Bookmark Links:

- Free Trial and Free Tier | Google Cloud
- Regions and zones | Compute Engine Documentation | Google Cloud
- Global Locations - Regions & Zones | Google Cloud
- Using resource hierarchy for access control | Cloud IAM Documentation
- Resource hierarchy | Resource Manager Documentation | Google Cloud

Introduction

What Is Cloud ?

- The "Cloud" refers to servers, storage, and other computing resources that are available over the internet.
- Instead of storing data or running programs on your own computer, you access them through the internet.
- It's essentially a computer , but that is someone else. Its is a computer that is not owned by us (customers)
- When I say computer, we are referring to a group of machines.

What is Cloud Computing

https://www.youtube.com/watch?v=XZmGGAbHqa0&t=213s

Accessing physical resources over the internet at any time and from any device from anywhere.

Cloud computing is the delivery of computing services like Servers, Storage, databases, networking tools software over the internet.

Cloud computing enables companies to consume a compute resource, such as a server, storage or an application, **as a utility like water or electricity, rather** than having to build and maintain computing infrastructures in house.

- **First**, you get computing resources **on-demand and self-service**.
    - All you have to do is use a simple interface and you get the processing power, storage, and network you need, with no need for human intervention.
- **Second**, you access these resources over the net from anywhere you want.
- **Third**, the provider of those resources has a big pool of them and allocates them to customers out of that pool. That allows the provider to get economies of scale by buying in bulk and pass the savings on to the customers. Customers don't have to know or care about the exact physical location of those resources.
- **Fourth**, the resources are elastic. If you need more resources you can get more, rapidly. If you need less, you can scale back.
- And **last**, the customers pay only for what they use or reserve as they go. If they stop using resources, they stop paying. That's it. That's the definition of cloud.
    - GCP billing is for seconds.

What is GCP ?

- Google Cloud Platform is a suite of cloud computing services offered by Google.
- Google Cloud Platform is a suite of cloud computing services offered by Google. It provides infrastructure, tools, and services for building, deploying, and managing applications and workloads in the cloud.

Why GCP ??

- Trust and Security
- Open CLoud Platform
- Global Network:
    - Largest SDN (Software Defined Network) across globe. Worlds 40% traffic will flow in this traffic,
    - Same network where Google uses, network uses.
    - No internal n/w in Other clouds, (Mumbai to Us-Central)
- AI Driven

Why Choose GCP ??

- **Highly Secure, Reliable, and Scalable Infrastructure**:
    - GCP allows developers to build, test, and deploy applications on the same infrastructure that powers Google's services, ensuring top-notch security, reliability, and scalability.
    - **Comprehensive Developer Tools**:

It offers a wide range of tools and services for application development, including APIs, managed databases, DevOps tools, and serverless options.

Types Of Clouds:

- **Public Cloud:**
  - Multi Tenant Implementation
  - Owned and operated by provider
  - Pros:
    - No upfront capital expenses
    - No Maintenance
    - High Reliability
    - Easy scalability
  - Cons:
    - Less customizable
    - Governance issues
- **Private Cloud:**
    - Single Tenant Implementation
  - Owned and operated by our own companies
  - Cons:
    - Higher upfront cost expenses
    - Risk under resource utilisation
    - Higher ongoing costs
    - More mainitance
  - Pros:
    - Fully customizable
    - Higher level of security

        - Better performance
- **Hybrid Cloud:**
    - Combination of public and private cloud
    - Pros:
        - Greater Flexibility
        - Can sustain in outages
        - Can be manageable
    - Cons:
        - Higher upfront capital expenses
        - More maintenance
        - Under utilisation of resources

Free Tier Creation:

- Sign in with your gmail account
    - Console.cloud.google.com
    - You will be getting the below message after successfully login
        - Start your free trial with $300 in credit. Don't worry – you won't be charged if you run out of credit. Learn more
- Make sure you have steps already in place before creating af= free tier account. Until you have these steps , dont try to create a free tier account
    - Make sure the card that u enter(credit/debit card) should be having **INTERNATIONAL TRANSACTIONS ENABLED**
    - You card should have **RECUERSIVE PAYMENT ENABLED**
- After 90 days, still if you want to use GCP,
    - Create a new gmail account and follow the same steps
- Errors:
    - This action couldn't be completed. Try again later. [OR-BSBBF-103]

- An unexpected error has occurred. Please try again later. [OR-3DSRFT-08]
- Your request failed. Use a different payment method, or contact us. Learn more [OR-CCSEH-21]

- Free Trial and Free Tier | Google Cloud

Interact with GCP :

Five ways to interact with Google cloud Platform:

1. GCP Console.
2. Command line interface (CLI) > CLoud Shell and Cloud SDK
3. Client Libraries > GO, Java, NodeJs, Python, Ruby , Php, C#, C++
4. Google Cloud Mobile App. > Appstore/Playstore
5. REST API's > Postman, curl

Infrastructure :

https://cloud.google.com/compute/docs/regions-zones
https://cloud.google.com/about/locations

Regions:

- Regions are independent geographical areas that host GCP data centres

- Regions are collections of zones
- At this moment 39 regions are been available worldwide and growing
- Typically a region consists of 3 or more zones

Zone:

- A zone is an isolated location within a region.
- Fully qualified name for a zone is **<region>-<zone>**
- On an average we have typically three or more zones and a given region.
- These zones are multiple individual data centre buildings in that same geographical region.

Google Cloud Supports :

- Zonal Resource
- Regional Resource
- Mutli Regional Resources
- Dual Regional Resources (especially in GCS)

Resource Hierarchy:

https://cloud.google.com/resource-manager/docs/cloud-platform-resource-hierarchy

- The Resource Hierarchy is almost similar to the linux file system.
- Each child object has only one parent
- You may find it easiest to understand the GCP resource hierarchy from the bottom up.
- All the resources you use--whether they're virtual machines, Cloud Storage buckets, tables in BigQuery, or anything else in GCP--are organised into projects.
- Optionally, these projects may be organised into folders; folders can contain other folders.
- All the folders and projects used by your organization can be brought together under an organization node.
- Projects, folders, and organization nodes are all places where policies can be defined. Some GCP resources let you put policies on individual resources too, like Cloud Storage buckets.
- Permissions are inherited from top to down

Organization Resource:

- Represents an organization (e.g., a company) and is the root node in the GCP resource hierarchy.
- IAM access control policies applied to the Organization resource are applied throughout the entire below hierarchy (through Folders, Projects, and Resources)
- Can grant access to different people in organization
- Not applicable to personal (e.g., Gmail) accounts

Folders:

- Additional (optional) grouping and isolation boundary between projects
- Collection of projects and other folders
- IAM roles applied to folder apply to all projects inside

- Not applicable to personal (e.g., Gmail) accounts
- Folders are optional

Projects:

- Each GCP project has a name and project ID you assign
- The project ID is a permanent, unchangeable identifier, and it has to be unique across GCP.
- You'll use **project IDs** in several contexts to tell GCP which project you want to work with.
- On the other hand, **project names** are for your convenience, and you can change them
- Required to use any GCP resources
- Basis for creating, enabling, using, and paying for GCP service
- Identifiers
    - Project ID (must be globally unique)
    - Project number (automatically generated)
    - Project name ("Friendly name")
    - Good practice to have identical Project name and ID
- The Cloud Resource Manager provides methods that you can use to programmatically manage your projects in Google Cloud Platform.
- With this API, you can do the following:
    - Get a list of all projects associated with an account.
    - Create new projects.
    - Update existing projects.
    - Delete projects.
    - Undelete, or recover, projects that you don't want to delete

Resources:

- Everything that is created and used on GCP

Google Cloud Resource Hierarchy

- Organization (root node)
- Folders (optional)
- Projects
- Resources (inside projects)

Project			
Project ID	Globally Unique(Across Google cloud)	Chosen by us	Can not Changed (Inmutable)
Project Name	Need not be Unique	Chosen by us	Can be Changed (Mu table)
Project Number	Globally Unique	Chosen by Google	Can not Changed (Inmutable)

Project ID we can create for the first time only.

Organization Account:

- Inorder to have a organisation , firstly we need to have a **domain name .**
- Go to godaddy or namecheap or any other domain name providers and purchase a domain.
- https://cloud.google.com/identity/docs/set-up-cloud-identity-admin
- https://workspace.google.com/gcpidentity/signup?sku=identitybasic
-

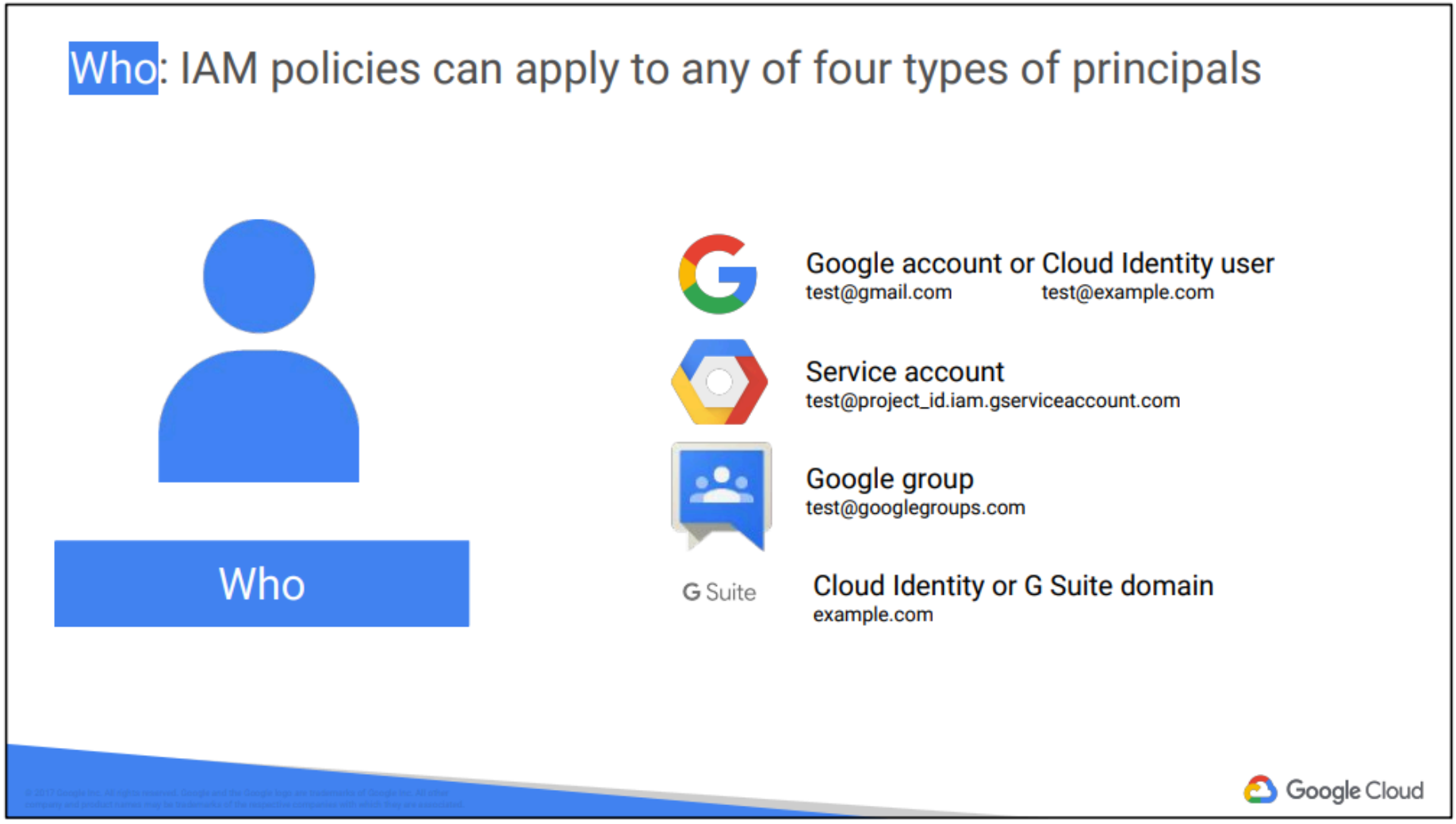
IAM:

- So what is Identity and Access Management (IAM)?
    - Technical definition
        i. - With Cloud IAM, you can grant granular access to specific GCP resources and prevent unwanted access to other resources. Cloud IAM lets you adopt the security principle of least privilege,so you grant only the necessary access to your resources.
        ii. Simple breakdown:
            - Who = Member
            - Can do what = Role
            - On which resource = All resources on GCP

- Who = Member = Identity
- Person or a service account
- Identity = email address

    - Person who logs in with email Ex: siva@siva.com
- People authenticate via google account
- Service account = application / server account
    - Not associated with person
- People accounts are created and managed outside GCP
- Service accounts are created

‘People’ account types:



Personal - devopswithcloud@gmail.com or can be non gmail account like siva@yahoo.com

G Suite/Cloud Identity domain -

- Managed Organisational accounts are outside GCP and managed in the Admin console. Ex: learnwithsiva.com
- Workspace:

    - Use Company wide Google apps (Docs, slides, maps ,etc)
- Cloud Identity:
    - No Access to G Suite applications (email, drive, etc)
    - can sync existing Active directory to cloud identity.
    - Note: GCP doesn't differentiate between G Suite or Cloud Identity. within GCP

Google Group:

collections of google accounts.

Apply single policy to collection of users in group

Special Account Types:

allAuthenticatedUsers

Special identifier to represent any google account

Anonymous users are not allowed.

allUsers: (public)

Another special identifier

Anyone and everyone

Both are not represented as an email account .

Roles:

- Roles: are collection of Permissions
- Permission: Determine what operations are allowed on resource
- Permission format: **<service>.<resource>.<verb>**

    - **Compute.instance.create**
- Permission are not directly assigned to member/identity(WHO)
- Permissions are bundled into Roles. Those roles are assigned to the User(WHO)
- Members can have multiple Roles as well.

Role Type: Primitive | Predefined | Custom

Primitive Role:

- Very broad rule existed before IAM came into picture. Applies to all resources in a project
    - **Owner**: Full control over all resources, including IAM and billing
    - **Editor**: Same as Owner, but without IAM and Billing
    - **Viewer**: View access only.

