtual machines, datastore, databases, networking, and developer services.

- Deploy, scale, and diagnose production issues in a simple web-based interface
- Search to quickly find resources and connect to instances via SSH in the browser



Zergaw Cloud Console

Helps users of cloud infrastructure to view and use their cloud resources, including Instances, Templates and ISOs, data volumes and Snapshots, Guest Networks, and IP addresses.



ETHIOPIAN IT PARK

CITCOT

ZERGAW CLOUD

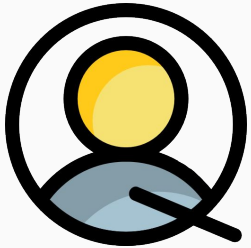
IAM(Identity and access management)

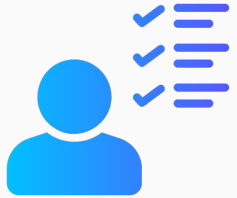
- What is IAM
  - refers to policies, tools, and technologies used to manage digital identities and control access to resources within cloud environment
  - ensures that the right individuals and services have the appropriate access to resource and can perform the necessary actions, while unauthorized access is prevented

IAM

Generally IAM is comprised of the following

- Users
- Groups
- Permissions





IAM

Roles are collection permissions that define what resource are granted; and they can be of the following type:

- Primitive/Basic role
- Predefined role
- Custom role

Image Service

- What is an Image

an image is a server or disk **template** that contains an **operating system, service data, and necessary application software**.

allows users to **create, read, update and delete** their own images, which users either have authored on their own or copied from another source

Image Types

- Standard images provided by the cloud provider
- Contains an OS and various pre-installed applications, and is available to all users

- Created by users and is visible only to the user who create it

- Private image another user has shared with you

- Depending on how and who uploaded the template, they might appear on different places. For example templates or iso's that are made public by other tenants of the cloud will appear on the **community** category of the template section
- Templates that are managed by the cloud administrator and set as **featured**, will appear on the featured category of the template section

Cloud-init

- Utility that **automates the initialization of cloud instance** during system boot.
- Cloud-init is available in various type of **Linux** cloud images by default. i.e ubuntu cloud images, fedora, Debian

- For windows, you will need to use third party software like Cloudbase-Init or write scripts
- Can be used to perform variety of tasks
  - configure hostname
  - install packages on an instance
  - run script
  - store ssh keys and password

cloud-init

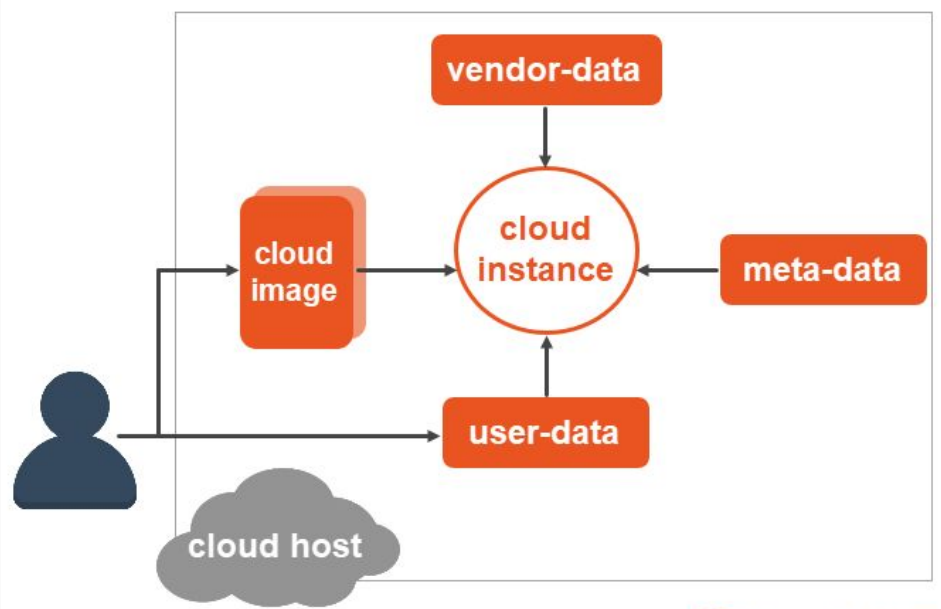


Image Service

Why Imaging Service?

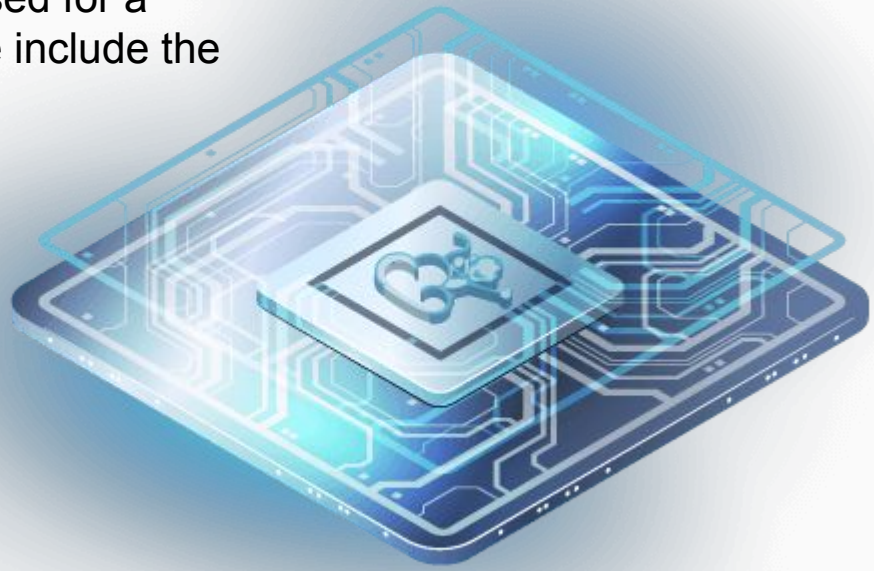
Unified: you can uniformly deploy applications and operating systems ensuring consistency of you application environment

Flexibility: you can easily manage your cloud images though the management console or API

Save time and effort: deploying instances on cloud is much faster and easier when you use images

Compute Service

A compute resource is a measurable amount of computing power that can be requested, allocated, and used for a compute activity. Common computing resource include the CPU and Memory



A computing cloud service is a service or product that can provide computing resource on the cloud

Examples of a compute service include:

- Instances
- Bare-metal Servers
- Auto-scaling
- Containers and container orchestration

Instances

Instances are computer systems that has complete hardware, operating system, and network functions and runs completely isolated.

After an instance is created, you can use it on the cloud similarly to how you would use local computers or physical servers

Instances work with other other products and services to provide **compute**, **storage**, **network** and **images** installation functions

ETHIOPIAN IT PARK

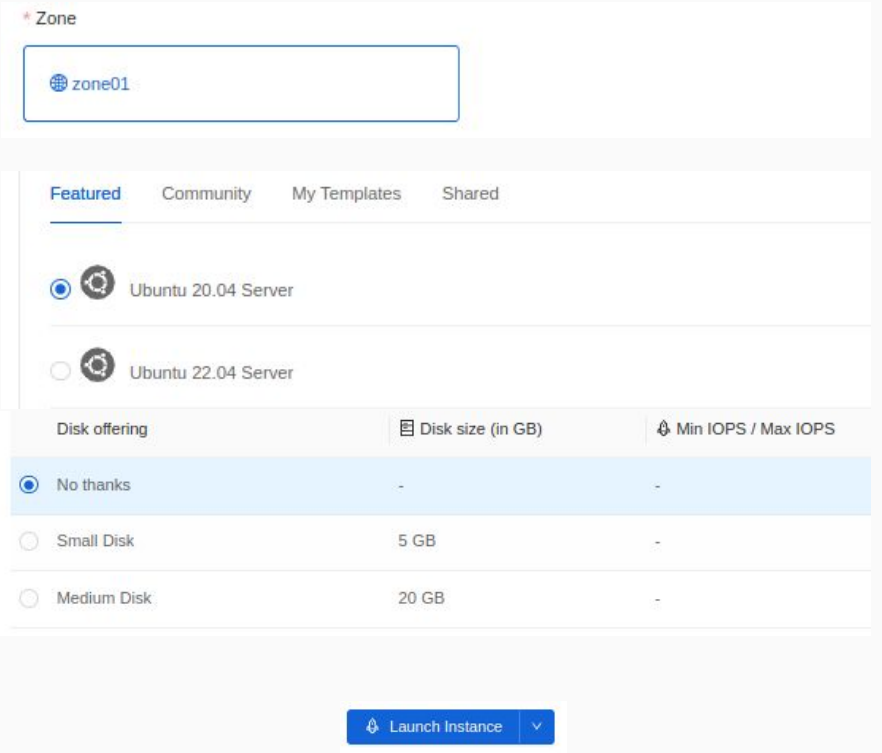
CITCOT

ZERGAW CLOUD

Creating an Instance

Launching an instance with basic configuration

1. Click compute -> instance -> add instance
2. Select the zone
3. Select the template or ISO
4. Select compute and disk offering
5. Select/add a network
6. Click on launch instance and your instance will be created and started



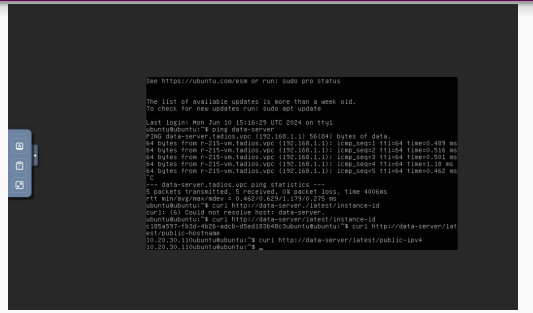
Accessing an instance

- You can access your instance through different methods based on the operating system
- Generally you can access your all your Instances through the cloud console VNC proxy, compute -> instance -> console button

