INTRO

Monday, September 5, 2022 8:43 AM

Google gives a containerized data center in cloud for the infrastructure
PAAS - APPENGINE,CLOUD FUNCTIONS
IAAS - GCE
SAAS - GMAIL,GOOGLEWORKSPACE
MOST OF THE HARDWARE IN THEIR DATA CENTERS ARE CUSTOM MADE BY GOOGLE
QUOTAS AVAILABLE:

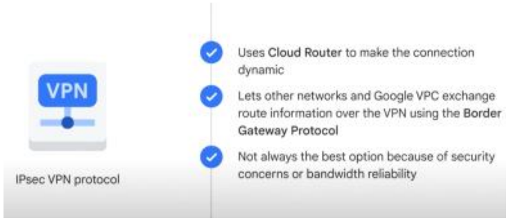
- Rate quota(resets after a specific time)
- Allocation quota(does not reset over time need to delete resources for using the same quota for another)

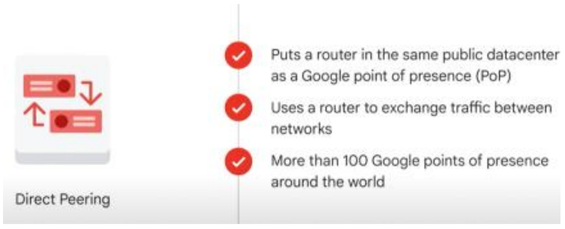
Level1: resources
Level2:projects
Level3:folders
Level4:organizaion node

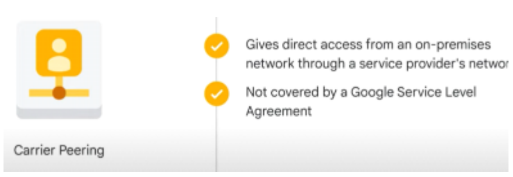
IAM basic role:	**IAM predefined role:**	**IAM custom role:**
Owner,Editor,Viewer,Billing Admin	like Compute instance admin etc..	More granular one(only project/organization level)

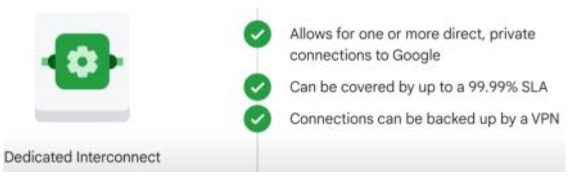
Using cloud identity we can maintain ad groups or ldap systems and create or add or delete users in that group easily through admin console
IAM policies that are implemented by lower-level policies can override the policies defined at a higher level.
Load balancing cloud helps to route traffic for tcp,udp,ssl,https

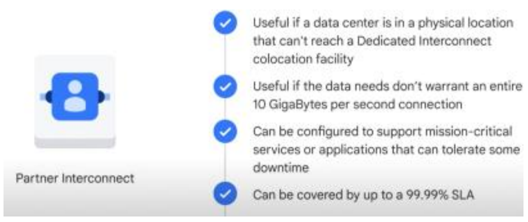
Cloud CDN caches and gives less network latency(we can use cloud cdn interconnect to reuse the existing cdn system we use)
<u>CONNECTING TO VPC NETWORK:</u>


IPsec VPN protocol
- Uses Cloud Router to make the connection dynamic
- Lets other networks and Google VPC exchange route information over the VPN using the Border Gateway Protocol
- Not always the best option because of security concerns or bandwidth reliability


Direct Peering
- Puts a router in the same public datacenter as a Google point of presence (PoP)
- Uses a router to exchange traffic between networks
- More than 100 Google points of presence around the world


Carrier Peering
- Gives direct access from an on-premises network through a service provider's networ
- Not covered by a Google Service Level Agreement

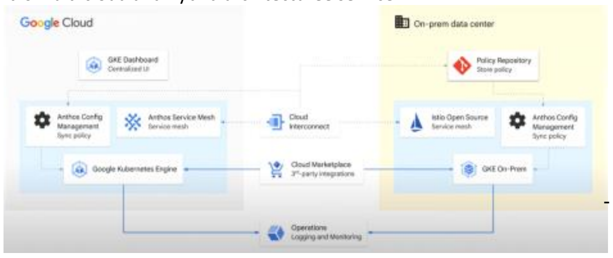

Dedicated Interconnect
- Allows for one or more direct, private connections to Google
- Can be covered by up to a 99.99% SLA
- Connections can be backed up by a VPN


Partner Interconnect
- Useful if a data center is in a physical location that can't reach a Dedicated Interconnect colocation facility
- Useful if the data needs don't warrant an entire 10 GigaBytes per second connection
- Can be configured to support mission-critical services or applications that can tolerate some downtime
- Can be covered by up to a 99.99% SLA

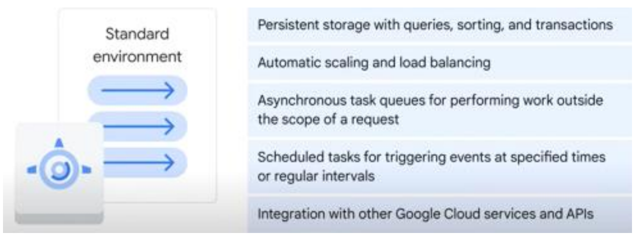
THE GCS CAN STORE RELATIONAL,STRUCTURED,UNSTRUCTURED,TRANSAC TIONAL DATA
The cloud sql is used for gb's and spanner for pb's
Firestore for tb's

A kubernetes pod is a group of containers
A group of machines where Kubernetes can schedule workloads is called kubernetes cluster
Containers use a shared base operating system stored in a shared kernel layer.