Predefined Role:

- More granular and specific to a certain service or resource
- Permission to a specific resource
- By default , there are 100's of roles available in GCP.
  Predefined(static) group of permissions.
- Compute instance admin (role) > he/she will be get the access or permissions to GCE only , but not o toher resources

Custom Role

- Mix and Match of the individual permissions.
- These are more granular, and i create what are exactly needed.
- I can a custom role, from an existing role and remove what are not required.
- Or, i can create a custom role from the scratch, and can include only permissions that are needed.

Resources:

Resource = Everything in GCP =What are we giving access to?

All of the components of GCP

- Compute Engine VMs

- Cloud Storage buckets

- Includes Organizations, Folders, Projects, Services, and all resources

Ex: compute Instance Admin = edit resources to GCE, but not for other resources

Custom Role:

Mix and Match individual permissions

Even more granular. choose only the specific permissions you need.

- Custom role to create instance:
  - Permissions to create Instance
    - Compute.disks.create
    - Compute.instances.create
    - compute.instances.setServiceAccount
    - Compute.subnetworks.use
    - compute.subnetworks.useExternalIP
    - **Oslogin**

Service Accounts:

- Why is it so important ?
  - Service account is a very special account that does not belong to humans, rather it belongs to an application or VM or resource in GCP

- Special Type of Member:
    - Not attached to any user
    - Don't need end user authentication like Humans.
        - Service accounts will not login from console like humans do, why?
            - Because they dont have userid and passwords
        - But , still SAl needs authentication
            - We will create keys for SA
    - Identity represented by email address
        - ex: projectnumer-compute@developer.gserviceaccounts.com
    - By Default, GCE Instances use service accounts for GCP access
- There are two Types of service accounts:
    - User managed
    - Google Managed

Types Of Service Account:

- User Managed:
    - Created by user and google (automatically)
    - User managed accounts access via IAM/Scopes
    - Google created Service account:
        - ex: PROJECT_NUMBER-compute@developer.gserviceaccount.com

User Created:

SERVICE_ACCOUNT_NAME@PROJECT_ID.iam.gserviceaccount.com

Google Managed:

- Created and Managed by Google - IAM Roles are automatically controlled by Google

ex: cloud service account - PROJECT_NUMBER@cloudservices@cloudservices.gserviceaccounts.com

These are created when Project is created, and runs internal google process on our behalf

Cloud Shell:

- https://cloud.google.com/sdk/docs/quickstart
- https://cloud.google.com/shell
- https://cloud.google.com/shell/docs/how-cloud-shell-works
- Web browser access
- No need for local terminal
- Have Editor by default
- Automatic ssh key management
- 5GB of persistent storage
- Web Preview showing in localhost
- Easy access to man pre installed tools and all are upto date
    - ex: gcloud, gsutil, kubectl, python , etc

VPC:

Network:

- Sharing the resources.
- But in n/w , we are having some specialised devices such as **Hubs, switches, routers, firewalls, lb** etc
- What is a server ?
    - Server is just providing a service or some functionalities to clients , in what we call as Client server Model.
- What is client ?

    - A client can be called as a piece of hardware or s/w that access a service that is made available by the server.
- The server is often an Other Computer which is serving some requests.
- How is client communicating to server ?
    - Ex: Google server will be serving requests ,by listening on a port number.
    - By default there will be different application serving on different port numbers.
    - Different port numbers will be having different protocols.
- What is a protocol ?
    - Set of rules used for communicating between devices

What is a Hub ?

- Hub is a multiport repeater ?
- What is a repeater?
    - It will be sending the signal from one port to other port.
    - It amplifies the signal but dont understand the actual signal.
- The purpose of Hub is , to connect all of our networking devices together into an internal network.
- Hub does not have any intelligence.
-

What is a Switch?

- A switch i very similar to hub.
- Switch will also have multiple ports, that accepts Ethernet connections from all the n/w devices that are connected in the networl.
- But ublike Hubs, a switch is intelligent .
- Hubs and switches are used to exchange the data within a local area npetwork.
- These are not used to exchange data outside the network .
- Inorder to achieve this, ata should be exchange outside own n/w,

- We can implement this,, we are able to read IP address.

What is a Router ?

- Now we are having something called as router, which will use ip address to route from one network to other network.
- These routers will be at Layer 3 devices.
- Router, will send / forward the data from one network to other network based on the ip address, we mentond,
-

- VPC = Software defined Network
- If we want to implement a **private network** on Google cloud for your specific project, we implement using VPC
- Central foundation for all networks in google cloud
- So What is a SDN ?????
- Traditional Network:
  - Multiple hardware components(Routers, servers, switches, load balancers, etc)
  - Removes maintenance and over head
  - Rapidly can be scalable
- RFC- 1918 - Private networking and IP addressing standards.
- By default, when we are creating a brand new project, Google will be creating **default** vpc.
- Projects:
  - Hold one or more VPC's
  - By default we can have 5 vpc's created but , we can raise an exception if we want any.
  - projects seperate users, where as vpc seperate networks

GCP Subnet Modes:

- VPC N/Ws consists of one or more IP range partitions called Subnets.
- Default:
    - Created by google for every new project
- Two types of modes:
    - Auto, custom mode
- Auto Mode Network:
    - automatically google will be creating subnet for all regions
    - One subnet for every region
        - subnet range will be 10.x.x.x/20
        - Creates quickly and ready to use
    - Why use auto mode ??
        - Easy to setup and use
        - Predefined ip address does not overlap with each other
    - Why not use auto mode ??
        - Not as flexible as custom mode
        - don't need subnet for all regions
        - often not suitable for production environments
        - Connecting 2 vpcs, can cause issue - overlapping subnets
    - Custom Mode Network:
            - No subnets automatically created
            - Much more flexible
        - VPC mode conversions:
            - one way only
            - can convert auto mode to custom mode, but not vice versa

IMP Pointer:

- In VPC, we can add multiple subnets
- When we are creating a subnets, make sure we consider 2 areas
    - Name of the subnet
        - We need to be making sure, the name of the subnet will not be the same with other subnet in the same region under the same project.
        - But we can have the same name, from a different regions subnet even though they are in the same project,
    - Range of the subnet
- We cant delete the subnets created with auto subnet mode.
- We can delete the subnets which we have created even though they are in the auto vpc.
-

Protocol:

- Set of rules, used for communicating between devices.

-

Firewall in GCP:

Before we go into firewalls, lets try to understand how a networking works:

- Computers need to speak a common language to communicate
- TCP/IP network protocol suite is the most common language used.
- TCP/Ip is not a single protocol , but it is a suite or collection of protocols.

What are network Ports?

- Protocols higher than level 4 of the OSI model have a port number assigned.
- Port = channel
- 65535 ports available for both TCP and UDP protocols.
- Examples:
    - HTTP: 80
    - HTTPS: 443
    - RDP: 3389
    - MYSQL: 3306
- Not every protocol has a port number :
    - Below level 4 wont have a port number
    - ICMP
- Default VPC's have firewalls created and enabled. Custom VPC's - No firewalls are present by default.
- Allow/deny traffic to and from instances
- Manage both inbound (ingress) and outbound (egress) traffic
- Defined at network (VPC) level, but enforced for each instance
- Firewalls defined at VPC level applies to all resources inside VPC network
- Firewall enforced at instance level is done by Network Tags.
- **Billing Note:**

- Only Physical resources (VMs, DBs, Clusters etc..) are billable/chargeable
- Firewalls, VPC, Subnets, IAM Roles are not billable/chargeable in GCP.

Pre-populated rules in the default network

The default network is pre-populated with firewall rules that allow incoming connections to instances. These rules can be deleted or modified as necessary:

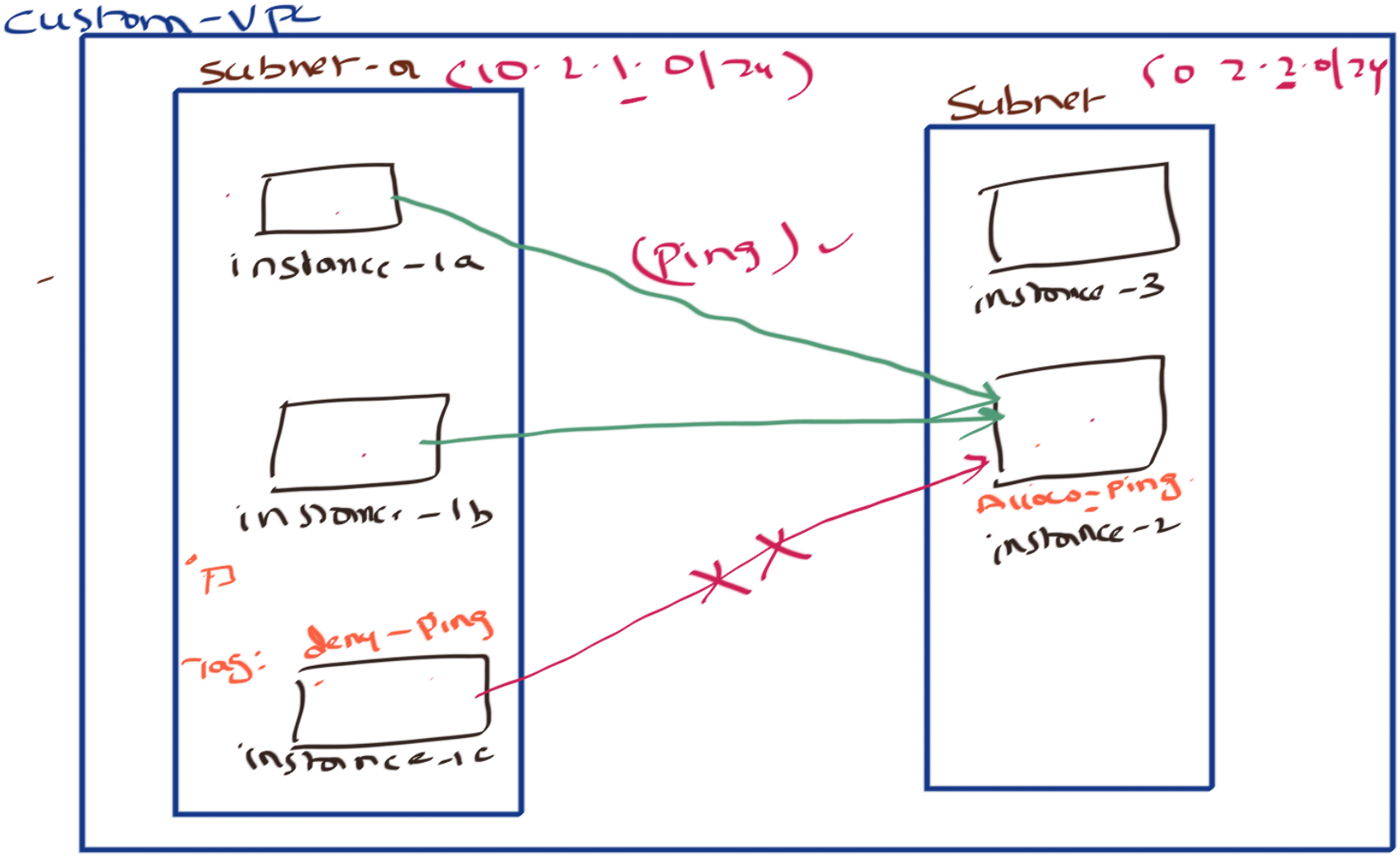
Rule name	Direction	Priority	Source ranges	Action	Protocols and ports	Description

`default-allow-internal`	`ingress`	`65534`	`10.128.0.0/9`	`allow`	`tcp:0-65535`
					`udp:0-65535`
					`icmp`
						Permits incoming connections to VM instances from other instances within the same VPC network.

`default-allow-ssh`	`ingress`	`65534`	`0.0.0.0/0`	`allow`	`tcp:22`	Lets you connect to intances with tools such as `ssh`, `scp`, or `sftp`.
`default-allow-rdp`	`ingress`	`65534`	`0.0.0.0/0`	`allow`	`tcp:3389`	Lets you connect to instances using the Microsoft Remote Desktop Protocol (RDP).
`default-allow-icmp`	`ingress`	`65534`	`0.0.0.0/0`	`allow`	`icmp`	Lets you use tools such as `ping`.

Network Tags:

- Instance-level, granular enforcement
- Apply tag to instance
- Rule is enforced on only tagged instances and not entire network



VPC Peering:

What is VPC Peering ?

Till now we have mainly dealt with a single VPC , but in organisations there will be cases where we need to connect two different vpc, and establish a communication between them.

Those two vpc can be in the same organisation or they may be in different organisations also.

The only requirement is that they are in GCP (not in any other cloud provider)

Traffic stays in google private network and does not touch public internet and still has RFC 1918 .

But Why not VPN:

- Lower latency in VPC peering compared to VPN
- Security:
    - All network traffic will be inside google network and does not touch the public internet
- Cost vpn: egress traffic charges even in same zone
    - Peering = no egress charges even in same zone
- Simplicity:
    - A lot easier to set up

Restrictions:

- CIDR range on one network should not overlap with other network
- GCP vpc only
- Cannot restrict which subnets are shared / routed. Everything connects to everything.
    - We can use firewall rules to restrict traffic

- Network tag/ service account filter won't be applied to VPC peering
- Firewalls rules are not exchanged between vpcs.
- No transitive peering is supported in VPC Peering

Shared VPC:

- We will be creating a project specific for the network , where all the firewall rules routes and all are created. This has been managed by a specific team.
- Typically used in large organisations where there is a dedicated networking team.
- Without a network we can't provision a vm.
- So now we will consume the network from this project. This way we are removing the overhead of managing the network by yourself in our specific gcp project.
- Shared VPC shares a VPC across multiple projects in the same organization.
- Terminology we need to understand:
    - Host-Project: Project hosting the shared VPC
    - Service Project:Projects with permission to shared VPC
    - Standalone Project: project not using any shared VPC
- Shared VPC IAM Roles:
    - Creating an Shared VPC required organization or folder level roles
    - Sometime, we might refer shared vpc as XPN (cross project network)
    - Organization Administrator:
        - Assign **Shared VPC Admin** Roles
    - Shared VPC ADMIN:

        - Assigned at org or folder level
        - Enables shared vpc for host project
        - Attaches service projects
        - Assigns access to subnets shared by shared VPC.
    - Service Project Admin:
        - Owner/Editor/Compute Instance Admin/**Network User** of service project
        - Assignment of Network User role in host Project -allows Service project users access to Hose Project network subnets
        - Can assign per shared subnet
- Why separate projects to begin ?
    - Why not place everything in the same project.
    - Separation of projects for access control and billing , but still need access to same VPC resources
    - Projects are the primary method of separating access and billing.
- Consideration of using Shared VPC:
    - Only within the Single **GCP Organization.**
    - **Service project can only link to single host Project**
    - Project cannot be both host and service project
    - Static IP address associated with (and billed to) project that reserved it.
- Resources that use Shared VPC:
    - Compute Engine Instances
    - Compute Engine Instances templates
    - Compute Engine Instances Groups
    - GKE Clusters
    - Internal IP Addresses
    - Internal DNS
    - Cloud DNS Private Zones
    - Load Balancing

Note:
Host-project-1
Host-project-2

Service-Project-A → host-project1, host-project-2, (this does not work)

Private Google Access:

Private Service Access:

Identity-Aware Proxy (IAP) | Google Cloud.