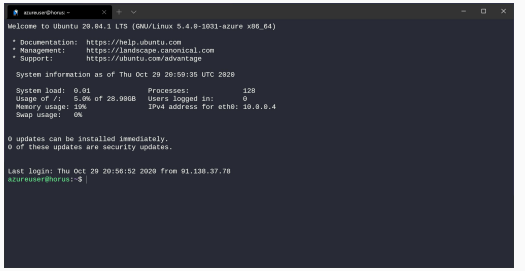
To access an instance directly over the network:

- The instance must have some port open to incoming traffic
- The network firewall should allow incoming traffic for that port
- Configure port forwarding on the cloud network
- Based on the operating system you are using you need to install or enable additional packages that are relevant for the connection

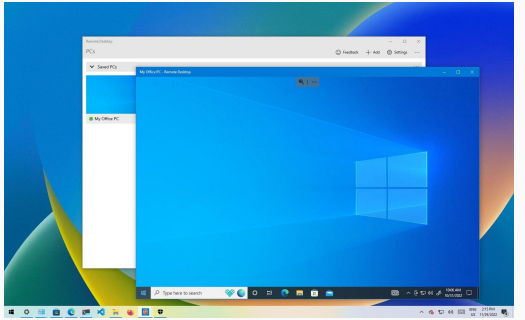
Console



SSH

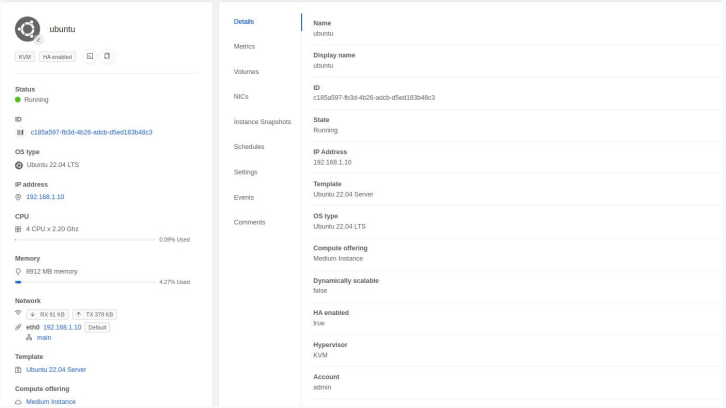


RDP



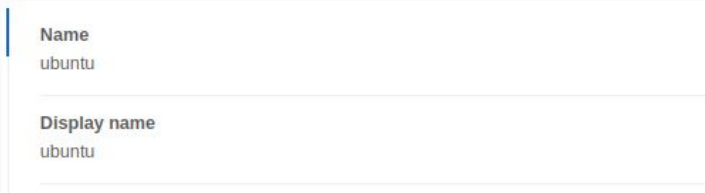
Working with instances

The cloud console provides several guest management operations for end Users. Instances may be stopped, started, rebooted, and destroyed



Instances can have two names

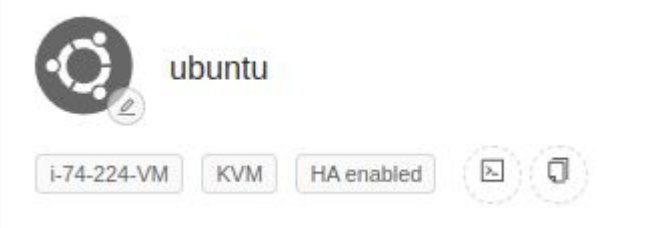
- Display name: the name displayed in the cloud console web UI, set by the user(defaults to instance name)
- Name: host name that the DHCP server assigns to the instance. can be set by the user(defaults to instance name)

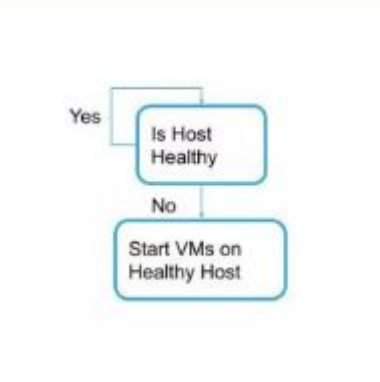


Instance High Availability

Instance can be configured to be high available

- HA enabled instances is **monitored** by the system. if the system detects that the instance is down, it will attempt to restart the instance.
- The cloud management cannot distinguish a guest Instance that was shut down by the User (such as with the “shutdown” command in Linux) from an Instance that shut down unexpectedly.
- If an HA-enabled Instance is shut down from inside the Instance, the cloud manager will restart it. to shutdown an ha-enabled instance, you must go through the cloud manager or cloud API





Instance Life cycle

creating -> created -> starting -> running -> stopping -> stopped -> destroyed -> expunging -> expunged

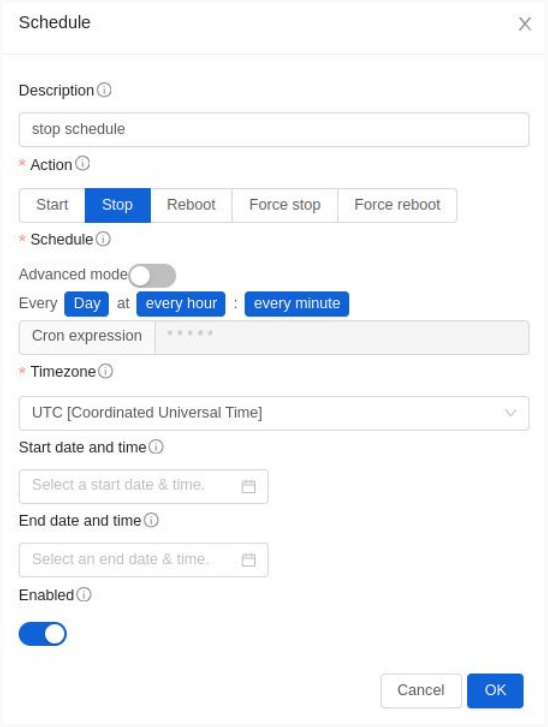
- Once an instance is created, you can **stop**, **restart**, or **delete** it as needed.
- A **stop** will attempt to gracefully shutdown the operating system via an ACPI 'stop' command
- A stop with **force** has the same effect as pulling out the power cord from a physical machine
- You can **destroy** an instance by going compute -> instance -> destroy, optionally both expunging and the deleting of any attached volume can be handled
- A running Instance will be abruptly stopped before it is deleted
- When destroying an instance, if the **expunge** option is not enabled the instance will still consume logical resource for the next 24 hours, and can be recovered by the root admin

Scheduling Instance Operations

After an instance is created, you can schedule instance life cycle operations using Cron expressions
Operations that can be scheduled are start, stop, reboot, force reboot, force stop.

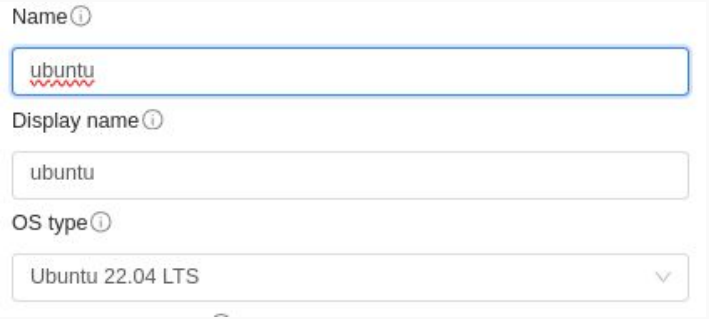
To schedule an operation on an instance through the cloud console

1. Click the instance you want to schedule operations for
2. On the instance details page, click the schedule button
3. Click add schedule button
4. Give it a description
5. Select the action
6. Select the frequency using cron format
7. Select timezone
8. Select start and end date
9. Click on OK to save the schedule



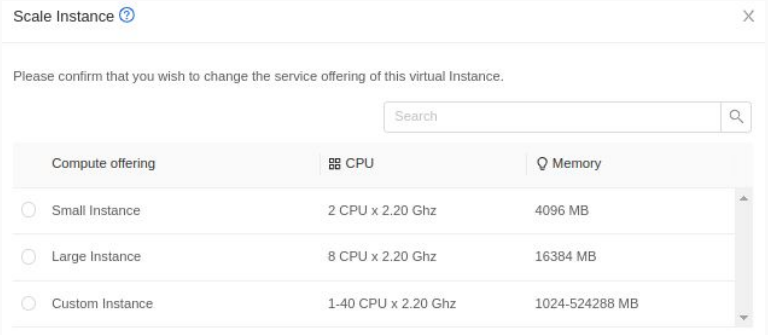
Editing an Instance

- After an instance is created, you can modify the display name, operating system, and group it belongs to.
- You first need to stop the instance to change modify som parameter



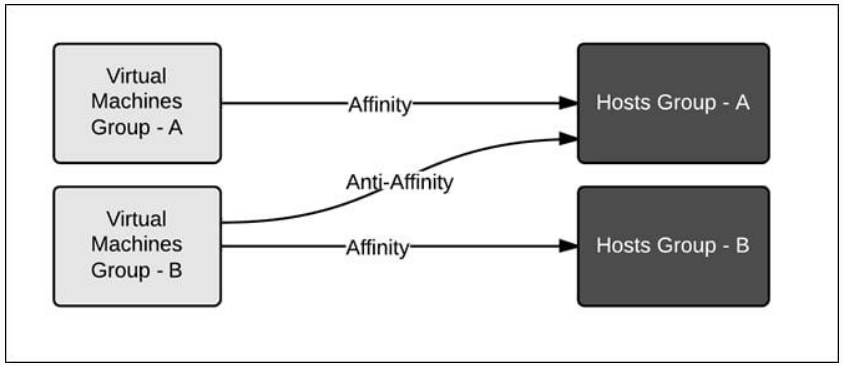
To upgrade or downgrade the level of compute resource available to an instance, you can change the instance compute offering