Anthos :
It is multi cloud and hybrid architecturee service

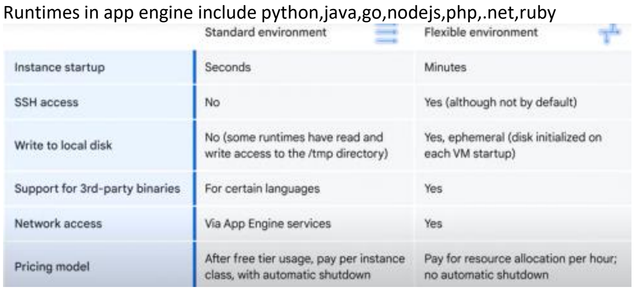


App engine:
- Standard(java,python,go,nodejs,ruby,php)


Standard environment
- Persistent storage with queries, sorting, and transactions
- Automatic scaling and load balancing
- Asynchronous task queues for performing work outside the scope of a request
- Scheduled tasks for triggering events at specified times or regular intervals
- Integration with other Google Cloud services and APIs

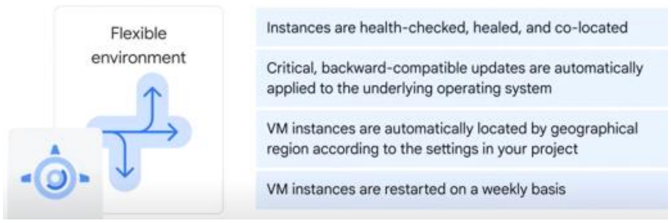
-

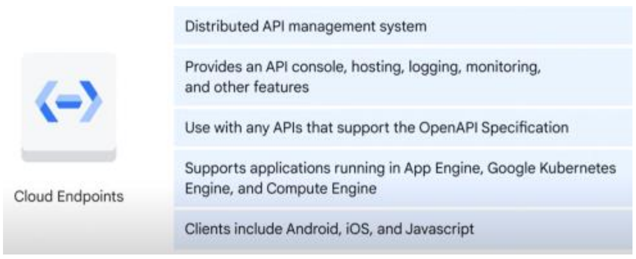
Runtimes in app engine include python,java,go,nodejs,php,.net,ruby



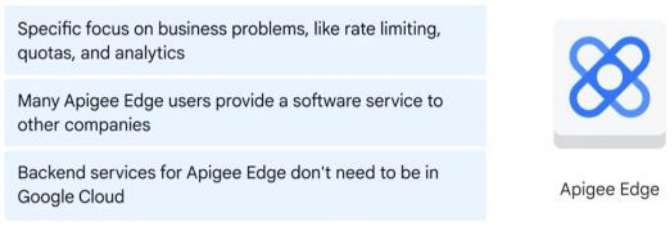
	Standard environment	Flexible environment
Instance startup	Seconds	Minutes
SSH access	No	Yes (although not by default)
Write to local disk	No (some runtimes have read and write access to the /tmp directory)	Yes, ephemeral (disk initialized on each VM startup)
Support for 3rd-party binaries	For certain languages	Yes
Network access	Via App Engine services	Yes
Pricing model	After free tier usage, pay per instance class, with automatic shutdown	Pay for resource allocation per hour; no automatic shutdown

- Flexible(can write to local disk,ssh to vm,install 3rd party binaries

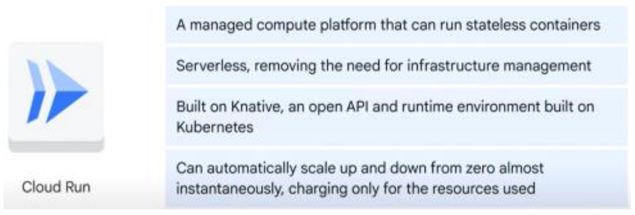

Flexible environment
- Instances are health-checked, healed, and co-located
- Critical, backward-compatible updates are automatically applied to the underlying operating system
- VM instances are automatically located by geographical region according to the settings in your project
- VM instances are restarted on a weekly basis


Cloud Endpoints

- Distributed API management system
- Provides an API console, hosting, logging, monitoring, and other features
- Use with any APIs that support the OpenAPI Specification
- Supports applications running in App Engine, Google Kubernetes Engine, and Compute Engine
- Clients include Android, iOS, and Javascript


Apigee Edge

- Specific focus on business problems, like rate limiting, quotas, and analytics
- Many Apigee Edge users provide a software service to other companies
- Backend services for Apigee Edge don't need to be in Google Cloud