To allow IAP to connect to your VM instances, create a firewall rule that: applies to all VM instances that you want to be accessible by using IAP. allows ingress traffic from the IP range **35.235.240.0/20** for port 22 or you can open for all TCP .

Compute Scenario:

- We have been working as a Cloud engineer in i27Academy.
- Krish is been hired as a Cloud Engineer
- Its a ecommerce **application**, developed on the below technologies
    - Multiple microservices , been developed in Java
    - They have the frontend developed in Angular.
    - They are using Mysql as a database to store all the data.
- As a cloud engineer I have been asked to deploy these applications, so that customers will be able to access my application.

- So we have chosen **Google cloud as a strategic partner**, so from now any application that our company develops should deploy on GCP only .
- What all applications should I deploy in the cloud ?????
    - We are having a webserver(apache) > deploy in us-central1-a
    - We want to deploy java based applications (microservices)
    - We need to deploy frontend with Angular components
    - Our data should be in Mysql databases
    - I should be having some load balancer configured.
    - DNS routing should also be taken care,
    - There should be a mechanism for autoscaling my application based on the traffic.
    - For security I should be implementing all the https certificates for sure .
- How to deploy these apps in GCP ?
- Google is providing multiple options for hosting your applications in GCP:

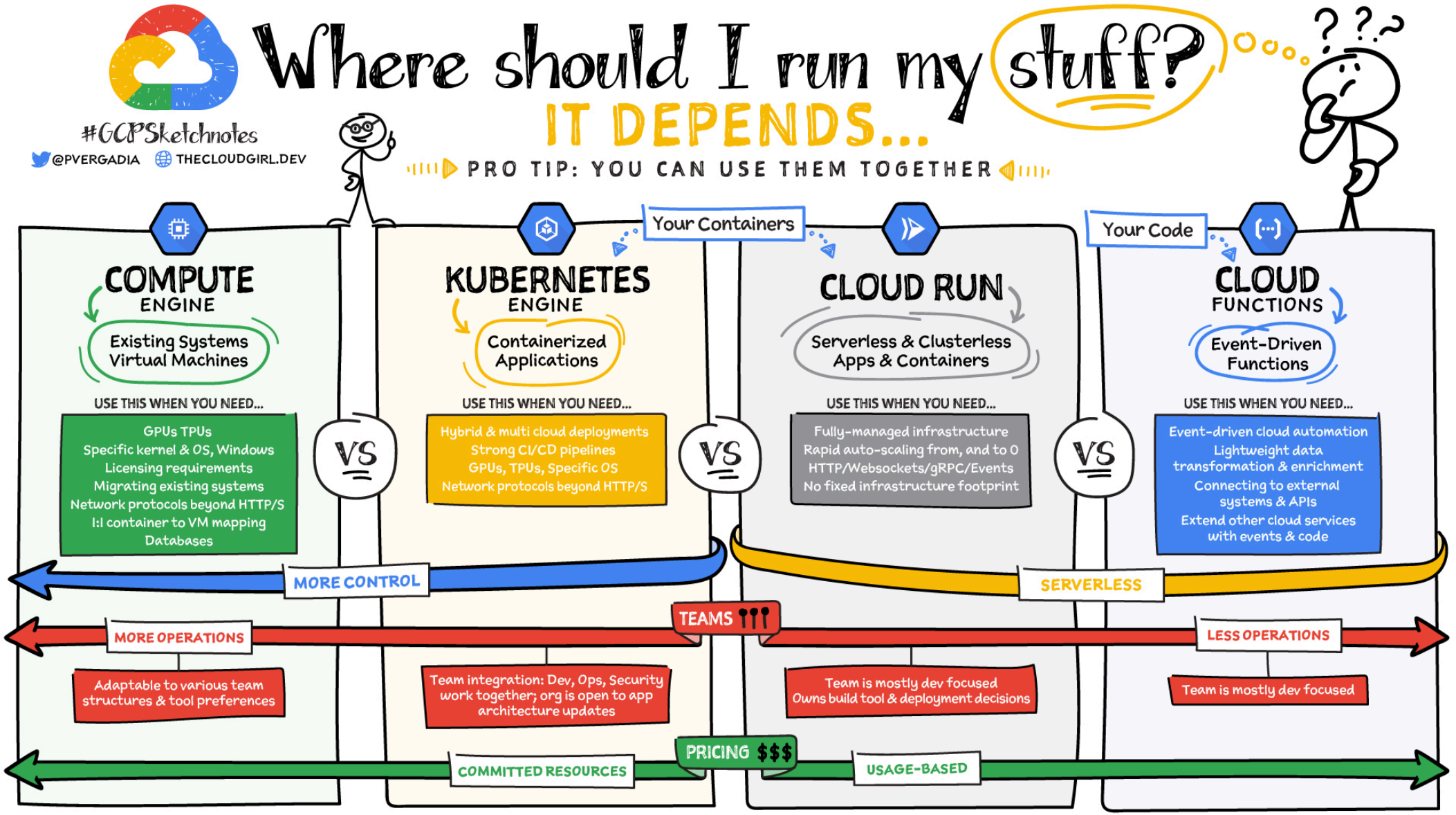
Compute Compute Options Comparisons:

Helpful URL: https://cloud.google.com/blog/products/compute/choosing-the-right-compute-option-in-gcp-a-decision-tree

There are multiple options available for hosting our applications:

- Google Compute Engine
- Google Kubernetes Engine
- Google App Engine
- Google Cloud Run

- Google cloud functions



Compute Options Overview – Compute Engine (Virtual Machines, Disks, Network)

- Infrastructure as a Service (IaaS)
- Virtual machines – referred to as 'instances'
- EC2 in AWS and Azure Virtual Machine in Azure
- Virtual version of physical PC
    - CPU/GPU
    - Memory
    - Disk space
    - OS
    - Firewall controls
    - Network connection/management
- Offers complete control and most flexibility, but also comes with most operational overhead.
- Great fit for organisations moving existing servers into the cloud and are used to managing VM's

Container Engine(GKE)

- Deploy containerized applications
    - De-couples application components from OS
    - Run app in multiple environments, regardless of OS
- Powered by Kubernetes – open source container orchestration system

- **“Manage applications, not machines”**

Google App Engine:

- Platform as a Service (PaaS)
- Developers can focus on writing code, while Google handles the rest
- Build scalable web applications and mobile backends
- Managed service
    - Never touch the underlying infrastructure
    - Deployment, maintenance, and scalability handled
    - Reduces operational overhead

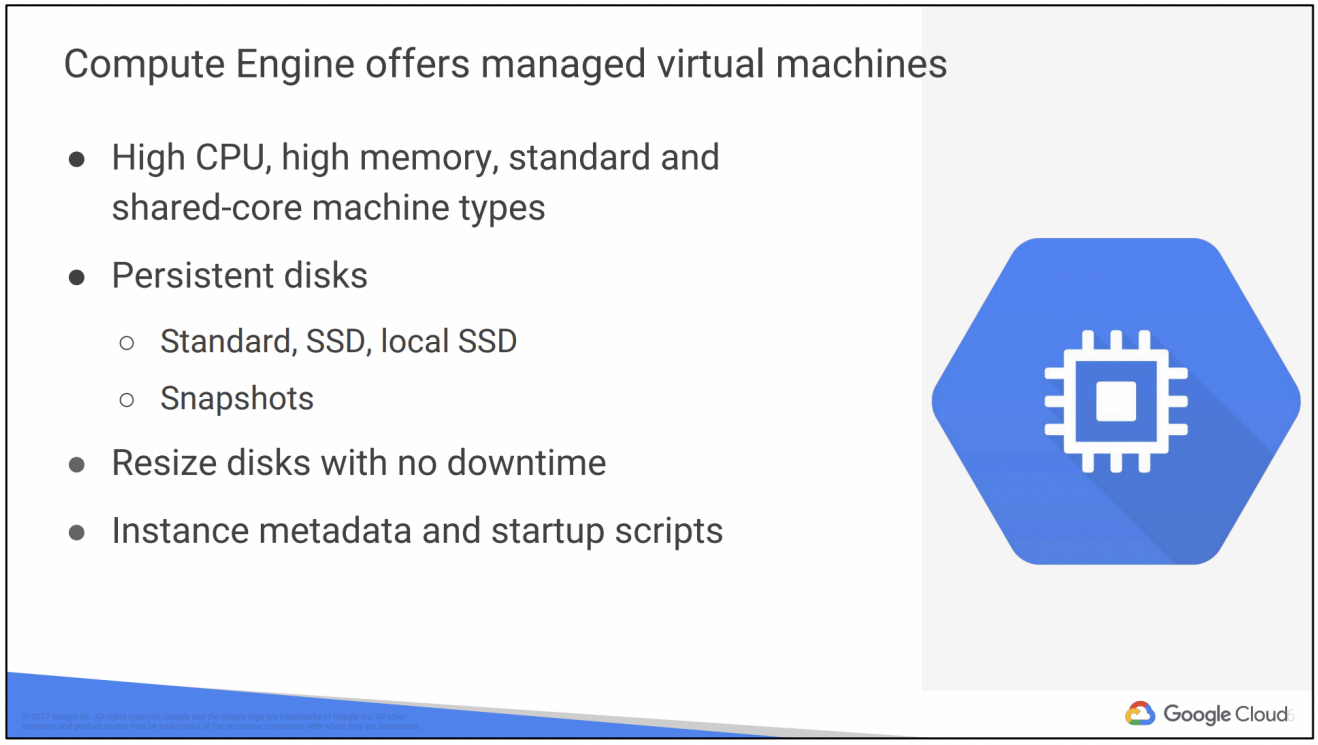
Google Cloud Run:

- Mainly used for container based applications(https)
- If you have a image and you want to deploy, cloud run is the best solution

Google Cloud Functions:

- Serverless environment for building and connecting cloud services
- Example: A file is uploaded to Cloud Storage (event), function executes in response to event (trigger)

Compute Engine:



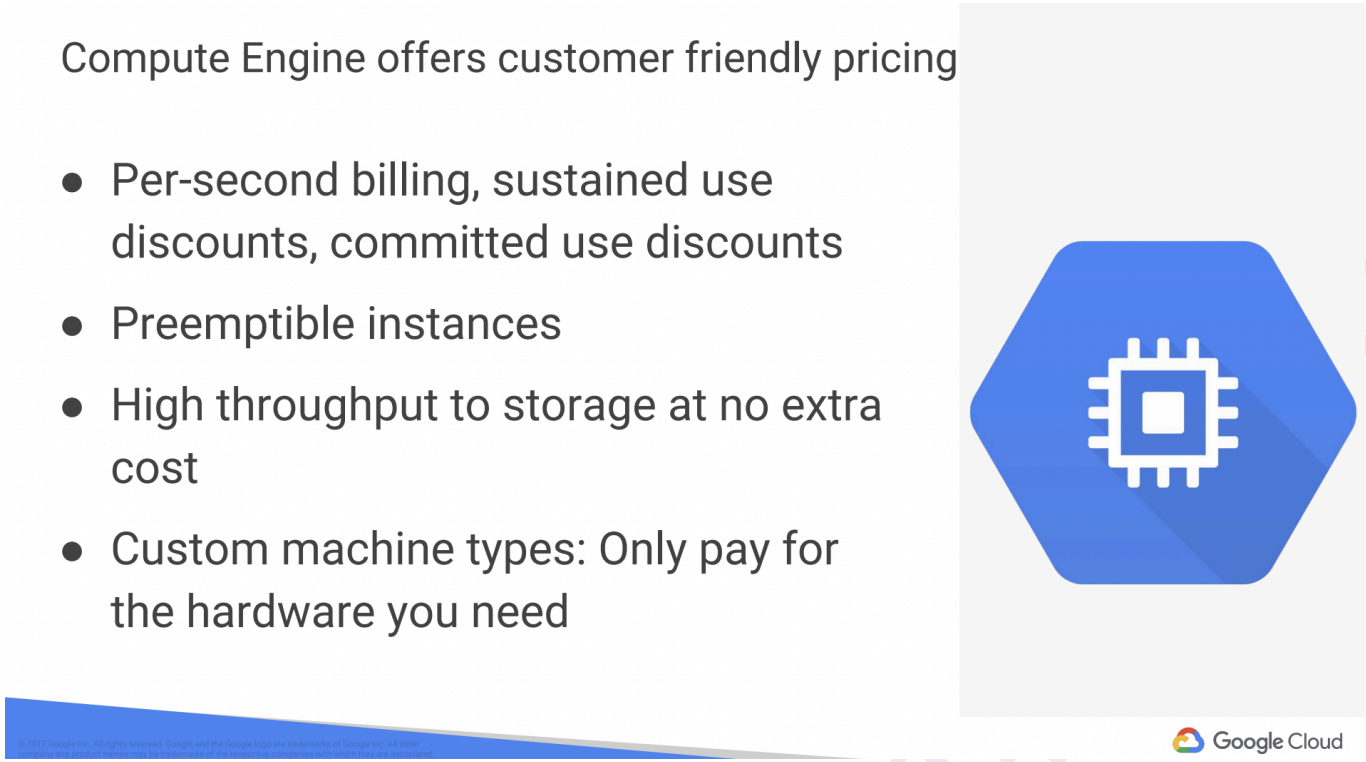
What is a Compute Engine ?