1. Go to compute -> instance
2. Choose the instance that you want to work with
3. Stop the instance
4. Click the change service button
5. Select the offering you want to apply to the selected instance
6. Click Ok



Instance Affinity Group

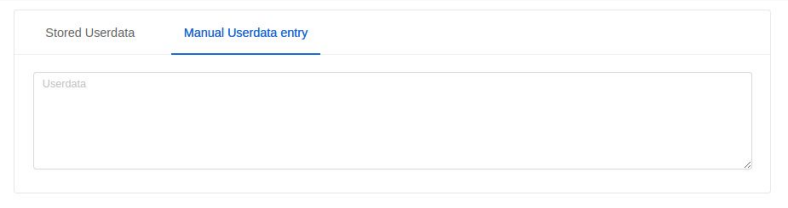
- By defining affinity group and assigning instance to them, the user or administrator can influence(but not dictate) whether instances should run on either the same or separate host
- Instance with the same host anti-affinity type won't be on the same host, which serves to increase fault tolerance
- Instance with the same host affinity type must run on the same host, which can be useful in ensuring connectivity and low latency between guest instances

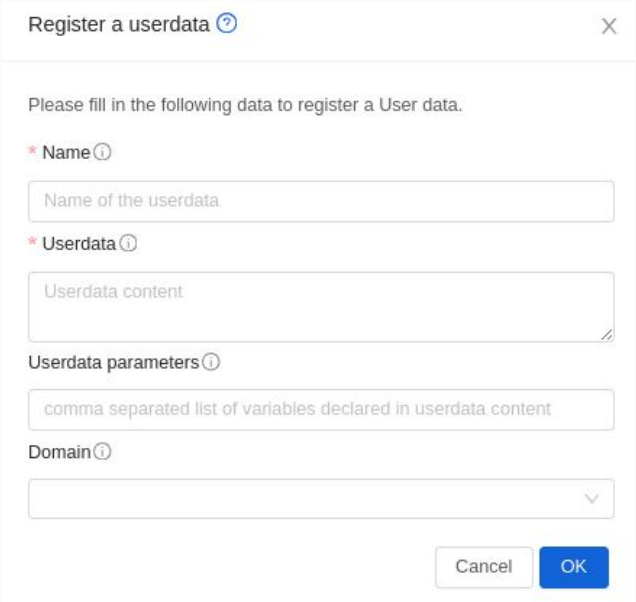


- The non strict host affinity groups are similar but more flexible that host affinity, in which case instances are deployed to same or different hosts as long as there are enough hosts to satisfy the requirement, otherwise they might be deployed to the same host
- You can add an instance into an affinity group on instance creation stage by going into advanced option or stopping the instance and clicking on the change affinity button

User Data and Cloud-init with Instances

- Use data is the user custom content exposed to a guest instance by the currently deployed and running cloud infrastructure.
- Its purpose is to provide additional data for the instance to customize it as much as you need





User Data Examples

Automatically Update and Upgrade System

```
# cloud-config
package_update: true
package_upgrade: true
```

Install Packages

```
#cloud-config
packages:
  - apache2
  - htop
  - tmux
```

Install Packages

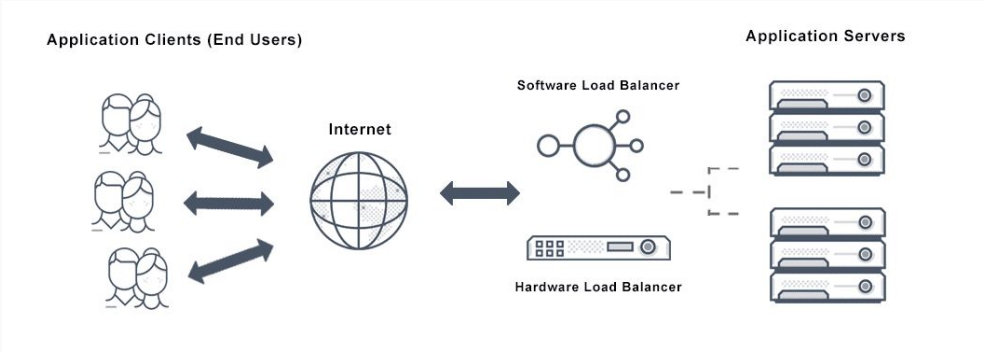
```
#cloud-config
write_files:
- content: |
    SMBDOPTIONS="-D"
  path: /etc/sysconfig/samba
```

Creating a User

```
#cloud-config
users:
- default
- name: username
    gecos: username
    groups: sudo
    lock_passwd: false
    plain_text_passwd: password
    sudo: ALL=(ALL) NOPASSWD:ALL
    shell: /bin/bash
```

Load Balancing and Auto scaling

- **Load balancing** provide business continuity, and enable seamless resource movement within a cloud environment.
- **Auto Scaling** allows you to scale your back-end services or application Instances up or down seamlessly and automatically according to the conditions you define.

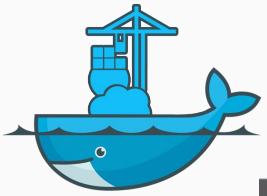


Prerequisites

- ensure the necessary templates are prepared before configuring auto scaling
- prepare the necessary application for the load balancing and auto scaling
- prepare the network for you load balancing and auto scaling requirements

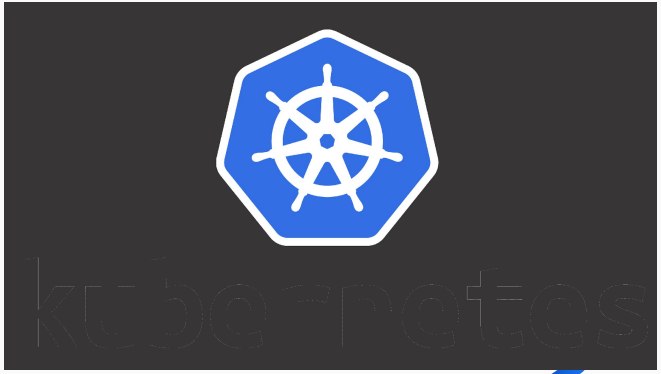
Container Orchestration

A container: is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.



Container orchestration: automatically provisions, deploys, scales, and manages containerized applications without worrying about the underlying infrastructure

Kubernetes: is an open-source Container Management tool that automates container deployment, container scaling, descaling, and container load balancing (also called a container orchestration tool).



Components of Kubernetes

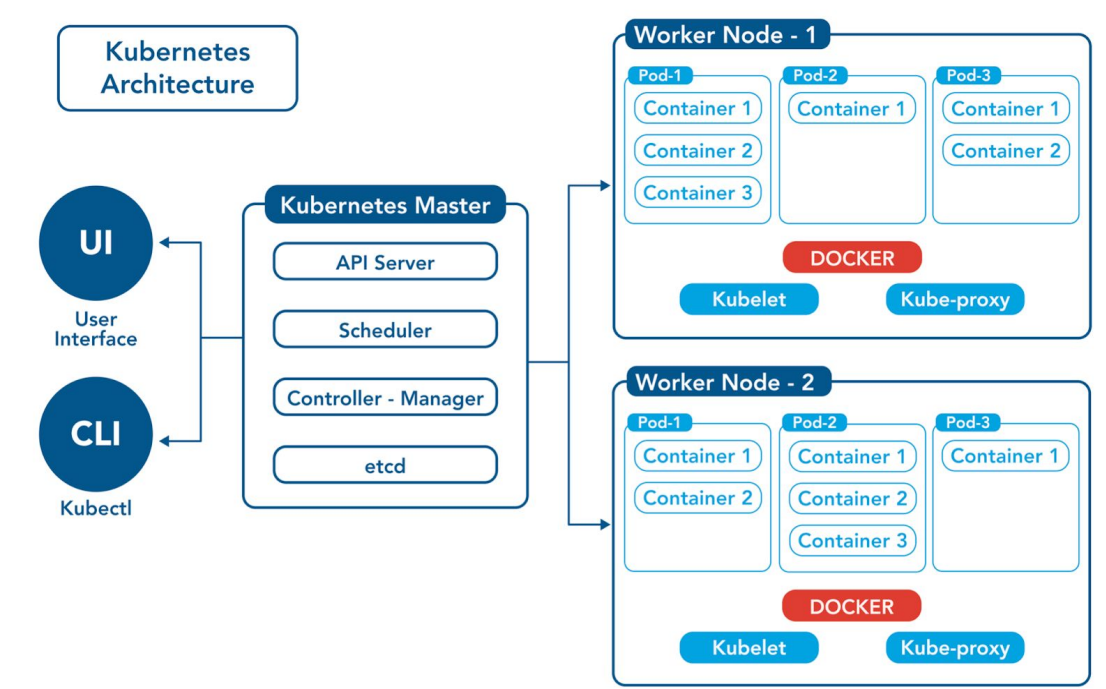
Kubernetes - Master Node: responsible for managing the entire cluster, coordinates all activities inside the cluster, and communicates with the worker nodes to keep the Kubernetes and your application running.

- Consists of components like: API server, scheduler, controller manager, etcd

Kubernetes - Worker Node: contains all the necessary services to manage the networking between the containers, communicate with the master node, and assign resources to the containers scheduled.