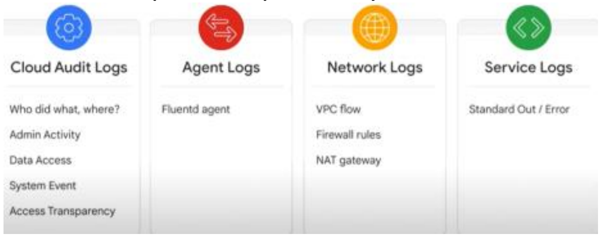

Cloud Run

- A managed compute platform that can run stateless containers
- Serverless, removing the need for infrastructure management
- Built on Knative, an open API and runtime environment built on Kubernetes
- Can automatically scale up and down from zero almost instantaneously, charging only for the resources used

Cloud run will push the containerized image to artifact registry

Source code >> containerized image >> cloud run >> webapp
Source code >> buildpacks(open source project) >> (container image >> webapp)=>cloud run
Client will hit the https endpoint cloud run provides >> cloud run has a Cloud run proxy which
Receives the request and forwards to container through http
As long as our app handles web request we can usecloud run
gcloud beta run services delete helloworld-- to delete cloud run service (if we delete the container we specified still it will run until we delete the run service)

Cloud functions are written in nodejs,python,go and executed in a nodejs environment in gcp
Terraform is used for dealing with infrastructure as a code to manage resources
Saturation in operaions means to measure the utilization
Service level indicators(SLI) - carefullly selected monitoring metrics that measure one aspect of service reliability
Service level objective - combines a sli with a target reliability and will genrally be 100%/99.9%
This should be specific , measurable, achievable , relavent , timebound
Service level agreements - commitments made to your customers that your systems , apps will have a certain amount of downtime
We can also use **Open Source OpenTelemetry** to use custom metrics



Cloud Audit Logs	Agent Logs	Network Logs	Service Logs
Who did what, where?	Fluentd agent	VPC flow	Standard Out / Error
Admin Activity		Firewall rules	
Data Access		NAT gateway	
System Event			
Access Transparency			

Latency,traffic,saturation,errors

Cloud profiler allows developers to analyze apps running anywhere incl gcp with programming languages
Profiler agent is initialized in programming language apps to start collecting logs by including profiler code in them

In Cloud shell to persist the environment variables we should use .profiler
nano .profile; source infraclass/config;

Vpc networks:
============
Default,automode,custom
Automode to custom possible but custom to automode not possible
The auto mode subnets have default /20 CIDR but can be expanded upto /16 only and not able to shrink the range, (eg: if there is a subnet with 10.0.0.0/29 and all the ip addresses are consumed by VM's created then we can edit that subnet and put 10.0.0.0/23 to have more vm's in that ip range
(cidr should not be greater than 29 else will not allow)

For calculating the ip address range that a subnet has we can use a formula for that (eg: 10.0.0.0/29 then use 2^32-prefix i.e 2^32-29 = 8 (ip addresses)

The internal IP address is assigned to the vm's via DHCP protocol and the dns server in gcp will do something like vm name + ip for weburls
We can chose to not to have external ip's when creating vm's also can use reserved static address

The external ip address is not known by VM so it is mapped to the internal ip address

Each instance has a hostname that can be resolved intoa ip address:
- The hostname is the same as the instance name
- FQDN is [hostname/instancename].[zone].c.[projectid].internal
- Eg:myserver.us-central1-a.c.projectname.internal

For external ip's existing dns servers can be used to host , also using cloud dns
Public dns records are not published automatically
The ingress is free and egress is free as long it is with internal ip address and withing the same zone
Static address that are reserved and not in use are charged in a higher rate

Network designs:
- Cloud NAT is used for making connections or connecting between the private instances we created to internet to access it(NAT-network address translation)
- The instances which have the external ip addreses can be directly communicate to google api's and services without intervention but if an instance has only internal ip address then it's subnet

must have the private google access turned on
- The external ip addresses communicate to api's through vpc peering and internet gateway
- In this case the internal ip address can communicate to google's api's and services but not to public internet for that we need to use and configure the cloud NAT gateway for the network
- The cloud nat logs are shown when
  When a network connection using NAT is created.
  When a packet is dropped because no port was available for NAT.

Compute Engine:
- For storage we can use local ssd,zonal,regional persistent disks(hdd/sdd)
- Os linux or windows
- 1 vCpu == 1 hardware hyper thread
- Local ssd are for particular machine types(physically attached to vm)
- When the vm is in termination stage the image cant be changed but others can be changed
- Free command for knowing free space
- Sudo dmidecode -t 17 - for details for the memory installed
- Nproc - for number of processors(Vcpu)
- LSCPU - details of the cpu
- Persistent Disks are not physical disks, they are a virtual-networked service. Each persistent disk remains encrypted either with system-defined keys or with customer-supplied keys.