- Compute engine is a customizable compute service that lets you create and run virtual machines on Google's infrastructure.
- You can create a Virtual Machine(VM) that fits your needs based on the requirement.

Why are we focusing more on GCE?

Virtual machines have the power and generality of a full-fledged operating system in each. You configure a virtual machine much like you build out a physical server: by specifying its amounts of CPU power and memory, its amounts and types of storage, and its operating system. Compute Engine lets you create and run virtual machines on Google infrastructure. There are no upfront investments, and you can run thousands of virtual CPUs on a system that is designed to be fast and to offer consistent performance. You can flexibly reconfigure Compute Engine virtual machines. And a VM running on Google's cloud has unmatched worldwide network connectivity. You can create a virtual machine instance by using the Google Cloud Platform Console or the gcloud command-line tool. A Compute Engine instance can run Linux and Windows Server images provided by Google or any customised versions of these images. You can also build and run images of other operating systems.

- VM's are the heart and soul of GCP
- GCE | GKE | GAE all run on VM's(GCE)
- Let's start with Single instance, then we shall discuss about automation, auto scaling, load balancer, etc



Compute Engine bills by the second for use of virtual machines, with a one-minute minimum. And discounts apply automatically to virtual machines that run for substantial fractions of a month. For each VM that you run for more than 25% of a month, Compute Engine automatically gives you a discount for every incremental minute. You can get up to a 30% net discount for VMs that run the entire month.

Machines Types:

In compute engine, **machine types** are grouped and curated by **families** for different workloads, you can choose from General-purpose , memory-optimised, compute-optimised, and accelerator optimised family

Note:

- If you are creating a vm from cli and havnet mentioned the machine_type, google will defaultly take n1-standard-1

MachineType	Purpose
General Purpose (E2)	Day-to-day computing at lower cost Use case: ● General servers ● Websites ● Databases ● Developer environments, virtual desktops, ● Microservices deployment
Standard (N1,N2,N2D)	Balanced price/performance across a wide range of VM shapes. Web serving, App serving, Back-office applications & so on
Memory-Optimised (M1, M2)	Ultra high-memory workloads. Useful in Large in-memory databases like SAP HANA, In-memory analytics
Compute-Optimised (C2)	Ultra high performance for compute-intensive workloads. Useful for HPC, Electronic Design Automation (EDA) & Gaming,N2
Shared-core	f1-micro(1 vCPU with 0.6 GB Memory) , g1-small (1 vCPU with 1.7 GB Memory. Cloud Shell will be of type f1-micro

N1 - Intel Sandy Bridge, Ivy Bridge, Haswell, Broadwell, and Skylake CPU platforms.
N2 - Intel Cascade Lake CPU platforms
N2D - AMD EPYC Rome Processor

GCE Notes:

- H/W is Chargeable (Memory and CPU are charged only when VM is running)
- Disks are Chargeable (Will be charged even VM running or stopped)
- IAAS Solutions: If VMs' boot disk fills up, need to manually increase size.
- PAAS Solutions: If size fills up, GC will automatically increase it.
- Scopes: By default when Compute Engine API is enabled, it will create a SVC account.
- VMs are created and can be modified though it depends on dev/prod to determine if a VM needs to be stopped.
- GC provides default firewall options for http/https access.
- In order to create a VM we need a Network to be in place (Default or Custom networks)
- Private / Public IP are ephemeral means not permanent. VM restart doesn't guarantee the same Public IP.
- If a VM is public facing and needs public ip to be the same then need to request for a Static IP.
- Reserve Static IP and use – No Charge. If Reserved and Not Used – High Charge.
- Public/external ips are used mainly for Load Balancers only. Implying it will be touched by the Internet.
- Disks – Can attach N number of disks. Can be modified run time or downtime.
- Shielded VMs implies secure and protected VMs.

Custom Machine Types:

Lets say if the predefined machine types do not match our needs, we can independently specify the number of vcpus and the amount of memory to pur instance.
Example:
If a client asks us to create an instance with 6vcpu's and 17.34 gb ram, then we can create a custom instance.
But the ram will be of 0.25 increment, and cpus will be of even numbers only.

Discounts:

Google will be having 2 types of discounts.

- Sustained Discount :
    - If we are using a particular virtual machine continuously for a month , google will automatically give a discount of **30% .**
    - This discount has been calculated on a pro rata basis.
- Committed Discount
    - Compute Engine offers the ability to purchase committed use contracts in return for deeply discounted prices for VM usage.
    - These discounts are known as **committed use discounts.**
    - If your workload is stable and predictable, you can purchase a specific amount of vCPUs and memory for up to a (50-55%) 57% discount off of normal prices in return for committing to a usage term of 1 year or 3 years.
- Discounts on Preemptible vms(spot instances):
    - These VM's are shortlived VM's , and are mainly used for batch processing/ job processing .
    - So there are possibilities that, google might provide around **80%** discount .
- Google offers customer friendly pricing:
    - Google is the first major cloud provider to deliver **per-second billing ,** for is Infrastructures as a service compute offering (GCE) or google kubernetes engine.

Reservations:

Preemptible VM's or Spot instances :

Short-lived, low-cost VM

24 hours max

Can be shut down at any time (30-second warning)

Disposable?? not for critical single VM?s

Ideal for fault-tolerant, batch processing workloads:

Rendering

Big data analytics

Hadoop and big data

Most often used in managed

Fixed pricing, up to 80% off regular instance price

computing(cpu/memory) is cheaper. storage and licensing will be the same cost.

https://cloud.google.com/blog/products/gcp/220000-cores-and-counting-mit-math-professor-breaks-record-for-largest-ever-compute-engine-job

-
- Google Sends out a 30sec notification before terminating the pre vm. Its a log message that gets generated at the VM level.
- Only Cpu and memory will be taken back by google , but our disk will be in our account only.
- Cant convert regular vm to preemptible vm and vice versa.
-

SSH-Keys:

- Each linux instance requires a private and public key to access .
- Match private key on our machine + public key on server

Everything's been set in Meta data section in VM Instances:

- Its a programmatic access to instance info + custom values
- Set by Key:Value pair

Two methods of connection:

- Google managed SSH Keys(both console, gcloud commands)
    - Process of creating these keys is automatic

    - Maintianing/managing those keys is automatic, all these are been handled by Google
    - Gcloud compute instance ssh <instance_name>
- Custom SSH-key:
    - It a complete Manual process
    - We need to create ssh keys ,
        - We will get both public and private keys
    - Add public ssh key info to metadata
    - Then you can Ssh into the GCE VM using private key

Creating custom ssh-keys:

- Create a key on local machine
    - ssh-keygen -t rsa -f ~/.ssh/ssh-key -C admin
    - Restrict access to private key
        - chmod 400 ~/.ssh/ssh-key

Locate SSH key:

- Windows:
    - Public key file: ~/.ssh/[KEY_FILENAME].pub
    - Private key file: ~/.ssh/[KEY_FILENAME].ppk

Add public ssh key to meta data (project or instance)

Connect to instance with ssh key:

ssh -i <path_to_key> <username>@<public_ip_address>

To Create a User in Ubuntu follow the below steps:

$ `adduser username`

Add the new user to the sudo group

`usermod -aG sudo username`

```
Switch to newly created user:
su - username
```

How to Enable SSH Password Authentication

To enable SSH password authentication, you must SSH in as root to edit this file:

```
/etc/ssh/sshd_config

PasswordAuthentication yes

.ssh/authorized_keys
sudo service ssh restart
```

Disks:

- If we want to work with storage on Google compute engine, we are having 3 main options:
    - Block Storage
        - Persistent Disk (PD)
        - Local SSD
    - File Storage
        - Filestore
    - Object Storage
        - Google Cloud Storage (GCS) buckets

- Now PD is the only boot disk option available for Google Compute Engine which means that 100% of any compute engine instances that we use should be having boot disk as PD only. That p can be again ssd or hdd.
- Now, what that unique available in pd ?
- Disks are of 2 types:
    - Boot disks
    - Extra disks
-

Disk Scenario:

- Let's assume, Siva is working for an organisation as a cloud engineer.
- As per the requirement from devs, he should be creating a web server/application server to make sure there are applications deployed.
- At the same time, he should be creating a kubernetes cluster for the application team.
- We are in the process of creating a GCE in us-central1-a with the following configurations.
    - Zone : us-central1-a
    - Machine-type: e2-medium
    - Network: default n/w with same subnet
    - Public ip: yes
    - Boot disk: Ubuntu
    - Boot disk size: 10 gb
- After we created the vm, we got one more requirement from the devs stating that , siva, we want to make sure our microservices logs are getting stored in the VMs and our data from the elastic search should also be stored.
    - Now we have 2 options,

        - One is ask the devs to store the logs in the same boot disk that we created,
        - Second is we can add **one more disk to the GCE VM for logs storage**
        - We can add one more disk to the elastic search data.,
        - By adding individual disks, we can make sure they're development will be easy, and there will not be any conflicts with the data.
    - Action Item:
        - After our initial analysis, we want to create 2 pds.

We're going to talk all about disks and let’s see how to store data and manipulate data on Google Cloud platform within the computer engine.

So, when we work with storage on a computer engine instance, we have three main options that we want to talk about and the majority of our discussions is a persistent disc as it is the most common default option.

Persistent disk Is not directly attached to any instance in the google cloud. In other words, it's actually a network attached storage.

And it comes in the **standard mechanical** and also these **solid-state drives** as well.

Another One is we have a **local SSD** that actually is directly attached to your virtual machine in the same manner as a traditional physical disk that you may be used to working with.

And we also have **cloud storage buckets,**

Let's start with the most important one, which is a persistent disc.

Now persistent disc is the only boot disk option available on Compute engine, which means 100% of any compute engine instance that you use, the operating system is booting from a persistent disk whether it is a standard mechanical version or a solid state drive variety as well.

Now what's very unique about persistent and disk is that it's not just a single physical disk directly attached to your computer like you'd be used to when working with a physical desktop or laptop, or maybe even a physical server.

Rather, a persistent disc is actually a series or an array of multiple disc **within the same zone that are attached over a network connection.**

You can preserve data on a disc after deleting an instance because they're not directly attached. You can delete an instance but keep a disk that was attached to that instance, and then later decide to reuse that disk in another instance or may be converted into a custom images.

There's a lot you can do with the process. In dis, you can resize it. You can move it. You can attach an additional disc.

Yes, you could actually increase the size of your disk. While the instances are running and have the operating system take advantage of that increased space again, when it was running. You simply can't do that with a physical disk

Since we can easily resize or add new disc, you simply don't need to have to worry about partitioning and existing hard drive. You can just simply either add more space or at an additional disk if we need another drive letter or another mounted disc and a Lennox instance as well. A few

A local SSD is most similar to a physical disk because it is indeed a physical disc with a set amount of size that is physically attached to your virtual machine. It's literally the exact same style of a single physical disk that you're probably used to working, too.

A local SSD **cannot** be a boot device. All it can do is be an attached disc, your boot device or your root disk must always be a persistent disc, and also be aware that the only time you can set up a local SSD is when you create the instance the first time. This means that if you create an instance and later decide that I want to have a local SSD attached to it, you can't do that. And this because again the SSD is physically attached to the instance. You could only set up. Your local SSD is when you set up the instanceindistance at the same time,

Also, because a local SSD is directly physically attached to your instance, once you delete your instance, your local SSD is the lead as well effect with some of the more updated user interface. When setting up a local SSD, Google Cloud will directly refer to it as a local SSD scratch space, which means you should not be storing any data on your local SSD.

Types of Disk in Compute Engine:

- Network Attached Storage – Standard PD, SSD PD, Balanced PD .
- Machine Attached Storage – Local SSD

Handson-Scenario:

- Lets **create a disk** in us-central1-a with size **30 gb.**
- Lets create a VM, in us-central1-a with a boot **disk of ubuntu with 10gb ,** and attach the disk that we created in the previous step.
- Later, after few months we came to know that our **initial boot disk** is not sufficient and we want to increase the **boot disk**
    - What are the various ways ?
        - We can increase the book disk size from **10gb to 20gb.**
        - But unfortunately, the increased disk size is not populating in my vm , when I execute **df -h** command.
        - We have 2 options to fix this issues
            - First Option:
                - Restart the VM instance using "**sudo systemctl reboot**" and the increased disk size will appear.
                - But let's assume this is our prod server, and we are in the middle of the day where our business is running,

        - Restarting a machine in the business hours is not recommended.
        - For that restart, we need to take multiple approval, change tickets, and then only we can take it forward.
    - Now again increase the boot disk size from **20Gb to 40Gb**
    - Second option:
        - We will grow and then resize the instance
        - Sudo apt install cloud-guest-utils -y # Install the packages
        - Sudo growpart /dev/sda 1 # Resize / grow my partition
        - Sudo resize2fs /dev/sda1 # extend the file system to use the added space
- But we also realised that the new disk that we created for the developer to store the **logs** is not getting populated.
- And as developers asked us to have a mount for the logs to store, below are the steps to achieve that
    - We can format and mount the non boot disk
    - Refer this documentation,
    - Sudo lsblk
    - sudo mkfs.ext4 -m 0 -E lazy_itable_init=0,lazy_journal_init=0,discard /dev/sdb
    - Sudo mkdir -p /logs/
    - sudo mount -o discard,defaults /dev/sdb /logs
    - If some time, the data is not able to write in the newly created folder, execute the following command
    - Chmod a+w /logs
    - Once this is done, u can inform the dev to use **/logs**
    - But agin, we restarted the machine, due to some issue and observed that, the extra mount we created disappeared, again we need to fix the issue
        - ??????????

https://cloud.google.com/compute/docs/disks/add-persistent-disk

Topics we will be discussing in this disk section:

1. How to access disk management menu
2. Create disks
3. Disk types
4. Interacting with instances
5. Attaching disks to instances
6. Resizing existing disks

Commands used in our exercise on disks:

Images:

Scenario:

- We got a requirement from the management regarding an ecommerce application we are working on .
- For now as we are working on Virtual machines, we want our applications to be deployed on Web Server, an application server.
- May be after sometime we might be moving towards containers/serverless.
- But for now we want to deploy our application on IAAS (GCE)
    - And these servers should be having some sort of packages already installed\.