- Consists of components like: Kubelet,kube-proxy, pod, docker

The main components of kubernetes are:

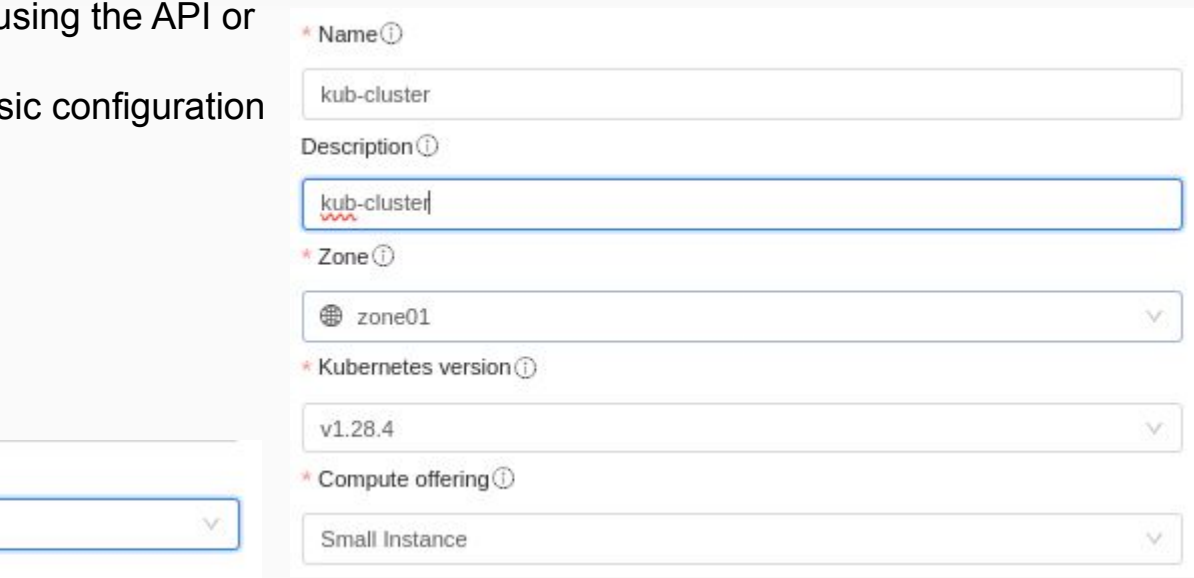


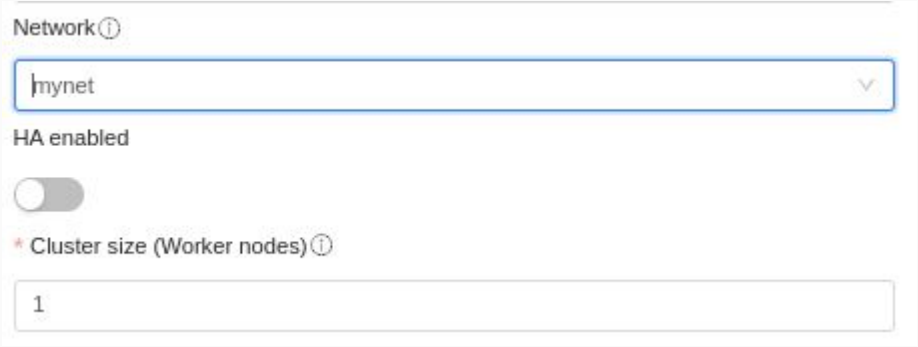
Creating a Kubernetes Cluster

New Kubernetes cluster can be created using the API or via the UI.

Launching an kubernetes cluster with basic configuration

1. Give it a name and description
2. Select a zone
3. Select a kubernetes version
4. Select compute offering
5. Select network
6. Click Ok



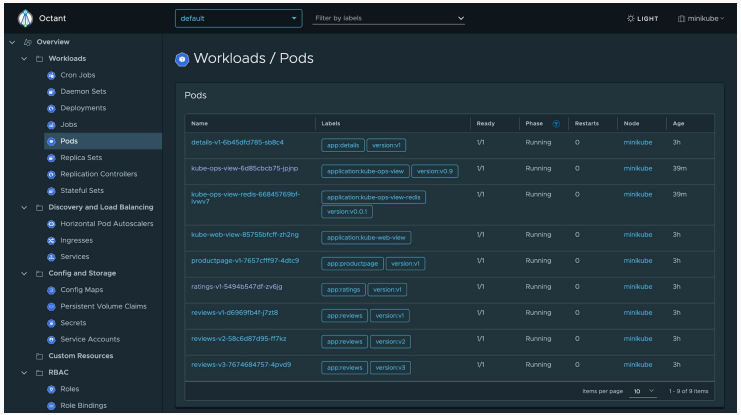


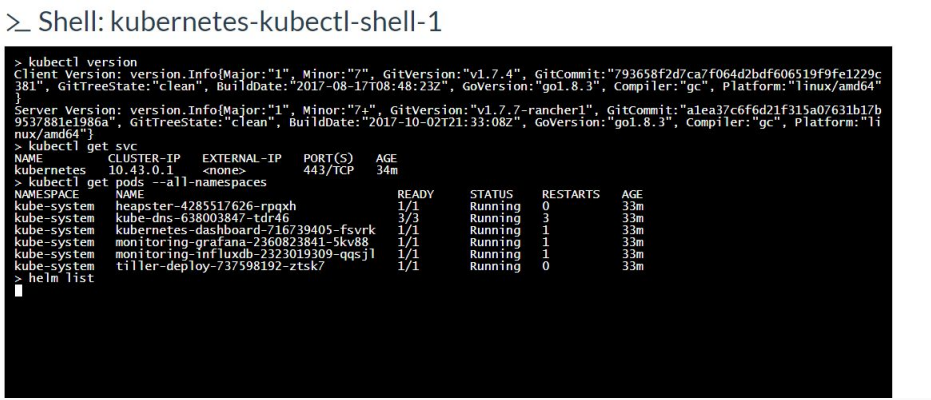
Accessing the kubernetes cluster

Instruction for accessing the kubernetes cluster can be found inside the kubernetes cluster access tab

- You first need to download the kubernetes cluster config from that tab
- Then download the kubectl tool and make it executable
- Run you kubernetes commands using the above executable and config file
  - *./kubectl --kubeconfig kube.conf get nodes --all-namespaces*
- To access the kubenetes dashboard, you need to run kubernetes proxy locally
  - *./kubectl --kubeconfig kube.conf proxy*
- Then open you browser to the following location

http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/



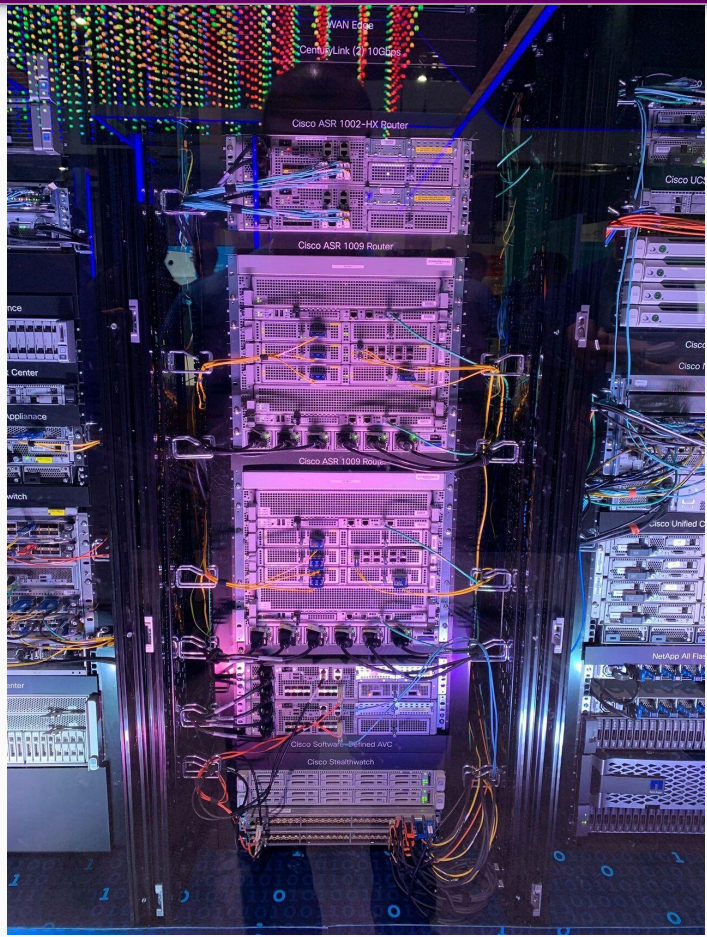


Network Service

A network is a system that **connects multiple computers** or other devices together so that they can **communicate and share resources** with each other. A network can be a different type, such as a local area network, a wide area network or the internet.