Machine family:
- General purpose(e2,n1,n2d,n's series)
- Compute optimized(c2,c2d)
- Memory optimized(m1,m2)
- Accelerator optimized(a2)

Discounts/vm's:
- Sustained use discounts- means the more we use the more the discount is given in terms of percentage
- Committed use discounts - committed to use
- Preemptible vm's is the lowest price given and it is terminated automatically between 24 hours
- Spot VM's (new version of preemptible) - here no minimum/max time like 24 hours)
- Sole tenant node - physically isolate our compute engine vms's from other customers like our own license
- Shielded VM's offer secure boot ,Vtpm INTEGRITY THAT REQUIERES SHIED IMAGE
- Confidential VM's allows you to encrypt the data in use(this machine family does this(N2D VM 2ND GEN AMD EPYC PROCESSORS)(only a specific machine type and boot disk can do this)
- VMs in Compute Engine are a collection of networked services which includes persistent disks that are network-attached. In some cases the Google Cloud VM behaves unlike hardware or other kinds of virtual machines, for example, when a multi-tenant virtual CPU "bursts", using excess capacity beyond the VM spec
- Ram disks is another option in choosing disks
- For the sharedcore machien type only 16 persistent disks can be attached
- For other machine types 128 disks can be attached

Common actions:
- While booting startup scripts
- While shutdown shutdown scripts
- Can move an intance to a diff zone (gcloud compute instances move)
  For that references that are there we need to update manually
- For regional move manually we need to snapshot all persistent disks on source vm,create a new disk in the destination restored from snapshots,create new vm and attach the new pd disk,assign statis ip to new vm,update referencesd to vm,delete the snapshots and stuffs created for migration like src vm,disks,snapshots(residing in gcs(not visible by us)(not available for local ssd's)
- We can grow pd but not shrink

you can use screen, an application that allows you to create a virtual terminal that can be "detached," becoming a background process, or "reattached," becoming a foreground process.
sudo apt-get install -y screen ->
- startup-script-url for startup ,shutdown-script-url for shutdown script

Script for background process of vm after ssh is over
#!/bin/bash
screen -r mcs -X stuff '/save-all\n/save-off\n'
/usr/bin/gsutil cp -R ${BASH_SOURCE%/*}/world gs://${YOUR_BUCKET_NAME}-minecraft-backup/$(date "+%Y%m%d-%H%M%S")-world
screen -r mcs -X stuff '/save-on\n'
- sudo crontab -e - for setting cron schedule (0 */4 * * * /home/minecraft/backup.sh inside it)

CLOUD INFRASTRUCTURE CORE SERVICES

Wednesday, September 7, 2022 9:50 AM

Organization is created and managed when a google workspace /cloud identity account creates a project.
Organization admin:
- Define IAM policies
- Determine structure of the resource hierarchy

Compute engine roles:
- In compute admin full control (compute.* roles)
- In network admin role can do whatever on networking resources except firewall rules,ssl certificates
- In storage admin change,modify disks,images,snapshots

Members:
- Google account
- Google group
- Service account
- Cloud identity/workspace domain

IAM Conditions:
If evaluates to true then only the user can access the resource
Organization policy:
- A configuration of restrictions
- Applied to node,folder,projects
- Defined by configuring constraint with desired restrictions

We can use **google cloud directory sync** to convert the microsoft active directory / LDAP into users and groups in your cloud identity domain

For single sign on (SAML SSO) we can use cloud identity to configure sso or can use okta,ping,adfs 3rd party solution to implement it

3 types of service accounts - custom,built-in,internal api's serviceaccount
Users can be granted the service account user role to act as the service account to get the roles that service account has

In user created service accounts can create upto 10 keys and those private keys should be secured by user
But in the google managed it will be secured and managed by google

Gcloud iam service-accounts keys list --iam-account email - listss all the keys

Granting access in service account page lets you to grant users the service account user role or other roles as well to grant access to resources we specify

While creating vm we can specify the service account we created(with service accountuser) to make user granted for that service account to interact with vm or the vm to interact with the permissions in the service account

In buckets when creating acl entries it is only allowed to have max 100 entries
Assigning permission as owner writer reader using acl's
allUsers,allAuthenticatedUsers,gmail account user etcc are the scopes of assigning roles

Gsutil signurl -d 10m pathtoprivatekey gs://bk/objname - creates a signed url

Data import services:
- Transfer appliance
- Storage transfer service

- Offline media import

Filestore is a NAS(network attached storage) for gce,gke instances(usefule for app migration,genomics processing,electronic design autommation,media rendering etc..)

gsutil acl set private gs://$BUCKET_NAME_1/setup.html - to make the object private
gsutil acl ch -u AllUsers:R gs://$BUCKET_NAME_1/setup.html - to make the object publicly accessible
Gsutil get gs://bk/obj - to getthe acl assigned

CSEK:
====

- Create an encryption key using python or some other tool
- gsutil config -n(will create a .boto config file that will be populated with the key we just created in the encryptionkey= field,decryption_key1.key2 etc fields)
- gsutil rewrite -k gs://$BUCKET_NAME_1/setup2.html - to rewrite the file with the boto file config
- To be able to download and see the encrypted file where the encryption keys are not specified correctly need to copy that into a new name in gsutil

Lifecycle/versioning:
=======

- gsutil lifecycle set life.json gs://$BUCKET_NAME_1 - for getting and setting use set/get
- gsutil versioning get gs://$BUCKET_NAME_1 - for getting the status on versioning and if it says suspended then it means that it is not enabled
- gsutil versioning set on gs://$BUCKET_NAME_1 - enabling versioning
- gsutil cp -v setup.html gs://$BUCKET_NAME_1 - to upload an object with versioning means it keeps tracks of the verisons of the object in the bucket with nonconcurrent versions in the object column
- gsutil ls -a gs://$BUCKET_NAME_1/setup.html - list all the versions of that particular objects
- We can use the listed versioned object path to copy to recover it or through console ui we can get it
- gsutil rsync -r ./firstlevel gs://$BUCKET_NAME_1/firstlevel - rsync syncs the directory specified to the target bucket
- gcloud auth activate-service-account --key-file credentials.json - will authenticate the service account and any access (cross project) will be able to do