    - When developers want to create a machine, for now they are able to use the public images, and they are spending some time in installing all the packages required for the application to run .
    - Ex: Once a dev creates a vm, inorder for this java app to run, he/she should be installing all the required packages:
        - JAVA
        - MAVEN
        - Websever
        - Apache tomcat server
        - GIT
        - Python 3.x installed,
        - Company packages, which should be by default be available when a person creates a vm in pur project.
        -

- Images are Necessary to create boot disks for GCE instances
- Unlike disks, images not limited to zone - much more accessible across other platforms
- Public and Custom images
- Public - provided and maintained by Google/community/vendors
    - All projects have access
- Image family simplifies image versioning
- Its a best practise, to stop the VM while creating an Image.
- **Deprecating Images::**
    - Deprecated: Still works but gives warning
    - Obsolete: new users can't use it, error if attempt to use, existing links still work.
    - Deleted: all users cannot use it

    - Active: mark deprecated image as active again(command line only)

Sharing and Moving Images:

- Share across projects
- Requires **compute engine image user** role to host project
    - Ex: If user in project A wants to access image in Project B , then user in project a must have compute engine image user role granted for project b
    - Role grants access to all projects in project
- For managed instances groups, Project A service account must be granted to project B

Export image to Cloud storage:

- Ideal for sharing with projects without host project access
- Export image as a tar.gz file to cloud storage
- Linux only, not available for windows
- Sharing with image user role is preferrable

Snapshots:

So a snapshot is a primary backup method in which compute engine takes a snapshot

In any sort of a backup or disaster recovery situation, you could simply take that **snapshot** of either an attached disc or a boot disk and simply create a new instance or a new disk from that snapshot backup.

What are snapshots ?

- Simple put : instance/disk backups
- Periodic backups via a point in time snapshot of disk
    - Incremental backup

- Can create snaphost while instance is running
- Can create snapshot of boot disk or attached disks
- Purpose - periodic incremental backup of existing disk/instance

Snapshots Best Practise:

- Pause applications/processes that write data.
- Take only one snapshot at a time
- Schedule snapshots during off-peak hours
- Use multiple persistent disks for large data volume

Realtime-Scenarios:

- **Requirement**: Client wants to increase Machine type from f1-micro to g1-small
  - Solution: Stop the instance, edit the machine type and start it again
- **Requirement:** Client wants to have the instance in us-central1-a instead of us-central1-c
  - **Solution1:** We can use below command to move from one zone to other
    - Gcloud compute instances move instancename –zone=us-central1-a –destination-zone=us-central1-c
    - During the process, a snapshot will be created by google, and using that snapshot a new vm will be created in the destination zone mentioned in the above command .
    - Note: This will only applicable of the boot disk is non-shielded
  - **Solution2:** Create a manual snapshot, create a instance in the destination zone
- **Requirement:** Client wants to have the instance in **asia-south1** instead of **us-central1**

- **Solution:** Use snapshot option

- **Requirement:** Client wants to move the instance from Dev project to prod project
  - **Solution:** Create a custom Image
- **Requirement: Client wants to share the image to sister company , who uses GCP**
  - **Solution:** Grant compute image user role(as they are in separate org)
    - gcloud compute instances create jdkins --zone us-central1-a --image=webser2-jdk11 --image-project=concrete-rig-333201
- **Requirement: Client wants to share the image to their sister company, who does not use GCP**.
  - **Solution:** Export image to Cloud storage.
  - Only requirement : the sharing image should be **Linux**

Snapshot	Images
Available within the same project	Across diff projects
Can be created while the instance is running	Can't be created while running, must stop the instance
Good for backup, disaster recovery. Snapshots are incremental	Good for instance/instance templates creation

Startup-scripts:

- Mainly used for easy management/creation of large number of VM's
- Automated software installation, updates etc during the boot proces.
- Mainly used to create Instance Groups using Instance Templates
- Two ways to pass the startup script:
    - Direct Input:
        - Paste the script in the field.
    - Link the script to GCS
        - Using metadata server URL
        - Must have access to bucket
            - startup-script-url

```
#! /bin/bash

sudo apt update -y

sudo apt install apache2 git -y

sudo systemctl enable apache2

sudo git clone https://github.com/devopswithcloud/static-website.git

sudo rm -rf /var/www/html/index.html

sudo cp -r static-website/* /var/www/html/
```

#/var/log/daemon.log

Shutdown-scripts:

- Used when we are using preemptible VMs.
- Shutdown scripts are especially useful for VMs in a managed instance group with an auto-scaler.
- Scripts are used before VMs are deleted or if autoscaler shuts down the VM in groups.
- Shutdown script runs before VM stops & shuts down any running processes, clean up tasks, export logs or sync with other systems.

Instance-Groups:

- Instead of working on a single VM, working with many instances at once.
- Automatically scaling the capacity based on the needs
- There won't be any need for overloading of servers.
- Our application can be deployed/available at a global space.
- There are 2 types of Instance-Groups:
    - Managed :
        - These are often paired with a Loadbalancer Backend.
        - Can have the capacity to automatically auto-scale based on the needs of the traffic.
        - We can deploy new versions of apps, without down time.
        -
    - Unmanaged:

    - Unmanaged instance groups are ideal for migrating existing configurations for load balancing tasks with minimal modifications. (Different types of m/c or diff h/w configured VMs are grouped together - Unmanaged Instance Group)

Notes:

- **Scaling Out/Scaling Up**: Adding more VMs to efficiently handle increases in traffic dynamically.
    - Adding More VM, when there is more load
- **Scaling In/Scaling Down**: VMs to efficiently handle reduction in load.
    - THis refers to deleting of VM's. when the load of the VM's is lowered.
- **Horizontal Scaling**: Increasing the number of VMs needed to address the demand.
- **Vertical Scaling**: Increasing the h/w specification of the VM server.

- TIll now we are dealing with a single VM, but now as the times we will deal with group of VM's
- This group of vms, is considered a s Force Multipliers
- BY doing this , i am basically autoscaling my application based on the need, rather than adding vms , on the fly.
- TIll now, all my application load is been handled by a single server,
    - But now the load will be distributed to the group of vm's
    - No more overloading of our servers.
- My application should be accessible globally.

Load Balancers:

- It distributes or balances the user network requests towards the backend pool of instance.
- From now onwards, we will be having a single point of access, we will be having multiple backend targets which will actually serve my requests.
- Is this load balancer in GCP, a software defined or a hardware defined ?
    - CLoud load balancers are Software Defined LB's
- Are the load balancers , Global or Regional or internal or external ?
    - IT basically depends on the Load balancer we select.
    - By default there are many load balancers types which will support pour requests .
- What are the various of Load Balancers available in GCP :
    - Load balancers are divided into few types:
        - Global External LB
        - Regional External LB
        - Regional Internal LB
    - Global Load Balancers
        - HTTP(S) Load balancer
        - SSL Proxy
        - TCP Proxy
    - Regional Load Balancer
        - Network Load Balancer
    - Regional Internal Load balancer
        - Internal TCP/UDP

        - Internal HTTP(s) Load Balancer
- Every load balancer can be either a Global or regional load balancer
- Global Load Balancer are called as Proxed Load Balancers
    - Proxy Load Balancers:
        - These will terminate the incoming traffic at the load balancer level, and then creates a new connection to connect to the VM's
        - These are available at the Global Load Balancers
        - These will also support ipv6
    - Non-Proxied or Pass through Load Balancers:
        - The traffic goes through the load balancers and it will directly reach the virtual machine , instead of establishing a new connection at the load balancer.
        - All regional load balancers are pass through
        - Does not support ipv6
- Load balancers can be regional and internal
    - When we say regional load balancers, the load balancer is going to be in a specific region.
    - And the requests does not come outside the world, they only be distributing from within the GCE Vm's
- Regional Load Balancers are called Pass through load balancers.

Storage Comparisons:

- Google Cloud storage:
  - If we want to deal with any unstructured data.
- Google Cloud Bigtable:
  - If we are having some petabytes-scale no-sql database
  - If we want to have some High-throughput and scalability
  - Key/value data which is wider
- Google Cloud BigQuery:
  - If we are having Petabyte-scale analytics for data warehousing
  - Fast SQL queries across larger datasets
  - Ai, BI this bigquery is the foundation
- Google Spanner:
  - Global SQL_based Relational database
  - Horizontal scaling and high availability
- Cloud SQl:
  - Managed Mysql, PostgeSQL, Sqlserver database
  - Vertical scaling
- MemoryStore:
  - Managed Redis instance
  - In memory db, cache or message broker
  - Vertical scaling
- Firestore:
  - Fully Managed NoSQl document database.
  - Realtime database for mobile sdks.

Google Cloud Storage(GCS):

- Google Cloud storage is the unstructured data service on google cloud that is arranged within the buckets and it contains objects.
- Google Cloud Storage offers developers and IT organisations durable and highly available object storage.
- It assesses no minimum fee; you pay only for what you use.
- Prior provisioning of capacity isn't necessary.
- Cloud Storage is the unstructured data service on Google Cloud that is arranged within buckets and objects inside of those buckets.
- What's object storage?
    - It's not the same as file storage, in which you manage your data as a hierarchy of folders. It's not the same as block storage, in which your operating system manages your data as chunks of disk. Instead, object storage means this: you say to your storage, "Here, keep this arbitrary sequence of bytes,," and the storage lets you address it with a unique key. In Google Cloud Storage and in other systems, these unique keys are in the form of URLs, which means object storage interacts well with web technologies.
- **What can a storage bucket hold?**
    - The answer is simply everything that exists in digital format that includes pictures, videos, files, scripts, even entire database is if you need to transfer or back of those databases from one location to another
- GCS is Immutable BLOB(binary large object) storage
    - I.e., for example if we upload a file, u can't edit it rather replace it .
    - If you want to edit the file we need to use BLOCK storage
- Google Cloud Storage is not a file system, although it can be accessed as one via third-party tools such as Cloud Storage FUSE. The storage objects offered by Google Cloud Storage are "immutable," which means that you do not edit them in place, but instead create a new version.

- Google Cloud Storage's primary use is whenever binary large-object storage is needed: online content, backup and archiving, storage of intermediate results in processing workflows, and more
- **A bucket has virtually infinite size**. You could go to the petabytes range and beyond. Of course, assuming that your budget allows for that, you don't pay for a pre allocated amount like a typical consumer cost storage service.Rather, you only pay for the amount of data currently in that bucket, and it is elastic, so you do not need to pre allocate any amount of space in advance.
    - **GCP has 5tb size per object , s3 has 5TB, oracle has 10Tb**
- GCS > Amazon CLoudFront + Amazon S3 Transfer Acceleration + Amazon Cross Region Replication + Amazon Glacier + Amazon S3
- Offline Media Import/Export is a third-party solution that allows you to load data into Google Cloud Storage by sending your physical media, such as hard disk drives (HDDs), tapes, and USB flash drives, to a third-party service provider who uploads data on your behalf.

	Standard				
Class	**MultiRegional(3 Regions)**	**Regional**	**Nearline**	**Coldline**	**Archive**
What data can be placed here	Frequently used data across regions	Frequently used data, within the region.	Access a file, once in a month	Access a file, once in quarter	Once in a year
Duralibility	99.999999999'S	99.999999999'S	99.999999999'S	99.999999999'S	99.999999999'S
Availability	99.95%	99.90%	99%	99%	No SLA
Storage Cost	0.026$ / GB /Month	0.02$ / GB /Month	0.01$ / GB /Month	0.007$ / GB /Month	0.004$ / GB /Month
Retrieval Cost	Free	Free	Applicable	Applicable	Applicable
Minimum Duration	NA	NA	30 Days	90 Days	365 Days

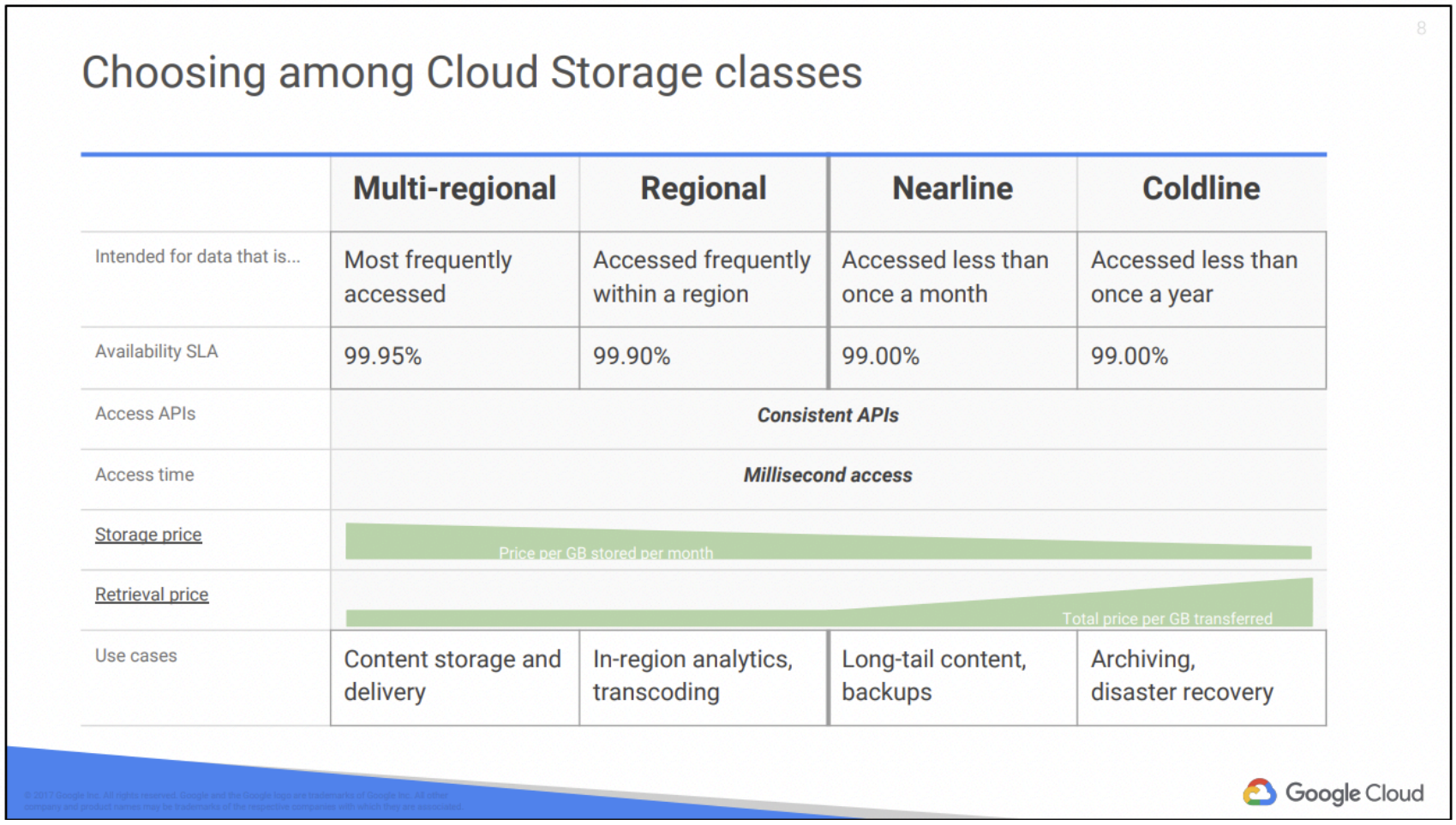
- **Changing Storage Classes:**
    - Nearline/Coldline can be regional or multi regional
    - Cannot cahnve from MR to regional
    - Chaging object , only newly created objects will be affected
- Media Import/Export is helpful if you're limited to a slow, unreliable, or expensive internet connection.
- Offline import is available through third-party providers: https://cloud.google.com/storage/docs/offline-media-import-export
- Cloud Storage Transfer Service enables you to import large amounts of online data into Google Cloud Storage quickly and cost-effectively. To use Cloud Storage Transfer Service, you set up a transfer from a data source to data sink.
- Data sources can be an Amazon Simple Storage Service (Amazon S3) bucket, an HTTP/HTTPS location, or another Google Cloud Storage bucket. Data sinks are always a Google Cloud Storage bucket. Example uses of Cloud Storage Transfer Service include: ● Backing up data to a Google Cloud Storage bucket from other storage providers.
- Moving data from a Standard Storage bucket to a Nearline Storage bucket to lower your storage costs.
- Google Cloud Storage always encrypts your data on the server side, before it is written to disk, at no additional charge. Data traveling between a customer's device and Google is encrypted by default using HTTPS/TLS (Transport Layer Security). In fact, Google was the first major cloud provider to enable HTTPS/TLS by default.
- Data is encrypted automatically at rest and in transit by default
    - GMEK
    - CMEK
    - CSEK (Customer supplied Encryption key) > Secure one
- Basically, you stick data in, and it automatically holds it. So regarding the structure, the primary unit in cloud storage is a bucket, and a project can have one or more bucket assigned to it