Different cloud provides use

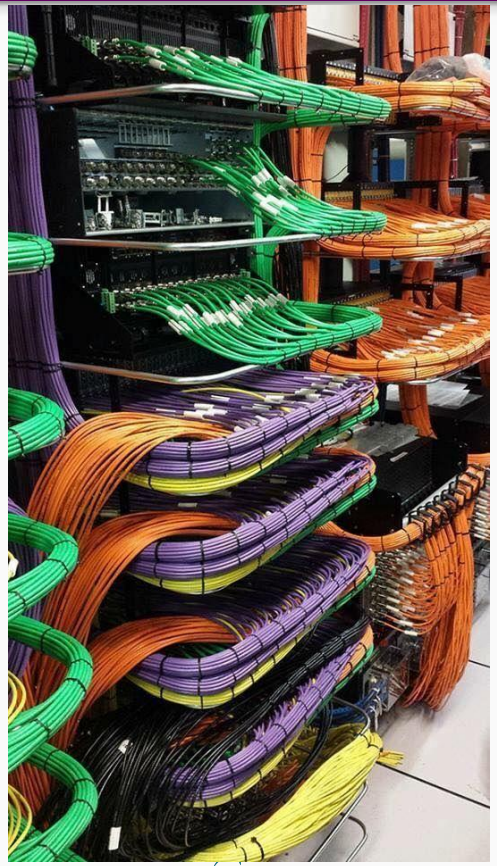
- **VPC** to provide an **isolated network environment**,
- Load balancer to distribute incoming traffic across multiple backend servers,
- NAT gateways to provide network address translation for cloud servers,
- Elasic IPs to provide independent public IP addresses for accessing the internet,
- **VPN**s to establish an encrypted channel between an on-premises data center and the cloud and so on



ETHIOPIAN IT PARK

CITCOT

Zergaw Cloud Network Categories



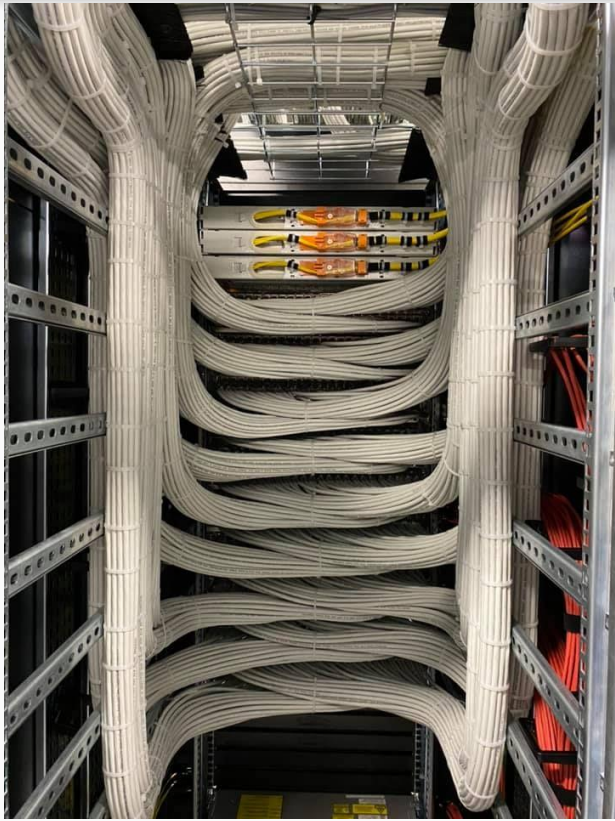
Guest networks carry the network traffic between the guest instances and their respective gateways.

- Each of the networks are isolated and therefore the same IP addresses can be reused in each network.
- This network can carry guest traffic only between Instances within one zone.
- Instances in different zones cannot communicate with each other using their private IP addresses; they must communicate with each other by routing through a public IP address.

Zergaw Cloud Network Categories

Public networks carry the network traffic between the guest instances, and the internet

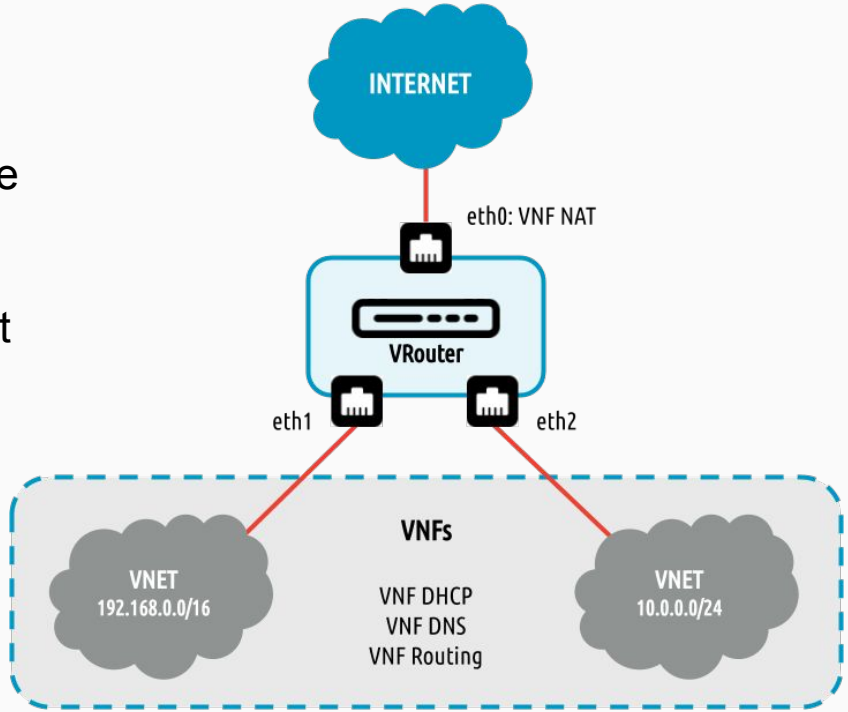
- The virtual router takes an IP address from the public range assigned by the cloud manager, and places it on the external face of the virtual router, thereby providing an externally accessible address.
- The cloud manager creates a virtual router which sits between the Guest network and the public network.
- External users or devices can connect to the isolated network behind the virtual router using this address.
- The cloud manager must provide one or more external (public) network ranges which will be assigned to the external interfaces of the virtual routers.



Virtual Routers

A virtual router is a special Instance that runs on the hosts.

- Each virtual router in an isolated network has multiple network interfaces.
- If multiple public VLAN is used, the router will have multiple public interfaces.
- The virtual router provides DHCP and will automatically assign an IP address for each Guest Instance within the IP range assigned for the network.
- The user can manually reconfigure Guest Instances to assume different IP addresses.
- Source NAT is automatically configured in the virtual router to forward outbound traffic for all Guest Instances



Network Offerings

Network offerings define the network services and configurations that are available for virtual networks.
These offerings specify the features and capabilities that networks can have, such as firewall rules, load balancing, VPN access, and more.

Components of Network Offering

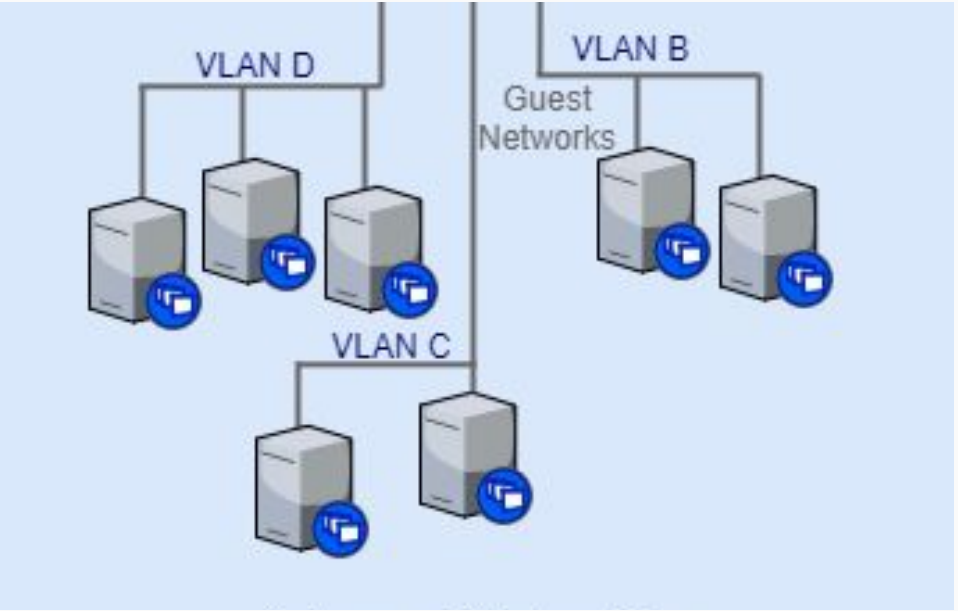
- Traffic types
- Network services
- Service providers
- Availability

Different type of Network Services

- DHCP
- DNS
- Firewall
- Load Balancer
- VPN
- Port forwarding
- Static NAT

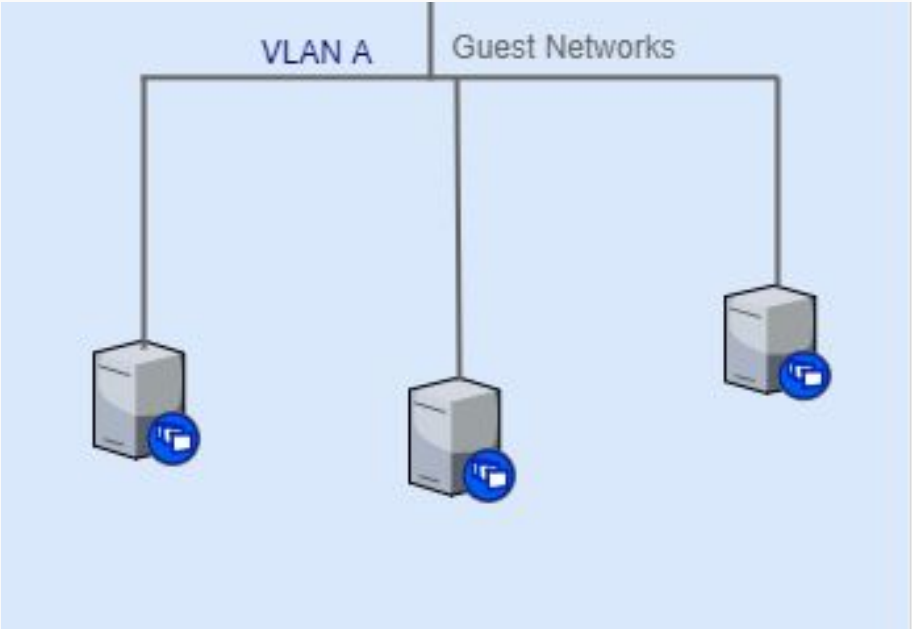
Isolated Networks

- An isolated network can be accessed only by Instances of a single account.
- Resources such as VLAN are allocated and garbage collected dynamically
- There is one network offering for the entire network
- The network offering can be upgraded or downgraded but it is for the entire network