CLOUD SQL:
For the internal vm/instances to access the cloud SQL instance then we can use cloud sql private ip to connect to the sql and using proxy we will use localhost to access the sql database

Cloud firestore:
Supports acid transactions
Powerful query engine
For new mobile and web apps consider native mode and for new server projects consider datastore mode

Bigtable:
If we don't need transactional consistency in nosql then consider cloud bigtable
In cloud bigtable tablets are similar to hbase regions
In bigtable minimum is 3 nodes then for that even we use it or not we pay for it

Cloud memorystore:

- Redis service
- Instances upto 300gb
- Sub millisecond latency
- Network throughput of 12 gbps

In IAM if child policies is less restrictive and parent level thing is more restrictive then the more restrictive one is overrided with the other one

Global : Images ,Snapshots,Networks

Regional : External IP address
Zonal : Instances, Disks

<u>Quota:</u>
Only 15VPC networks/project
Only 5 admin actions/second (api request limit)
Only 24 vCPU region/project(resources)

<u>Cloud operations suite:</u>

- Monitoring(site reliability engineering(agent can be installed(monitoring agent))
- Logging(platform,system,app logs(30 day retension)(logging agent(gce,ec2))
- Error reporting(apps of programming languages(app engine,run,,gce,gke)
- Trace(latency reporting,latency per-url sampling)
- Debugger(capturing call stack,localvariables of running app

Compute Engine service requires a logging agent installed to collect and send logs to Cloud Operationspp)
Error reporting is supported in gce,gke,app engine standard,flexible etc

<u>Cloud VPN:(connect onprem to cloud vpc network)</u>

- For low volume data connections
- 99.9% SLA
- Supports static routes,dynamic routes via cloud router
- IKEv1,IKEv2 ciphers supported
- On premises network has a VPN gateway which has the external IP address that connects to the internet and to the Cloud VPN gateway through VPN tunnel(encrypted traffic) and the cloud VPN gateway has the regional extenral IP address which is used in clloud routing table and using vp routing is routed to the gcp resources
- MTU(maximum transmission unit) in onprem should not be greater than 1400 bytes
- When connecting both networks through vpn tunnelling the bgp(border gateway protocol) session is created and the bgp should be supported in onprem network
- The both side of the vpn tunnel should have an ip address so that the bgp session is seamless and those must be link-local IP addresses

EG: we have a project with two vpc networks we can take them as one as onprem and other as cloud
The instances having internal ip addresses can only can use that internal ip address within that network not with another network instead they use external ip address but if we want to communicate through internal ip address then we need to establish a secure vpn gateway and tunnelling.

- Create a vpn gateway from the one network (reserve a static external ip address for both networks/ the region in which the subnets are there in those networks) using the static ip address generated for that networks subnet's region
- For the tunnelling to happen in the tunnel section we need to specify the static external ip address we generated for the network 2's subnet region and create a shared key to use when tunneling between two
- The routing options in tunnellling we can specify the routing based and specify the targets subnet's ip range in the form of cidr
- Do the same procedure in both networks
- Note to create the same share key to establish the connection
- The connection is established then we can connect both using internal ip's

<u>Cloud interconnect,peering:</u>

- Direct peering(dedicated,layer3) - no sla
- Carrier peering(shared,layer3) - no sla
- Dedicated interconnect(dedicated,layer2)
- Partner interconnect(shared,layer2)

Layer 2 connections uses VLAN that pipes directly into google network
Layer 3 connections uses public ip addresses to connect to g suite services(like using cloud vpn gateways)
Dedicated connections connect directly to the google's network where
as the shared interconnect uses a partner to connect to the google's network

In the **dedicated interconnect** which says it provides a physical connections will use a on premises router and clou router and make them have the link local addresses for connection and between the two routers the google peering edge existing in a colocation facility and within that the on prem network will/should be there that connects both

If we don’t exists in the colocation facilities available for dedicated interconnect then we can use the **partner interconnect** which uses a service provider network(like airtel,jio etc..) which has the service provider peering edge in the place of a colocation facility so the cloud peering edge in the gcp and the peering edge in the service provider edge will connect using bgp using cloud router In gcp,onpremises router in onprem

IPSec VPN tunnel is used for 1.5-3 gbps per tunnel
Dedicated interconnect for 10gbpsper link / 100gbps (beta)
Partner interconnect for 50mbps - 10gbps per connection

Peering:
- Unlike interconnect which requires a facility/datacenter peering means to connect or establish connection to the points of presence available around the world like the fiberoptic cable movement made available locations from continent to continent
- We can exchange/route our network traffic with google's network by using this
- Direct peering(10gbps per link) requires you to be in the points of presence and for carrier peering(varied based on service provider) requires a service provider for establishing connection
- For using the peering we need to have routable ASN(autonomous system number) ,upto date maintainer,as,as-set etc..)