Your Cloud Storage files are organized into buckets

Bucket attributes	Bucket contents
Globally unique name	Files (in a flat namespace)
Storage class	
Location (region or multi-region)	
IAM policies or Access Control Lists	Access Control Lists
Object versioning setting	
Object lifecycle management rules	

Google Cloud

- Your Cloud Storage files are organized into buckets. When you create a bucket: you give it a **globally-unique name**; you specify a geographic location where the bucket and its contents are stored; and you choose a default storage class.
- Pick a location that minimizes latency for your users. For example, if most of your users are in Europe, you probably want to pick a European location: a GCP region in Europe, or else the EU multi-region.
- If you need finer control, you can create access control lists ("ACLs") that offer finer control, ACLs define who has access to your buckets and objects, as well as what level of access they have. Each ACL consists of two pieces of information: A scope, which defines who can perform the specified actions (for example, a specific user or group of users). And a permission, which defines what actions can be performed (for example, read or write).



Cloud Storage lets you choose among four different types of storage classes:

1. Regional
2. Multi-regional
3. Nearline
4. Coldline

Multi-regional and Regional are high-performance object storage, whereas Nearline and Coldline are backup and archival storage. All of the storage classes are accessed in analogous ways using the Cloud Storage API, and they all offer millisecond access times.

Regional Storage:
lets you store your data in a specific GCP region, us-central1, europe-west1 or asia-east1. It's cheaper than multi-regional storage, but it offers less redundancy.

Multi-Regional Storage:
Costs a bit more, but it's geo-redundant. That means you pick a broad geographical location, like the United States, the European Union, or Asia, and Cloud Storage stores your data in at least two geographic locations separated by at least 160 kilometers.

Multi-Regional Storage is appropriate for storing frequently accessed data: website content, interactive workloads, or data that's part of mobile and gaming applications. People use regional storage, on the other hand, to store data close to their Compute Engine virtual machines or their Kubernetes Engine clusters. That gives better performance for data-intensive computations.

Nearline storage :
It is a low-cost, highly durable storage service for storing infrequently accessed data. This storage class is a better choice than Multi-Regional Storage or Regional Storage in scenarios where you plan to read or modify your data on average once a month or less. For example, if you want to continuously add files to Cloud Storage and plan to access those files once a month for analysis, Nearline Storage is a great choice.

Coldline Storage:
It is a very-low-cost, highly durable storage service for data archiving, online backup, and disaster recovery. Coldline Storage is the best choice for data that you plan to access at most once a year, due to its slightly lower availability, 90-day minimum storage duration, costs for data access, and higher per-operation costs. For example, if you want to archive data or have access in the event of a disaster recovery event.

The availability of these storage classes varies, with multi-regional having the highest availability of 99.95%, followed by regional with 99.9% and nearline and coldline with 99.0%.

As for pricing, all storage classes incur a cost per gigabyte of data stored per month, with multi-regional having the highest storage price and coldline the lowest storage price. Egress and data transfer charges may also apply.

In addition to those charges, Nearline storage also incurs an access fee per gigabyte of data read, and Coldline storage incurs a higher fee per gigabyte of data read.

Gsutil:

- Command line format for working with Cloud Storage buckets/objects
- Different format compared to gcloud commands
- Useful for:
    - Creating and deleting buckets
    - Uploading, downloading, and deleting objects
    - Listing buckets and objects
    - Moving, copying, and renaming objects
    - Editing object and bucket ACLs
- Accessing buckets
    - Bucket syntax = **gs: / / <BUCKET_NAME>/ <OBJECT_NAME>**
- gsutil Command Structure
    - gsutil <command> <options> <target >
    - gsutil <command> <options> <source> <target >
    - Very similar to Linux commands
        - cp, mv, rm, ls; same for both Linux and gsut i l

- Almost identical structure
- Example: `gsutil cp flle1.txxt gs://mybucket`
- gsutil ls - a gs://mybucket
- Top-level command options (before the command)

Signed URL's:

A Signed URL is yet another access control mechanism provided with cloud storage to control access to your cloud storage objects.

So what exactly is signed URLs ?
Sign URL's give limited time access to anyone who is having the URL of your object.

gstuil signurl [options] [key_location] [object url]
The URL is time-bond , and valid for **MAXIMUM of 7 days**
Ex: gssutil signurl -d 10m key.json gs://<name>/filename

If someone got issues with pyopenssl use this command from cloud shell

pip3 install --upgrade setuptools pip
pip install pyopenssl
pip3 install --user pyopenssl or pip install pyopenssl

Object Versioning :

What exactly is object versioning to start, so by default when working with a cloud storage,If you delete an object, or if you overwrite an object with another version of that same object name, that object is deleted

Now there are no recycle bins or deleted items folder within Google Cloud Storage. Once you delete something from Google cloud storage it is gone, and you cannot recover it.

So if there is a way, you can't accidentally delete or overwrite something that you need, even if you do so if there is an ability to grab those files then that would be great right, so now object versioning comes into picture.

So versioning works on GCS by allowing you to retrieve objects that are deleted or overwritten and is applied at the entire bucket level.

By default object versioning will be turned off, you need to enable versioning. And versioning can be applied at a bucket leve, rather than object level.

So when you create a brand new storage bucket, object versioning is off, and anything you delete is indeed deleted.

So when the object version is turned on if you delete an object or override an object in the cloud storage bucket that object is archived instead of getting deleted and the object still exists.
If you look at the Web console, it appears to be gone.But if you look via the command line,it will be still there.

Life cycle management:

So what life cycle management does is that it sets a set time to live or tt el for short on an object. So what happened is that when a series of conditions is met, lifecycle management will either automatically archives or delete an older version of your object
Or you also have the ability to instead of deleting it, downgrade the storage class for certain objects, from a standard storage bugger to either a near line or cold line storage class.
My cycle management is like versioning also applied at the entire bucket level. It's not object by object.

Cloud Storage also offers lifecycle management policies. For example, you could tell Cloud Storage to delete objects older than 365 days, or to delete objects created before January 1, 2013; or to keep only the 3 most recent versions of each object in a bucket that has versioning enabled.

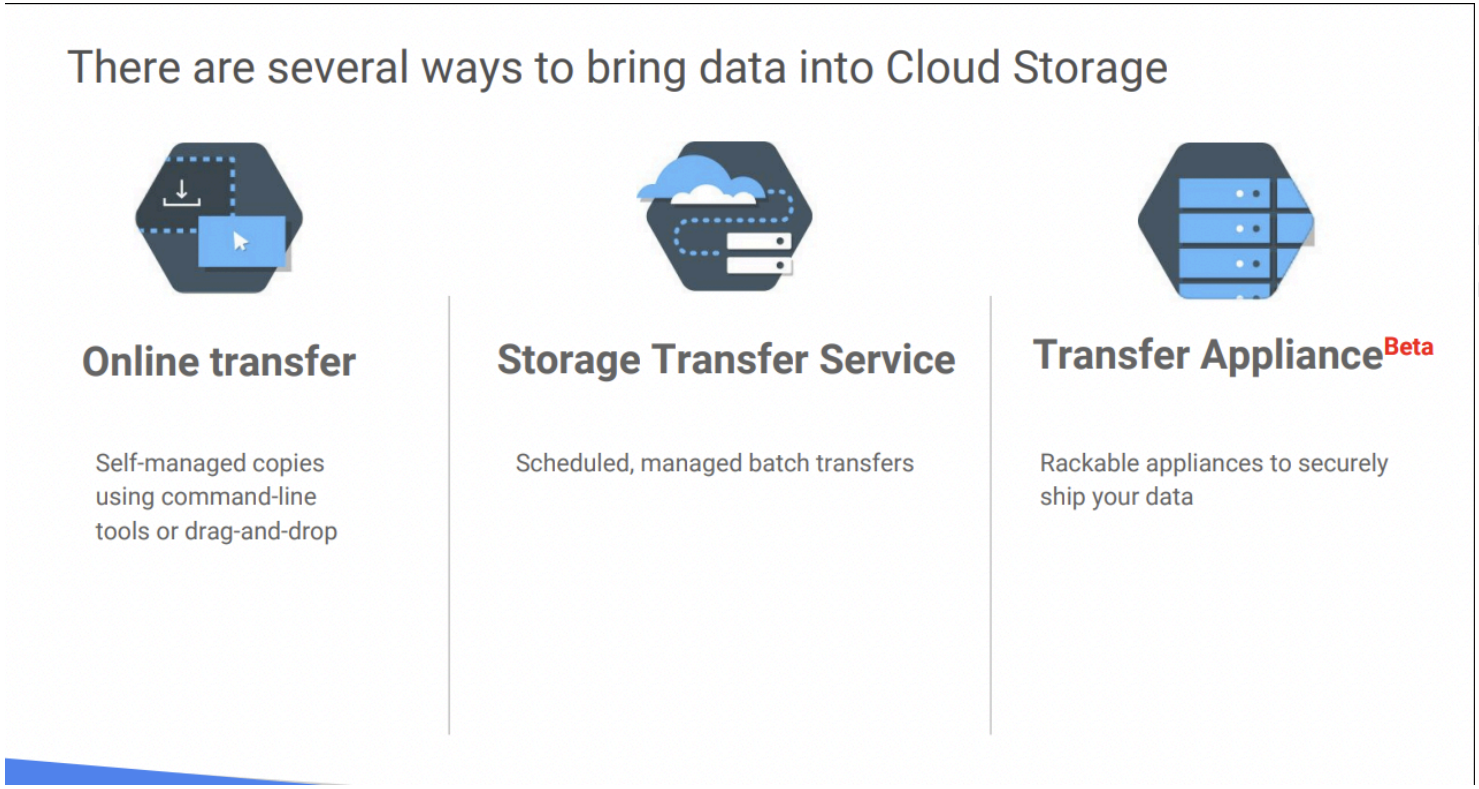
Changing storage Classes:

- Nearline/Coldline can be regional or multi regional
- Cannot Change from Multi Regional to Regional (Vice Versa)
- Changing classes only affect newly created objects

We can change the Multiregional (standard) to nearline. It affects newly created objects only. Older objects remain the same.
In UI we don't have the option to do, so we need to make it in the Command Line only

Transfer Appliances:

Regardless of which storage class you choose, there are several ways to bring data into Cloud Storage.



Many customers simply use gsutil, which is the Cloud Storage command from the Cloud SDK. You can also move data in with a drag and drop in the GCP Console, if you use the Google Chrome browser. But what if you have to upload terabytes or even petabytes of data? Google Cloud Platform offers the online Storage Transfer Service and the offline Transfer Appliance to help.

The Storage Transfer Service lets you schedule and manage batch transfers to Cloud Storage from another cloud provider, from a different Cloud Storage region, or from an HTTP(S) endpoint.

The Transfer Appliance is a rackable, high-capacity storage server that you lease from Google Cloud. You simply connect it to your network, load it with data, and then ship it to an upload facility where the data is uploaded to Cloud Storage. The service enables you to securely transfer up to a petabyte of data on a single appliance.

There are other ways of getting your data into Cloud Storage, as this storage option is tightly integrated with many of the Google Cloud Platform products and services. For example, you can import and export tables from and to BigQuery, as well as Cloud SQL.

You can also store App Engine logs, Cloud Datastore backups, and objects used by App Engine applications like images. Cloud Storage can also store instance startup scripts, Compute Engine images, and objects used by Compute Engine applications. In short, Cloud Storage is often the ingestion point for data being moved into the cloud, and is frequently the long-term storage location for data.