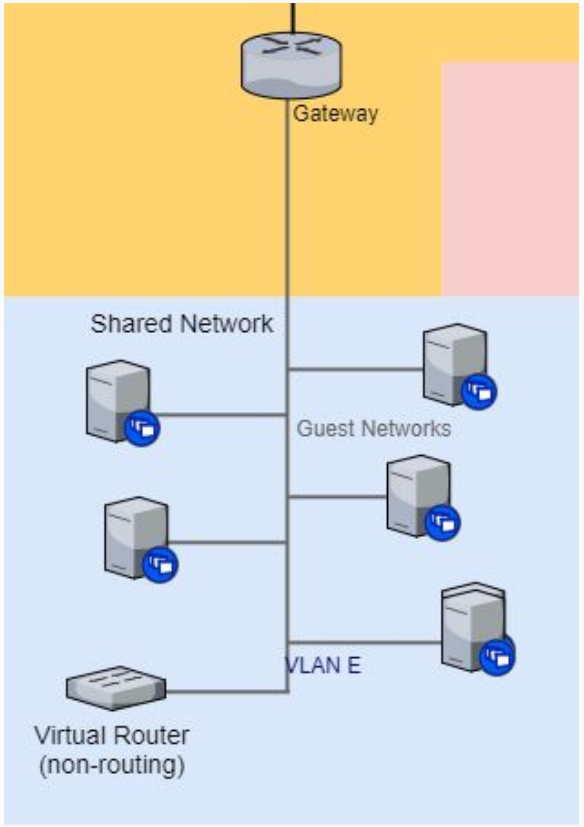
L2 Networks

- L2 networks provide network isolation without any other services
- It is assumed that the end user will have their own IPAM in place, or that they will statically assign IP addresses.
- Virtual Router is not required, and VMs deployed to the network will get their IP addresses and other network services from these external appliances.



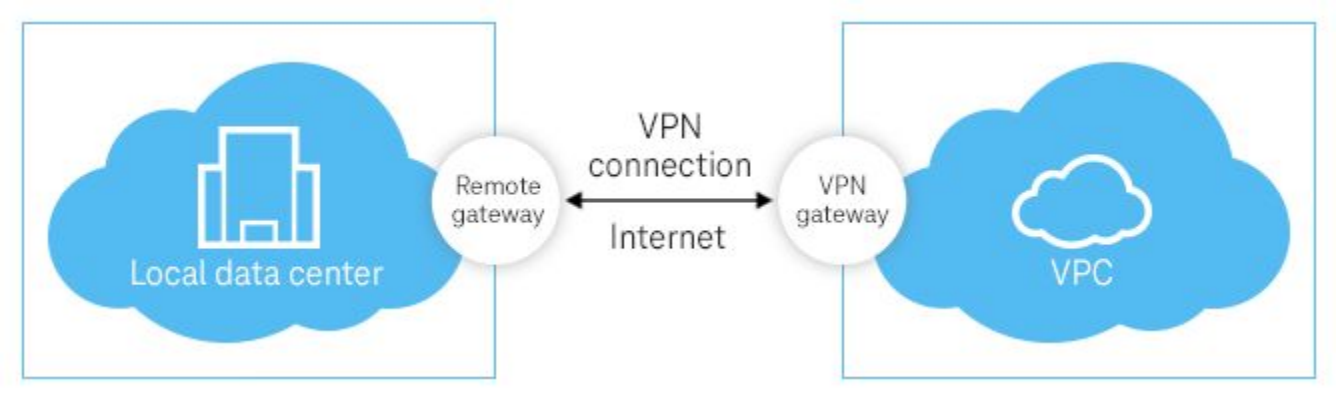
Shared Networks

- A shared network can be accessed by Instances that belong to many different accounts.
- Shared Networks can be designated to a certain domain
- Shared Network resources such as VLAN and physical network that it maps to are designated by the administrator
- Virtual router is involved for DHCP and DNS services but it is out of the data path



VPN(Virtual Private Network)

VPN or **Virtual Private Network** provides secure, encrypted connections over public networks. It ensures data privacy and security for users and applications.



On zergaw cloud there are mainly two types of VPN
Remote Access VPN: Allows individual users to securely connect to the cloud network.
Site-to-Site VPN: Connects two separate networks over the internet.

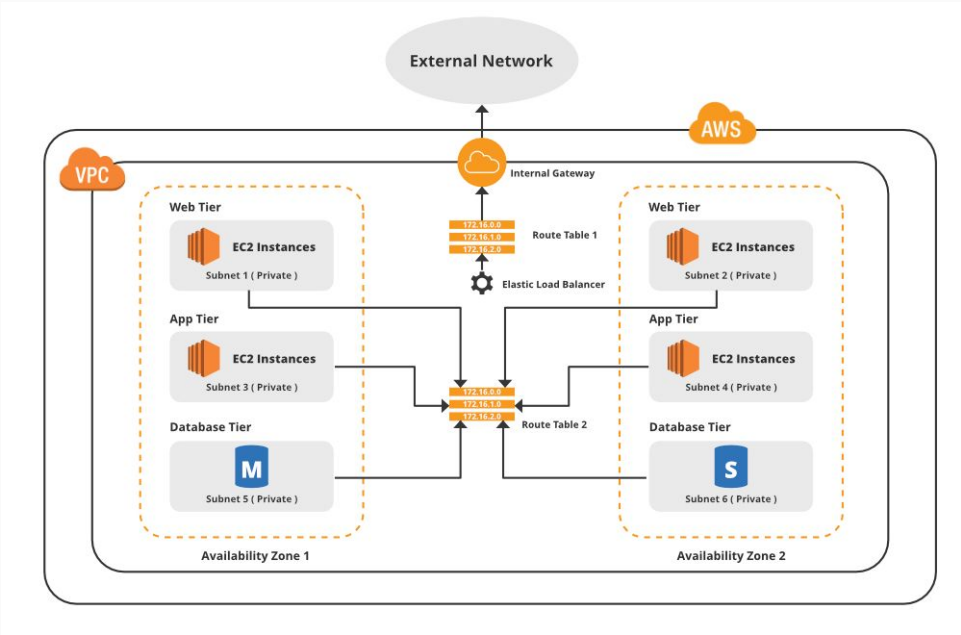
VPNs are essential for secure cloud network connectivity. Cloud platforms provides robust VPN capabilities for various needs. Implementing VPNs enhances security, privacy, and access control in cloud environments.

VPC (Virtual Private Cloud)

- Virtual Private Cloud lets you provision an architecture that resembles a traditional physical network.
- VPC provide private multi-tiered virtual networks
- It supports inter VLAN routing

VPC implements:

- Tiered isolation
- ACLs
- Site to site ipsec VPN
- User VPN
- Internal and External Load Balancer



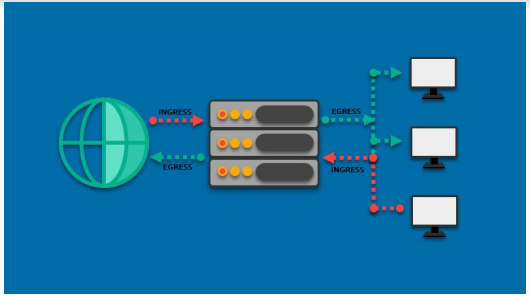
VPC Components

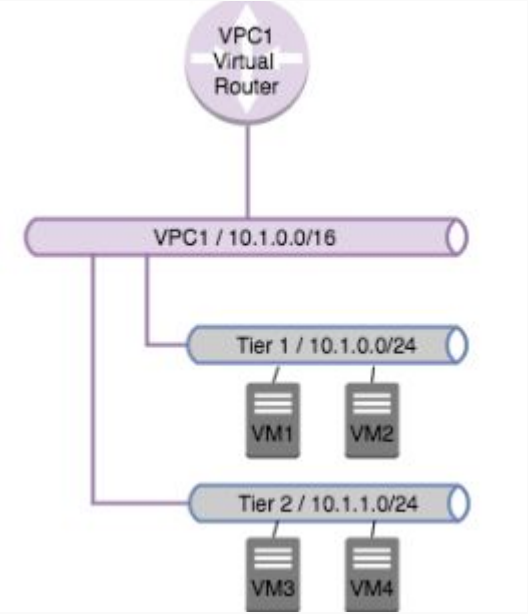
Network Tiers: isolated network, each with unique VLAN and CIDR

Virtual Router: Connects all the VPC components

Network ACLs: is a group of network access control list items

- Network ACL items are nothing but numbered rules that are evaluated in order, starting from lowest numbered rule.
- This rules determine whether traffic is allowed in or out of any tier associated with the network ACL.
- You need to add the network ACL items to the network ACL, then associate the network act with a tier.
- Network ACL is associated with a VPC and can be assigned to multiple VPC tiers within a VPC.
- A tier is associated with a network ACL at all times, and each tier can be associated with only one ACL.
- The default network ACL is used when no ACL associated
- The Default behavior is all the incoming traffic is blocked and outgoing traffic is allowed from the tiers
- Default network ACL can not be removed or modified.