Shared VPC:
In this the host project which has the vpc network configured willl be shared among multiple projects for using the same internal ip's for communicating with each other with ease(only for organizations)
VPC network peering:
It is decentralized and can connect vpc networks across projects,organizations through private ip's
Like oraganization a peers with organization b through ip and vice versa

ELASTIC CLOUD INFRASTRUCTURE: scaling and automation

Friday, September 9, 2022 7:31 PM

Load Balancers:

- Global
  (HTTPS,SSL proxy,TCP proxy)
- Regional
  (Internal TCP/UDP,network TCP/UDP,internal https)

Managed Instance Groups:
(zonal,regional)
Stateless mig for website frontends etc..
Stateful mig for database consistency etc..
Health check is required to check the health of the instances

Https load balancing:

- Global
- Anycast ip's
- Http or port 80/8080
- Https on port 443
- Ipv4/ipv6
- Url maps for mapping url paths to instances
- This requires SSL certificates to be installed in the for the target proxy and upto 15 ssl certificates needed per target proxy

Network endpoint groups(NEG) is a configuration object that specifies a group of backend endpoints or services
4 types:

- Zonal
- Internet
- Serverless(no endpoint)
- Hybrid Connectivity

SSL proxy load balancing:

- Global encrypted non http load balancing
- Erminates ssl session at the load balancing layer
- Ipv4/ipv6

EG:(the users connect through ssl and the traffic is sent to the cloud load balancing with SSL proxy with port 443)=> connection 1
(after reaching the load balancer the ssl connection is terminated)=>between connection 1,connection2
(
Splits the traffic between the MIG's
)=>connection 2

In the case of **tcp proxy load balancing** the tcp connection/session is terminated at load balancing layer and (SSL trafic <=> TCP traffic)

Network Load Balancing:

- Regional,non proxied load balancer
- Forwarding rules(ip protocol data)
- Traffic:UDP,TCP/SSL ports(ports that are not supported by ssl,tcp proxy load balancers)
- Backend service based architecture,target pool based architecture(upto 50 target pools per project)

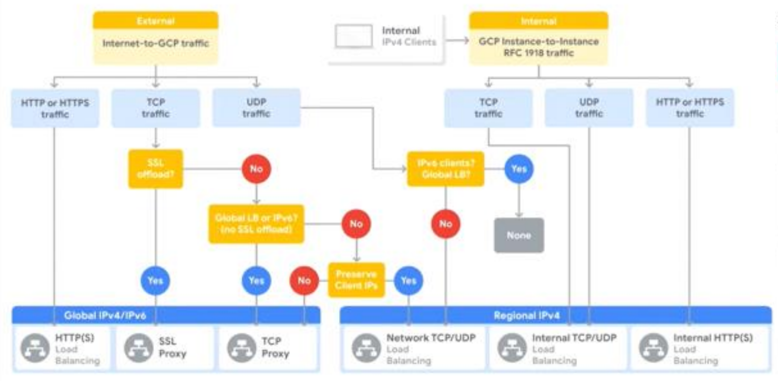
Internal TCP/UDP load balancing:

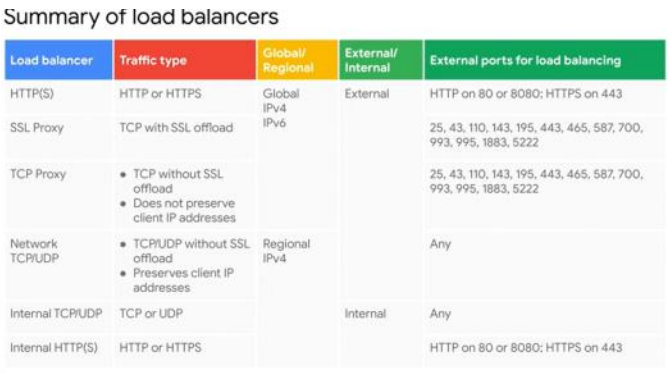
- Regional,private load balancing like vm instances in the same region
- TCP/UDP traffic

Internal HTTP(S) Load balancing:

- Regional,private load balancing
- HTTP,HTTPS,HTTP/2 protocols
- Based on open source envoy proxy

A common 3 tier model is that global https load balancer directs traffic to backend and the back end has internal load balancer for having webtraffic to backend instances for each region





Summary of load balancers

Load balancer	Traffic type	Global/ Regional	External/ Internal	External ports for load balancing
HTTP(S)	HTTP or HTTPS	Global IPv4 IPv6	External	HTTP on 80 or 8080; HTTPS on 443
SSL Proxy	TCP with SSL offload			25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
TCP Proxy	• TCP without SSL offload • Does not preserve client IP addresses			25, 43, 110, 143, 195, 443, 465, 587, 700, 993, 995, 1883, 5222
Network TCP/UDP	• TCP/UDP without SSL offload • Preserves client IP addresses	Regional IPv4		Any
Internal TCP/UDP	TCP or UDP		Internal	Any
Internal HTTP(S)	HTTP or HTTPS			HTTP on 80 or 8080; HTTPS on 443

Terraform:
=========
- Write code in main.tf
- Terraform init
- Terraform plan(refresh)
- Terraform apply(starts provisioning)