Transfer rehydrater appliance (Using Marketplace) will be helping us to get the data been moved to GCS< when using any appliances. The below links can help to get more understanding on these

https://cloud.google.com/architecture/migration-to-google-cloud-transferring-your-large-datasets
https://cloud.google.com/blog/products/gcp/bust-a-move-with-transfer-appliance-now-generally-available-in-us

Data Encryption:

- Encryption is a process, that takes the legit data (plaintext) as input and transform them into illegible data(Ciphertext)
- It will make the date unreadable, which will protect our data.
- We will be using algorithm , Google Cloud AES.
- Keys are used to encrypt and decrypt the data
    - These keys are called, Public and Private keys
- We can implement encryption using 2 ways:
    - Symmetric
    - Asymmetric

- Symmetric: We will be having a single key(Private key) , which will be used in encryption and decryption.
- Asymmetric: we will be having both Public and Private key. Public key is used for encryption. Private key is used for decryption.
- By default google is having 3 types of encryptions
    - Google Managed Encryption Keys (GMEK)
    - Customer Managed Encryption Keys(CMEK)
    - Customer supplied Encryption Keys (CSEK)
- Google Managed Encryption Keys:
    - Cloud storage always encrypts your data on the server side , before even writing to disk at no additional costs.
    - Cloud storage will manage these server-side encryption keys on our behalf.
    - GCS encrypts the data using AES-256 (Advanced Encryption standard)
- Customer Managed Encryption Keys(CMEK):
    - It will be having an additional layer on GMEK keys, as a customer i will be ,managing those keys to encrypt or decrypt the data using Key management service (KMS)
    - In CMEK, we will be creating Keyring, and that key ring will be having keyrings.
- Customer supplied Encryption Keys (CSEK):
    - It will be having an additional layer on GMEK keys , we can choose to provide our own keys (AES-256) ,encoded in standard Base64 format.
    - Google nor GCS, will not store your keys, we should store the keys outside Google cloud somewhere in onprim.

Serverless

CloudSql:

Scenario:

- We are working for an organisation where the data is getting stored in MYSQL.
- Till now we have been in on-prim and our applications are consuming the same database which we created in On prim data centre.
- Now in 2023, we are modernising our application and moving towards cloud, and we have chosen GCP as the primary cloud.
- So we need to migrate frontend applications and backend databases as well to Google Cloud.
- For now, we only thought to migrate DB first followed by spring applications.
- As an architect I need to understand what options GCP provides if I want to migrate the data.
- As of now the Database instance in my onprim is being built on IAAS solutions
- When moving to cloud, we confirmed that we shall not be using IAAS, rather we want to use PAAS solution.
- In GCP, we are having one of the PAAS solutions for databases called Cloud SQL.
- We want to implement CloudSQl as our relational DB, going forward.

CLoud SQL:

- CloudSQl is a **fully managed database service** that can be used to maintain, manage and administer your MySql, PostgreSQL, and SqlServers.
- CloudSQL helps the customers to focus on the application instead of the infrastructure . Google handles the infra part for us.
- CLoud SQL is a **PAAS solution.**
- It will automate your replica creations and back up implementations. It will provide 99.95% uptime.

- We need to just inform Google that, i need have a particular db, with this configuration. Rest infra creation, database setup/installations, backups restorations all are been taken care by Google.
- **CloudSQL instances are regional instances.**
- **They can span across multiple Zones in the same region(if high availability is turned on)**
- But they cant span across regions.
- CloudSQL instances can do,
    - vertical scaling, if you want the database to be having both Read and Write Operations
        - In Vertical scaling , we can increase the size of the VM
    - horizontal scaling , if we want only to read replicas.
        - In Horizontal scaling, we can create multiple instances
- CloudSQl instance allows the user to focus on the application rather than focusing on the underlying infrastructure.
- CloudSQL automated and replicated our backups by providing 99.95% availability.
    - Verify the uptime from this url https://uptime.is/
- CloudSQL can easily be accessible from the computer engine and app engine.
- Cloud SQl will be providing High Availability(it will have failover mechanism)
-

Cloud

Connect to the Instances:

- The user/application can connect to my DB using Public IP address.
- But inorder to connect, we need to authorize the public ip of the Client in my database.
- Configure private ip as well for Cloud SQL connections.

    - You can use private IP to have the services with your network connect to the Cloud Sql instances or they can use VPN connection to connect to the private ip from outside network.
    - Inorder to get privateIp, the cloud sql will be residing on its own VPC network(Google-hosted VPC IP). We cant put the cloudSQL instance on the vpc network that we created.
    - Benefits of going with Private IP:
        - Lower network latency compared to public
        - THis will have much more security compared to Public Ip connectivities
    - Now , how does it connect to other network ???????
- Till now, we are connecting to the CloudSQL instance, using Username/password.
- But , We can also implement Transport Layer Security (TLS?SSL) protocol when connecting to the cloudsql instances from the internet.
- When we are trying to connect with TLS/SSL , it wil ensure that the data that is transferred between the customer/applicaiton to the instance of the cloudsql is secure.
- CLoudSQL uses a self signed server certificate, for the instance and the it will generate public/private pair key for the cients.
- We can only download the respective client cert during the certificates createion.
- We can have upto 10 client certificates.
- However, appteams/applicaitons can still connect to the cloudsql instances, without ssl certificates as well.
- We need to ensure that, connections to the cloud sql instance will onl happen using SSL , for that we need to enable the checkbox to “allow ssl connections only”
- All the connections that we are making form Google App Engine, Google Compute engine are by default be encrypted.
-

PrivateIP Connection:

- We have enable public ip to the cloudsql during the instance creation .
- Now lets enable private ip as well, because from now on, we can ask out appthemes to connect with private ip.
- We have use Private service access.

Connection ways:

- To connect to cloud sql instance, we are having multiple ways
  - Mysql workbench > using public ip (We can connect using private ip as well, only if we have having VPN)
  - Using Google Compute Engine (Using PublicIP and privateIP)
    - If we are connecting with public ip, we should be whitelisting the publicip of vm in CloudSQL
  - Connect using Cloud shell (Using gcloud commands)
    - Using cloud automatic whitelist of public ip is happening from google side
  - Most important: I can connect using CLoud SQL Proxy (Much recommended approach)
    - The **Cloud SQL Auth proxy** is a Cloud SQL connector that provides secure access to your instances without a need for Authorized networks or for configuring SSL.
    - We will be connecting to the cloudsql from our local host once this proxy is cnfigured.
    - The Cloud SQL Auth proxy works by having a local client running in the local environment. Your application communicates with the Cloud SQL Auth proxy with the standard database protocol used by your database.

High Availability:

- By enabling High availability the cloudsql allows to create a standby instance with another zone in the sa,e region where your primary is available.

- Through syyncronous replication, any data that is written to the PD of the primary instance will also be written to the PD of the secondary instance.
- How HA Works:
    - If there is any **zonal failure** by enabling HA, we can reduce the downtime.
    - WHen the primary is down, our connections will be moving towards the standy instance.
    - Google CLoud SQL allows the customer to test the failover, and see that the connection string are going toward the standy.

Read Replicas:

- Read replica is a copy of your master instance. The changes in the read replica from the master will be affecting in near real time.
- By implementing read replicas, we are trying to offload the read data requests directly to the master, instead they will be pointed towards Read Replicas.
-

Google App Engine:

- What can be deployed in APP ENGINE is a web based(http/https) application.
- App engine is a PAAS solution.
- **Database is layer4 (TCP) based, so can't deploy in APP Engine .**
- We can't do anything in the UI , only we can provision this. But can't perform any action from UI> Only through CLI only ewe can perform.
- App engine is more of a dev resource, compared to Cloud Engineer.

The focus in App engine is always wrt writing the code, we no need to worry where exactly my code gets deployed.

- How scalable my application will be.
- What will happen if too many requests come?
- What will happen if no request comes? If my application id idle what happens ? whats the charge back ?
- How will my infra work ?
- I don’t need to worry about anything when I am working on GAE. Because all the above operations are been take care by GAE.
- This is the first product which Google launched. (PAAS)

Note:

- App engine standard is only server less
- It's a Platform As A Service.
- **AWS**: AWS Elastic Beanstalk
- **Azure**: App Services.
- App engine is web focused.
    - I.e., if you want to host a DB, you can't host here . U can go to other solutions like cloud sql in GCP

There are 2 types of environments:

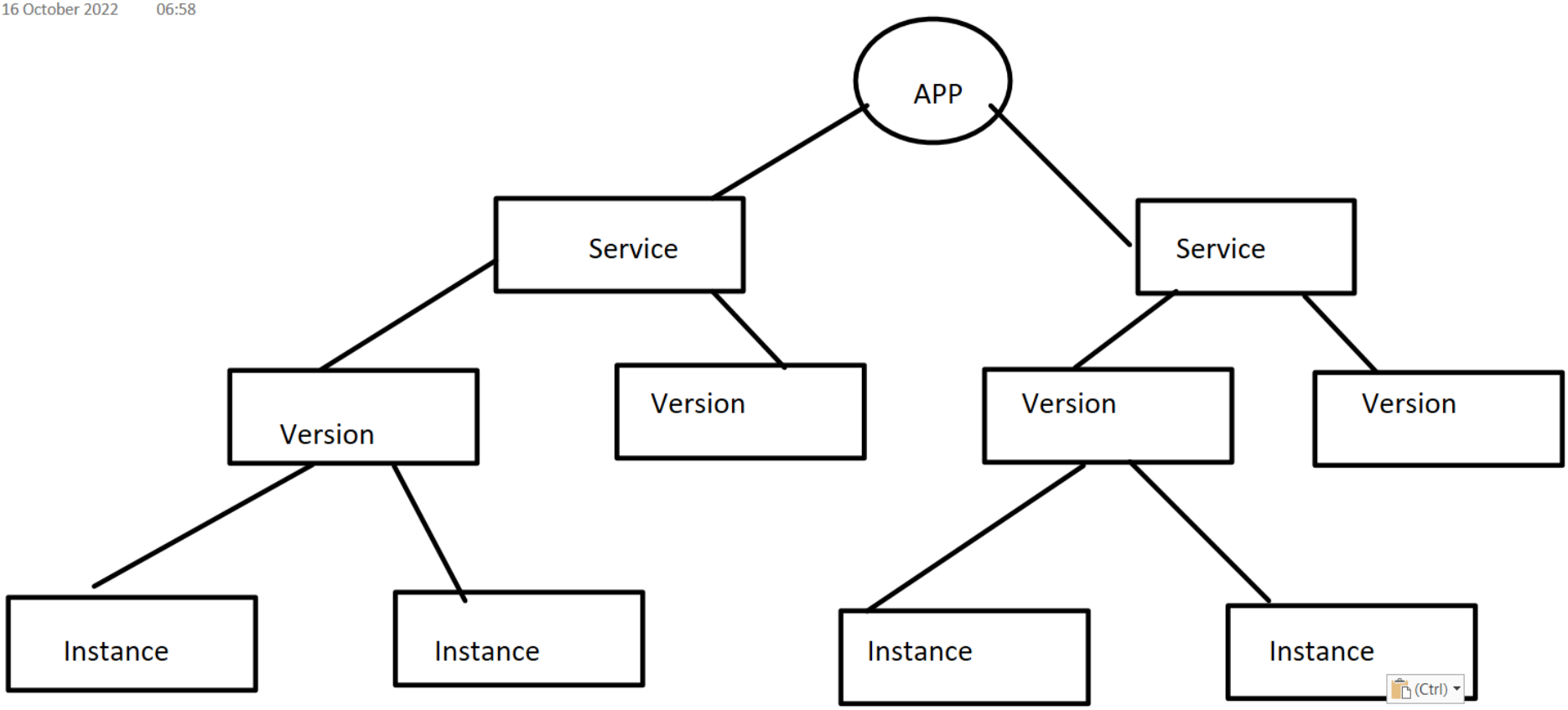
1. Standard Environment (https://cloud.google.com/appengine/docs/standard)
2. Flexible Environment (https://cloud.google.com/appengine/docs/flexible)

Standard environment languages and runtimes

	First Generation	Second Generation
Supported Languages	Python 2.7, Java 8, PHP 5.5, GO 1.11	Python3 , Java 11, PHP 7, Node.js 10,12,14, Go 1.12+

File System Access		
Isolation Mechanism	Proprietary	gVisor-based container sandbox

16 October 2022 06:58



Cloud run:

Develop, deploy highly scalable containerised applications on a fully managed serverless platform,.

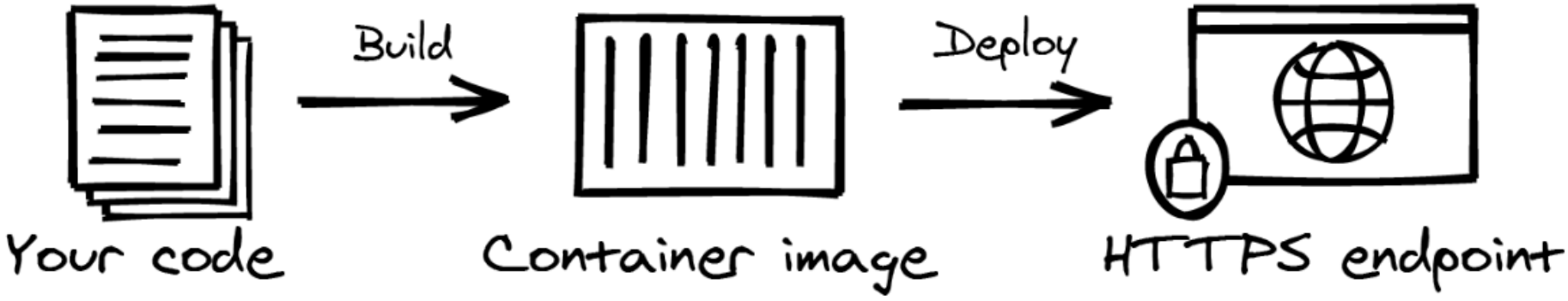
As a developer, i need not be worrying on infrastructure, and autoscaling

Cloud Run is a serverless platform that runs and scales your container-based applica- tion.

It is implemented directly on top of Borg, the hyper-scalable container infra- structure that Google uses to power Google Search, Maps, Gmail, and App Engine.

The developer workflow is a straightforward three-step process. First, you write your application using your favourite programming language. Your app should start an HTTP server. Second, you build and package your application into a container image. The third and final step is to deploy the container image to Cloud Run.

Cloud Run creates a service resource and returns an invokable HTTPS endpoint to you.