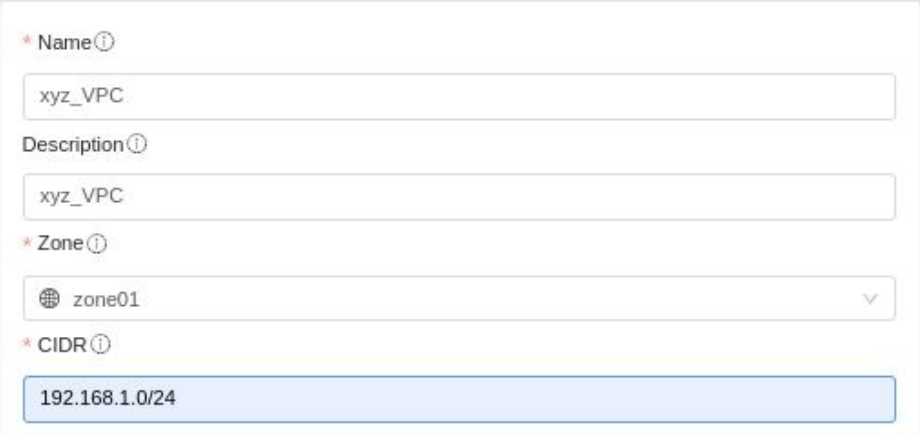


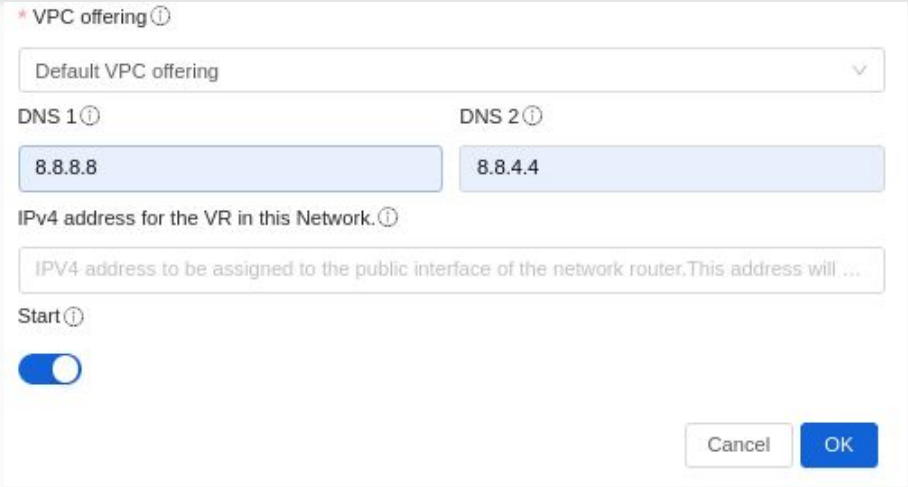


ETHIOPIAN IT PARK

CITCOT

Creating a VPC





Go to Network > VPC > Add VPC

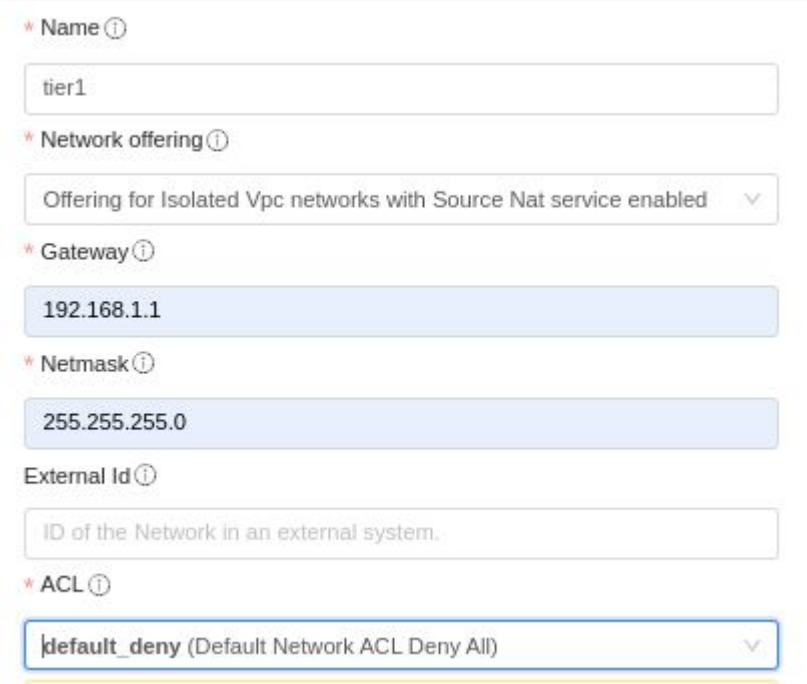
1. Give it a name and description
2. Select a zone
3. Specify the desired CIDR
4. Select the default VPC offering
5. Set DNS
6. And click on OK

Creating multiple tiers with a VPC

Go to Network > VPC > Created VPC > Networks > Add new Network Tier

1. Give it a name
2. Select the default network offering
3. Specify the gateway
4. Specify the Netmask
5. Select your desired ACL
6. And click on OK

- Repeat the above to create a different tier.
- When creating different tiers within you VPC, all subsequent network ranges should be a subset of the VPC CIDR



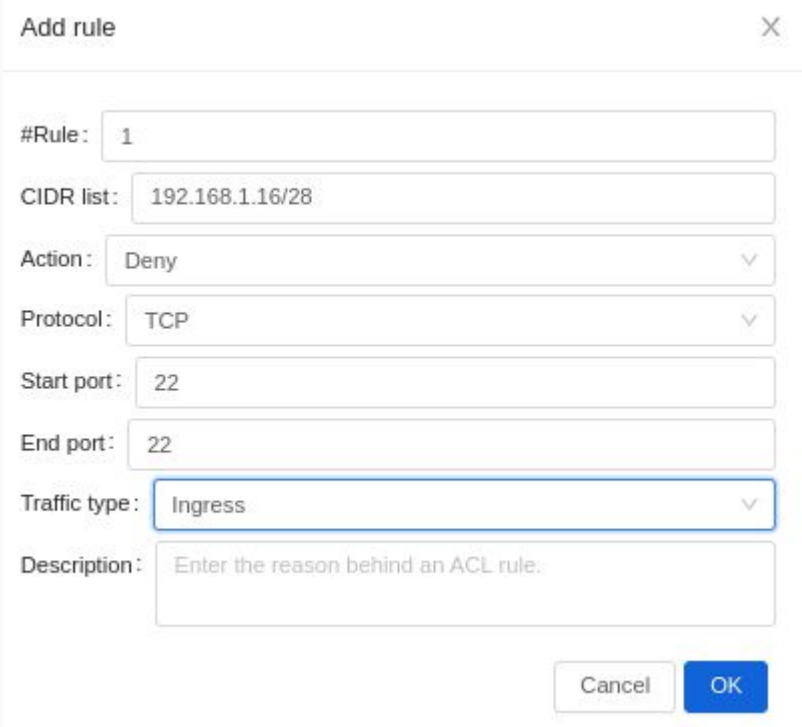
Creating and Modifying Network ACL

Go to Network > VPC > Created VPC > Network ACL lists > Add Network ACL list

1. Give it a name
2. Give it a description
3. And click on OK

- Once the ACL is created click on the newly create ACL and add ACL rules by going on the ACL list rules tab > Add ACL

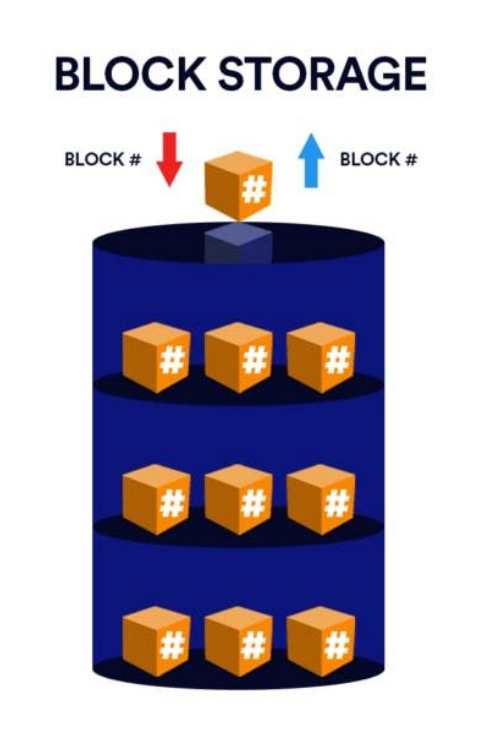
1. Give the rule an id
2. Specify the CIDR list
3. Select the Action
4. Select the Protocol
5. Define the start and end ports
6. Select the traffic type and
7. Give it a description
8. then Click OK



Types of storage in the cloud

Block Storage

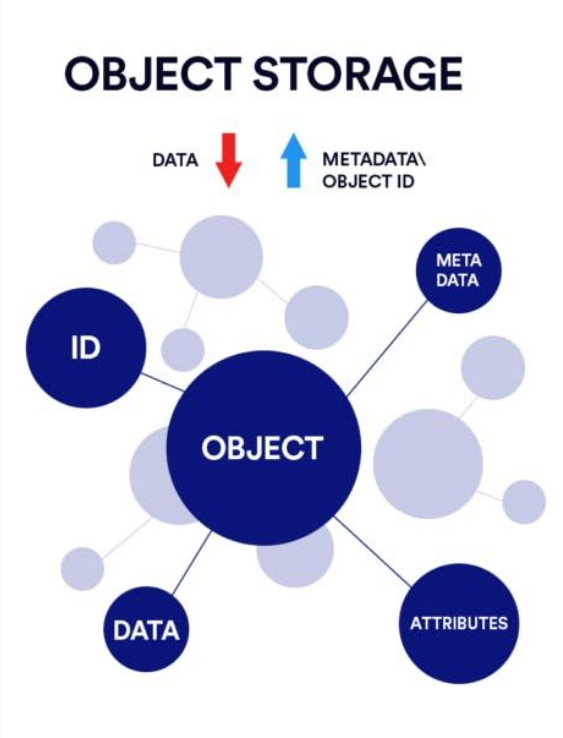
- Block storage chops data into blocks and stores them as separate pieces.
- It is often configured to decouple the data from the user's environment and spread it across multiple environments that can better server the data.
- Then when data is requested by the underlying software reassembles the blocks of data from these environments and present them back to the user.
- It is usually deployed in storage-area network environment.
- Different cloud providers have different names for their block storage services. i.e
  - AWS: Elastic Block Storage
  - GCP: Persistent Disk
  - Azure: Managed Disk