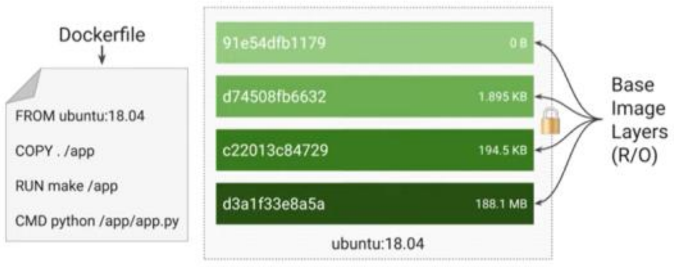
Cloud marketplace gives production grade solutions using terraform

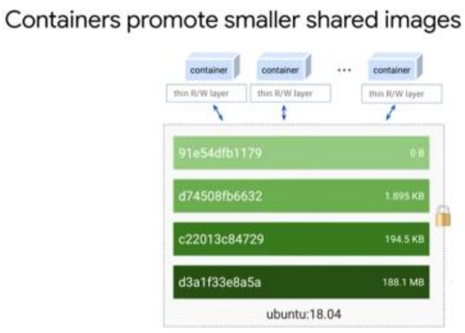
GKE KICKSTART

Sunday, September 11, 2022 11:07 AM

Containers share the same kernel and the hardware and the apps,dependencies are isolated using the container runtime provided after the kernel layer

Containers use a varied set of technologies:
- Processes
- Linux namespaces
- Cgroups
- Union file systems

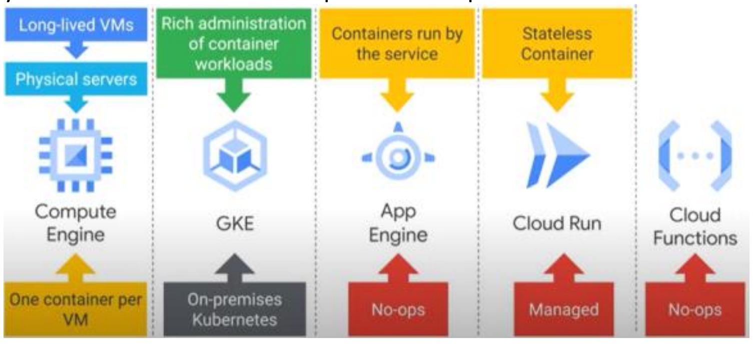




Linux containers use union file systems To efficiently encapsulate applications and their dependencies into a set of clean, minimal layers
The topmost layer's contents are lost when the container is no longer running. An application running in a container can only modify the topmost layer.

Kubernetes:
- Open source
- Automation capabilities
- Container management
- Declarative,imperative configuration
- Supports stateful(persistent),stateless(non persistent) apps
- Autoscaling,resource limits capabilities
- Extensible & portable

In declarative config you describe the desired state you want, and Kubernetes's job is to make the deployed system conform to your desired state and to keep it there in spite of failures



Kubernetes Concepts:
- Kubernetes objects are persistent entities representing the state of the cluster
- Two elements of a kubernetes objects :=> Object spec,Object status
- A kubernetes pod is an object that can have multiple containers and all the containers share the networking and the storage
- Kubernetes frequently monitores the current state(new specs given) and the desired state(already running containers) to communicate to control plane about whether the spec is completely new or is it already present

Kubernetes Components:
- In a kubernetes cluster there are Control Plane and nodes

- Node run pods and Control plane cooperate everything
- Control plane contains:
  Kube-API server(any state change will go to this)
  Etcd(all metadata , info's stores from kube-apiserver)
  Kube-scheduler(scheduling pods to nodes from kube-APIserver)
  Kube-controller-manager controls the cluster through kube-APIserver
  Kube cloud manager manages everything that interact with cloud providers
- Each node in a cluster runs a kubelet(kubernetes agent on each node) and this will monitor/runs the containers/pods and reports to kube-APIserver using container runtime
- Each node also has a kube proxy that is responsible for network connectivity between nodes
- Affinity specifications are used to specify t runn all pods on the same node
- Anti-affinity is vice-versa of affinity (not to run in the same node)

GKE CONCEPTS:
- In this instead of kube-APIserver it is called as clusterIP
- Instead of a single set of nodes we can manage different sets of nodes by using node pools like one node pool of one machine type nodes and other node pool with another big machine type
- Auto upgrade and auto repair of nodes are available
- We can use regional cluster which has separate control plane for each zone but a single clusterIP for all of them (by default the kubernets creates a single node pool with 3 nodes)

Kubernetes objects:
in the YAML specification we give a name to the object which should be unique across all the namespace of the object 253 character length allowed
Alos we can define labels to query or do something with ease
Eg: kubectl get pods --selector=app=nginx

Controller objects are responsible for maintaining the state of the objects
- Deployment
- StatefulSets
- DaemonSet
- Job

Namespaces:
By default three namespaces are ther:
- Default(by default all the pods,deployments sit here
- Kube-system(Config-Map,Controllers,Secrets,Deployments)
- Kube-public(if all namespace should communicate and needs info in common to cluster)

Namespaces let you implement resource quotas across your cluster.
Service:
Used to load balancing across pods
- ClusterIP(expose through an IP that is only acccessible from within the cluster)
- NodePort(expose the service in IP address of each node in the cluster ata specific port)
- LoadBalancer(expose the service extenrally using a load balanced service provider)

In gcp it uses regional network load balancing config by default but we can change it to https global load balancing using an ingress object

A **ReplicaSet** controller ensures that a population of Pods, all identical to one another, are running at the same time. Deployments let you do declarative updates to ReplicaSets and Pods. In fact, Deployments manage their own ReplicaSets to achieve the declarative goals you prescribe, so you will most commonly work with Deployment objects.