Service:

Every service has a unique name and an associated HTTPS endpoint. You'll interact primarily with a service resource to perform your tasks, such as deploying a new container image, rolling back to a previously deployed version, and changing configuration settings like environment variables and scaling boundaries.

Cloud Run is designed for stateless http containers , this dictates 2 main design points,

Cloud run is best suitable for web services,

- your container should include one web server
- The service will be externally be used using Google Load Balancer, i.e., SSL termination provided by Cloud Run (No need to put any certs inside your container)
- Your container should be stateless, same like functions. In order to have horizontal scaling your container should be stateless.
    - Each Container is an identical container
        - Should not use any internal storage
- Develop and deploy highly scalable containerized applications on a fully managed serverless platform.
- Cloud run must contain a http web server
- SSL termination provided by CLoudrun

Cloud Run Container Requirements:

- CR will be supporting any programming language or any framework , because we re deploying containerized images.
- The application we are trying to deploy, should be listening on http requests,
- Ideally CR deploys application on 8080
- The most important point to rember while choosing CR is , it canot rely on the persistent storage.
    - Meaning the data that is been stored in the container will not be permanent, and it is ephemeral.
    - Inorder to store the data persistently , we need to use other google services like Cloud Storage, Cloud SQL, etc ‘
- Cloud run takes container images and creates services.

- Those container images should be store in Google Container registry only.

A simple Developer Experience:

Eliminating infrastructure management means you can focus on writing your code and have someone else worry about deploying, running, and scaling your application. The platform will take care of all the important and seemingly simple details that are surprisingly hard to get right, like autoscaling, fault tolerance, logging, monitoring, upgrades, deployment, and failover.

One thing you specifically don't have to do in the serverless context is infrastructure management. The platform offers an abstraction layer. This is the primary reason we call it serverless.

When you are running a small system, infrastructure management might not seem like a big deal, but readers who manage more than 10 servers know that this can be a significant responsibility that takes a lot of work to get right. Here is an incomplete list of tasks you no longer need to perform when you run your application logic on a serverless platform:

- Provisioning and configuring servers (or setting up automation)
- Applying security patches to your servers
- Configuring networking and firewalls
- Setting up SSL/TLS certificates, updating them yearly, and configuring a web server
- Automating application deployment on a cluster of servers
- Building automation that can handle hardware failures transparently
- Setting up logging and metrics monitoring to provide insights into system performance

Stateless:
Each instance is an identical provider
Should not use any internal storage.

Main CLoud run features:

- Traffic management
- Deployments rollbacks
- App Versioning
- Schedules and triggers

Cloud Run Versions:

- Fully Managed Cloud Run
- Google Anthos
- These 2 are available previously. But now there is only one fully managed serverless platform,

Artifact and Container Registry:

What is Artifact registry:

- Artifact registry will allow the user/customer to store artifacts that are generated when we are trying to build the application.
- These artifacts are been produced during the build stage
- These artifact registry can be:
    - Container images
    - Language specific
        - Java artifacts
        - Node artifacts
    - OS packages
        - RPM

-

What is Google Container Registry?

- GCR is a private Docker registry
- All the image which we store here are default **private**, if required we can make them public as well.

Going forward, its recommended to use Artifact Registry, because artifact registry supports both container and artifacts as well.

- AIn artifact registry we will few more functionalities
- We can mange using IAM
- We will get something called as Container scanning
  - We can use container scanning to find vulnerabilities inside our image.
- Binary authorization:
  - We can have signed images to deploy in cloud run or gke.
  -

CI-CD

- CI/CD > Continuous integration / Continuous delivery or Continuous deployment
- Continuous Integration:
  - Devops culture, of submitting smaller code changes,
    - Developers will be submitting multiple checkins per day
    - Nowm, this code should be continuously update into the repo
    - Smaller changes ⇒ Fewer bugs to fix/troubleshoot
    - Faster Chnages ⇒ we can rapidly develop/deploy our application and we can get rapid feedback.
    - Rapid feedback ⇒ It will be easy to resolve the issue

        - This will result in Qulaity of the product.
    - CI Process:
        - We will be using Source
        - Build
        - Test
        - Report
        - Release
            - Or go to first step again
    - To make this work , we need to have component for the CI:
        - There should be a repository for your code.
            - This repo will be acting as “Single Source of truth”
        - We need to have build and test tools:
            - Now we need to do as much as automation.
            - We can use gradle, maven, etc
        - Artifact repository:
            - To store the artifacts that are been generated by the build process.
- Continuous Delivery:
- Continuous Deployment:
-
-

CSR:

What is Cloud Source repository:

- **It is a fully managed Private Git repository**
- As this is a Google Product, native GCP integration is done.
    - Integration with other google cloud products like Cloud Build, GAE, GCE, GKE, Functions, Monitoring
- If we are already having a repository in 3rd party apps like github, we can do sync/mirror.
    - One-way mirror from Github and bitbucket to CSR
- Powerful code search possible.
    - “Google search for your Code”
- I can grant IAM access to complete CSR or a specific repo.
- Soure Repository Admin/Write/Reader
    - Admin: Full access (adding new roles, creating/deleting the repo)
    - Writer: Only edit and commit the code
    - Reader: viewer only access

Cloud Build:

- If i want to deploy an application in kubernetes, i need to perform below steps:
    - Create/update the code
    - Build a container image
    - Push the image to registry
    - Create manifest files (Deployment, service….)
    - Apply those manifest files.
    - If anything is broken, need to perform the same steps again
- When we are performing lot of manual changes, there is a greate chanc of human error
    - More manual action = great chanc of human error.
    - Cloud build = Automation engine (Cloud build, will be used to automate all the above steps)

- The only step we need toperform is to update the code.

What is Cloud Build:

- It is a fully managed CI/CD engine for your pipeline.
  - Import source code
  - Build in managed space
  - Create artifacts
    - Docker images, java artifacts, and more
    - We are going to focus on docker images in this course

Cloud Builders:

- Cloud builders are Container images that run the build process as containers
- These images are Packaged with common languages and tools.
- We can Run specific commands inside builder containers using the builders.
- They are multiple types of builders:
  - Google Managed
  - Community Managed
  - Public DOcker hub images or customer created ones
- And also use custom build step.

Build File:

- Build files, Provides instruction for cloud build to perform certain tasks using cloud builders.
- Build file, tells cloud builders what to do .
- This will typical be in yaml or json files
- The file name can be cloudbuild.yaml or cloudbuild.json
- This build file provides parameters such as docker image(builder), argument, environmental variable, etc

What cloud build can do ?

- Cloud build, takes the build file and build the steps. And that build file is called as cloudbuild.yaml using **–config**
- Cloud build, can also build image and push to GCR with an option of **–tag**

Kubernete

What is Kubernetes :

- K8S is an Open-Source container orchestration system
- Automated application deployment, scaling and management. (We won't be informed if the container goes down in docker).
- Why to use K8S:
    - Can't live with single server deployment
    - Inter host communication of containers
    - Deploying and updating software at scale
    - Auto-Healing
    - Logging and Monitoring . (In docker we are stuck to a single system)

Kubernetes Architecture:

Cluster Architecture:

The purpose of k8s is to host the applications in the form of containers in an automated fashion, so that we can easily deploy as many containers as are required.
There are many components that will make this happen.

Master node is also known as control plane node .

Master node

It is mainly used for managing , planning, scheduling and monitoring the nodes and does all of these using Control plane components.

Worker node

Worker nodes are the one that is hosting your application .

Worker node is also known as data plane.

Kube-apiserver:

- The API server is a component of the Kubernetes control plane that exposes the Kubernetes API.
- The API server is the front end for the Kubernetes control plane.
- The kube api-server is responsible for orchestrating all operations within the cluster.
- When we run the kubectl command , the kubectl utility is in fact reaching the api server server.
- It exposes the Kubernetes API which is used by external users to perform management operations on clusters as well as various controllers to monitor the state of the cluster and make the necessary changes as required and by the worker node to communicate with the server.

- The kube api-server first authenticates the requests and validates .
- It then retrieves the data from the etcd server and responds back with the required information.

ETCD:

- ETCD is a backend data store for the Kubernetes cluster.
- ETCD is a distributed, reliable key-value store that is Simple, Secure and Fast.
- It provides high availability storage for all data relating to the state of the cluster.
- Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.
- The etc datastore, stores information regarding the cluster such as nodes, pods, config, secrets, accounts, roles, binding etc.
- Every information we see when we execute **kubectl get command** is from etcd server.
- Every change you make to the cluster, such as adding additional nodes, deploying pods, or replicas sets are updated in the etcd server.
- Only once it is updated in the etcd server the changes are considered to be completed

Kube controller manager :

Kubernetes Cluster Creation:

- THere are multiple ways to create a cluster. Following are the few ways to do that:
    - Minikube
    - Kubeadm
    - Kops
    - Installing through Binaries
    - Managed Kubernetes cluster(Ex: GKE , AKS, EKS)
- We will be going through creating a K8S Cluster using **Kubeadm**

- Kubeadm is a tool that simplifies the process of building Kubernetes clusters.
- The **kubeadm init** command initialises a control plane.
- [Creating Cluster using Kubeadm]

Kubernetes Objects:

Namespaces:

- Namespaces is logically dividing our cluster to deploy resources.
- Namespaces provide a way to keep your objects organized within the cluster.
- Every object belongs to a namespace. Namespaces are a way to separate and organize objects in a cluster.
- When no namespace is specified , the cluster will assume the default namespace
- Default Namespace: For objects with no namespace mentioned
- Kube-system: For objects created by the kubernetes system.
- Each Kubernetes resource can only be in one namespace.

Working with Kubectl:

What is a Kubectl ? Kubectl is a command line tool that allow you to interact with Kubernetes. Kubectl uses the Kubernetes API to communicate with the cluster.

1. Imperative
2. Declarative

The Kubernetes command-line tool, kubectl, allows you to run commands against Kubernetes clusters. You can use kubectl to deploy applications, inspect and manage cluster resources, and view logs.

GKE:

Packer:

Packer builds machine images.
Packer works cross-platform, which means we can create machine images for everything from cloud platforms, like AWS, to old-school virtualization tools, like VirtualBox.
This means outside of learning how to write JSON- or HCL2-based Packer templates, any concepts we're introduced to won't necessarily be new — we just need to learn how to use them within our Packer template.

What is a machine image ? a resource that has all confirmations packages to create a virtual machine
It is also called a s golden image

We can write the templates in both JSON and HCL 2

Why packer ??

- Creating images via templates :
    - No need to spin up the instances, and make changes yourself. Describe your desired machine image, and let packer generate it for you.
- Automate all the things:
    - Packer fits in nicely with existing CI/CD pipelines, Use it to automatically, test new configuration changes pr push auto approved builds to prod.
- Create identical images:
    - Cross-cloud, cross-environment or cross-platoform. A single packer template can create machine images for multiple use cases.

Packer breakdown:

What is packer build of ????

1. Builder : Define the desired platform and platform configurations , including APi key information and desired source image.
2. Provisioner: Define how to configure the image, most likely by using your existing env,
3. Post-processor: related to the builder, runs after the image is built, generally generates or supplies artifacts
4. Communicator: how the packer works on the machine image during creation.
    a. By default it is ssh and does not need to be defined
    b. Winrm communicator for windows

Template Language:

Build: The combination of previous components to create a machine image. A template can have multiple builds.

Artifact: the end result of a build, generally this is your build image itself and any generated log or metadata files.

Terraform:

- With Infrastructure as a code, we can manage any resource such as database, networks, storage or even application configuration.
- With the help of IAC tool, we can automate the resources in cloud/onprim as well.
- When we talk about IAC, we refer to terraform and ansible as well.
- Although Ansible and Terraform are IAC tools, they have some key differences in what they try to achieve.

Types of IAC tools:

- Configuration management
    - Below are the few examples of Configuration management:
        - Ansible
        - Puppet

        - Saltstack
        - Chef
    - These tools are designed to install and manage software.
    - By using these tools, we can maintain a standard structure.
    - These cm tools will help us to configure our code in any version control.
- Server templeting
    - Below are the few examples of this:
        - Docker
        - Packer
        - Vagrant
    - These are pre installed software , packages and dependencies.
    - Server templateing tools also prompt **Immutable infrastructure unlike Configuration management tools.**
    - This means, once the vm or the container is created/deployed using server templating tools, they remain unchanged.
    - If there are any changes that needs to be made to the image, instead of updating the running instance, like in case of configuration management , here we will update the image and redeploy a new instance using the updated image.
- Provisioning tools
    - Below are the few examples of this:
        - Terraform
        - CloudFormation
        - Google Deployment manager
    - Deploy immutable infrastructure resources.

Configuration management:

- This is mainly designed to install and manage software
- Maintain a standard structure for this

- All the configurations can be kept in a source code for version control
- Mainly it is idempodent

Server Templating:

- Server templating tools also promote immutable infrastructure unlike configuration management tools. This means that, once the vm or container is deployed it is designed to remain unchanged.
- If there are changes to be made to the image, instead of updating to the running instance, like in the case of configuration management tools such as Ansible , we update the image and then re-deploy a new instance using the updated image.

Terraform woks in 3 phases:

- Init:
    - During the init phase, terraform initialised the project and identifies the provider to be used in the target environment.

Hybrid Networking:

Connecting your network or your infra (from outside GCP) to GCP over private connection.

- By default google provides 3 products
    - Cloud VPN:
        - Private, encrypted tunnel to **GCP VPC** over public internet Connection.
    - Cloud Interconnect:
        - It's a dedicated physical connection to **GCP VPC** from my Source
    - Cloud Peering :

- Dedicated connection to **Google** , **but not your VPC.**
- **Not similar to VPC Perring**

Cloud VPN:

- By default Cloud VPN has 2 types of Routes:
  - Static Route
  - Dynamic Route
- Cloud VPN supports only **Site-to-Site** connection
  - Nosite-to-client option.
  - Example: connecting to GCP vpn via a laptop

Project Flow:

- Create a python application .
- Create a Dockerfile
- Containerize the application
- Build the image
- Push the image to GCR
- Create Cluster:
  - Create a GKE regional cluster
  - Create a Deployment with 3 replicas in the dev-ns namespace .
  - Expose the Deployment to outside world.
  - Create a DNS, which will map to the workload.

- Create a Repo in CSR
- Create a cloudbuild.yaml and write configuration wrt deploying in CR
- Implement the same with triggers.