Types of storage in the cloud

Object Storage

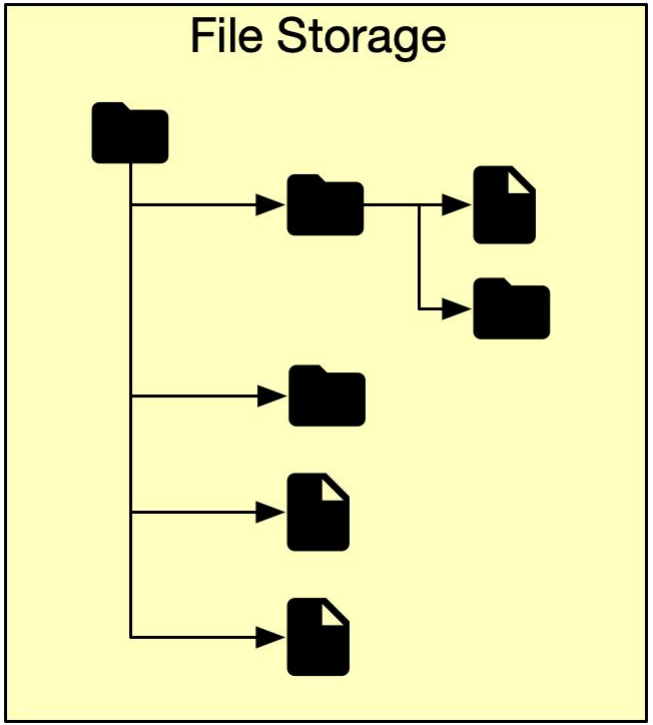
- A flat structure in which files are broken into pieces and spread out among hardware, the data is broken into discrete units called objects and kept in a single repository, instead of being kept as files in folders or as blocks on servers.
- Objects stored on object storage systems have an extremely detailed meta-data like age, securities and access contingencies.
- Object storage requires a simple HTTP API which is used by most clients in all languages
- Different cloud providers have different names for their object storage services. i.e
  - AWS: S3
  - GCP: Cloud Storage
  - Azure: Blob Storage



Types of storage in the cloud

File Storage

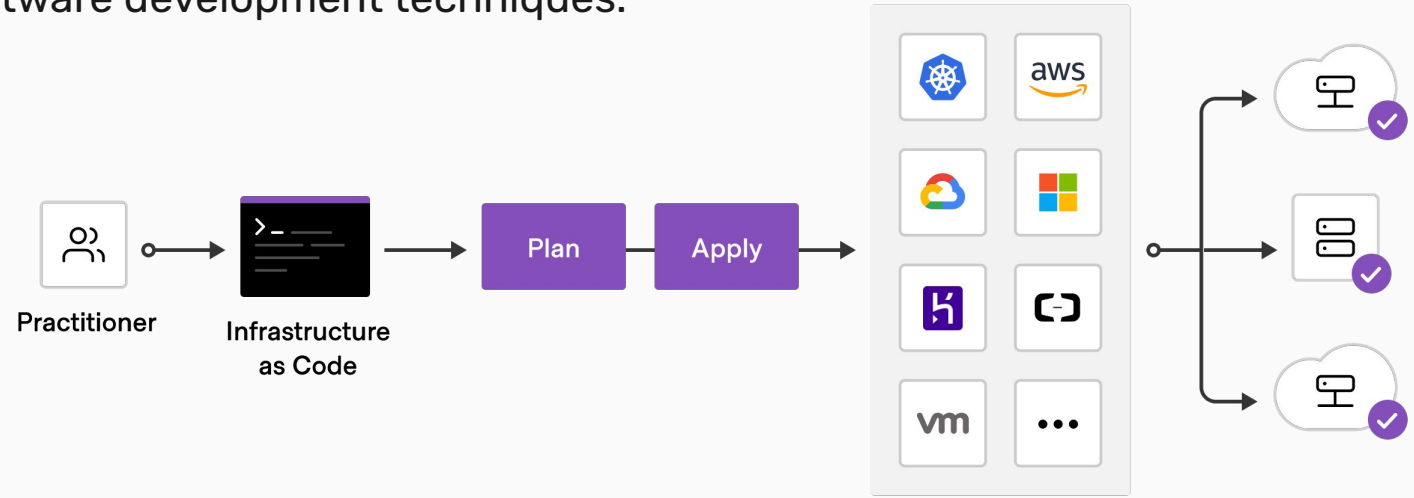
- In file storage document is arranged in some type of logical hierarchy
- Data is stored as a single piece of information inside a folder
- It is the oldest and most widely used data storage system for direct and network-attached Storage..
- Different cloud providers have different names for their file storage services. i.e
  - AWS: Elastic File Storage
  - GCP: Cloud File store
  - Azure: Azure Files



IaC(Infrastructure as Code)

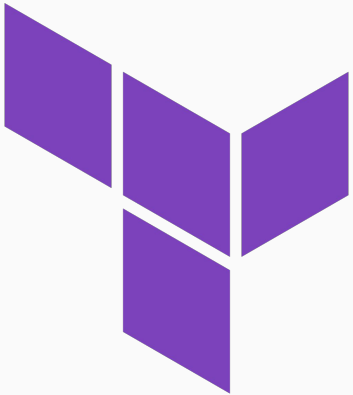
What is IaC?

- Infrastructure as Code is a practice by where traditional infrastructure management techniques are supplemented and often replaced by using code based tools and software development techniques.



Terraform

- Terraform is an open-source infrastructure code software tool that provides consistent work flows to manage hundreds of cloud services, codifying cloud APIs into declarative configuration files.
- Terraform allows infrastructure to be expressed as code in a simple, human-readable language called HCL (HashiCorp Configuration Language).
- It reads configuration files and provides an execution plan of changes, which can be reviewed for safety and then applied and provisioned.
- Extensible providers allow Terraform to manage a broad range of resources, including IaaS, PaaS, SaaS, and hardware services.



HashiCorp
Terraform

Terraform Example

main.tf

```
terraform {
  required_providers {
    cloudstack = {
      source = "cloudstack/cloudstack"
      version = "0.5.0"
    }
  }
}
provider "cloudstack" {
  api_url = var.api_url
  api_key = var.api_key
  secret_key = var.secret_key
}
variable "api_url" {
  description = "API URL"
  type = string
  default = "http://172.29.236.10/client/api"
}
variable "api_key" {
  description = "API key"
  type = string
  default = "xxx"
}
variable "secret_key" {
  description = "Secret key"
  type = string
  default = "xxx"
}
```

instance.tf

```
resource "cloudstack_instance" "instance" {
  name = "terraform-instance"
  service_offering = "Small Instance"
  template =
"a65d8d02-4abb-4fb3-a66a-5889ae379e3a"
  network_id =
cloudstack_network.isolated_net.id
  zone = "zone01"
  expunge = true
}
```

network.tf

```
resource "cloudstack_network" "isolated_net" {
  name = "terraform-network"
  cidr = "10.0.0.0/24"
  network_offering =
"DefaultIsolatedNetworkOfferingWithSourceNatSer
vice"
  zone = "zone01"
}
```

CITCOT

What is Configuration management?

- Configuration management is a process for maintaining computer systems, servers, and software in a desired, consistent state, It's a way to make sure that a system performs as it's expected to as changes are made overtime.

Ansible

- Ansible is a deployment and configuration management tool similar in intent to chef and puppet.
- It allows (usually) devops teams to orchestrate the deployment and configuration of their environments without having to re-write custom scripts to make changes.
- Controller/Master is the central configuration server where we will have our all configurations stored.
- Managed/Client nodes are all clients getting configuration from ansible master node
- Ansible has a text-based inventory file that have a list of individual servers and/or group of multiple servers.
- Ansible playbooks are a structured way to put all of the defined tasks for your application or your whole setup
- Asnible have a set of task which are unit of work that represents a single procedure to be performed.

ANSIBLE

Ansible Example

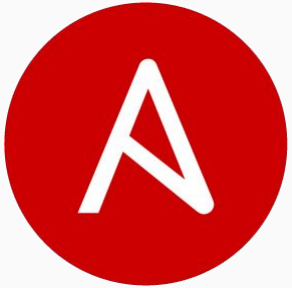
Install_apache.yml

```
---
- name: Install Apache2 on webservers
  hosts: webservers
  become: yes

  tasks:
    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Ensure Apache2 is installed
      apt:
        name: apache2
        state: present

    - name: Ensure Apache2 is started and
enabled
      systemd:
        name: apache2
        state: started
        enabled: yes
```



hosts.ini

```
[webservers]
server-1
server-2
server-3
```

Edit_web.yml

```
---
- name: Install Apache2 on webservers
  hosts: webservers
  become: yes

    - name: Create custom index.html
      copy:
        dest: /var/www/html/index.html
        content: |
          <html>
            <head>
              <title>Welcome to {{
ansible_hostname }}</title>
            </head>
            <body>
              <h1>Welcome to {{
ansible_hostname }}</h1>
              <p>This is the default web page
of server: {{ ansible_hostname }}</p>
            </body>
          </html>
```

CITCOT

Thank You

- +2519770355111
- info@citcot.com
- citcot.com