Deployments let you create, update, roll back, and scale Pods, using ReplicaSets as needed to do so. For example, when you perform a rolling upgrade of a Deployment, the Deployment object creates a second ReplicaSet, and then increases the number of Pods in the new ReplicaSet as it decreases the number of Pods in its original ReplicaSet.

Replication Controllers perform a similar role to the combination of ReplicaSets and Deployments, but their use is no longer recommended. Because Deployments provide a helpful "front end" to ReplicaSets, this training course chiefly focuses on Deployments.

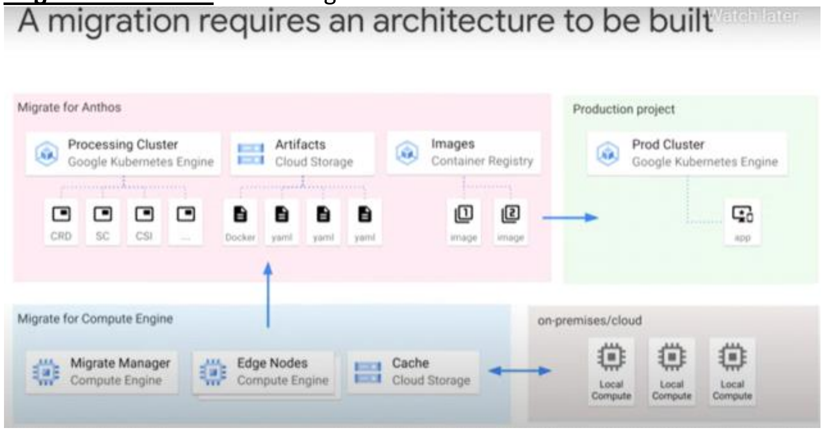
If you need to deploy applications that maintain local state, **StatefulSet** is a better option. A StatefulSet is similar to a Deployment in that the Pods use the same container spec. The Pods created through

Deployment are not given persistent identities, however; by contrast, Pods created using StatefulSet have unique persistent identities with stable network identity and persistent disk storage.

If you need to run certain Pods on all the nodes within the cluster or on a selection of nodes, use DaemonSet. **DaemonSet** ensures that a specific Pod is always running on all or some subset of the nodes. If new nodes are added, DaemonSet will automatically set up Pods in those nodes with the required specification. The word "daemon" is a computer science term meaning a non-interactive process that provides useful services to other processes. A Kubernetes cluster might use a DaemonSet to ensure that a logging agent like fluentd is running on all nodes in the cluster.

The **Job** controller creates one or more Pods required to run a task. When the task is completed, Job will then terminate all those Pods. A related controller is CronJob, which runs Pods on a time-based schedule

<u>Migrate for anthos</u> lets us migrate VM'S into containers



Steps:

- Configure a processing cluster and install migrate for anthos libraries/something
- Add migration source(gcp,aws,azure..)
- Generate and review path (generates yaml files that are used in creating)
- Generate artifacts(container images)
- Test
- Deploy

Step1:

Gcloud container --project projectid clusters create clustername --zone clusterzone --username "Admin" --cluster-version 1.14 --machine-type "n1-standard-4" --image-type "UBUNTU" --num-nodes 1 --enable-stackdriver-kubernetes --scopes "cloud-platform" --enable-ip-alias --tags="http-server"

Migctl setup install (installs required objects)

Step2:

Migctl source create ce my-ce-src --project projectname --zone zone

Step3:

Migctl migration create test-migration --source my-ce-src --vm-id my-id --intent Image

Step4:

Migctl migration generate-artifacts my-migration

Step5:

Migctl migration get-artifacts test-migration

Step6:

Kubectl apply -f deployment-spec.yaml

Kubectl:

- Configurations Relies on a config file : $HOME/.kube/config
- It contains target cluster name , credentials for that cluster
- Kubectl config view - to view the config file
- Gcloud container clusters get-credentials clustername --zone clusterzone

Syntax:

Kubectl command type name flags
Command=> get describe etc..
Type=>pods deployments nodes etc..
Name=>object name
Flags=> -o=yaml - output as yaml

Deployments:

- Rollout,rollback updates,upgrades to pods
- Scale,autoscale pods
- For stateless applications

Ways to create deployment:

- Kubectl apply -f .yaml
- Kubectl run deploymentname --image [image] --replicas 3 --port 8080 --save-config
- Console

Also we can output the deployment into a yaml file using kubectl get deployment name -o yaml > tt.yaml

- Scale the deployment manually using kubectl scale deployment name --replicas=5
- Autoscale using kubectl autoscale deployment name --min=5 --max=15 --cpu-percent=75

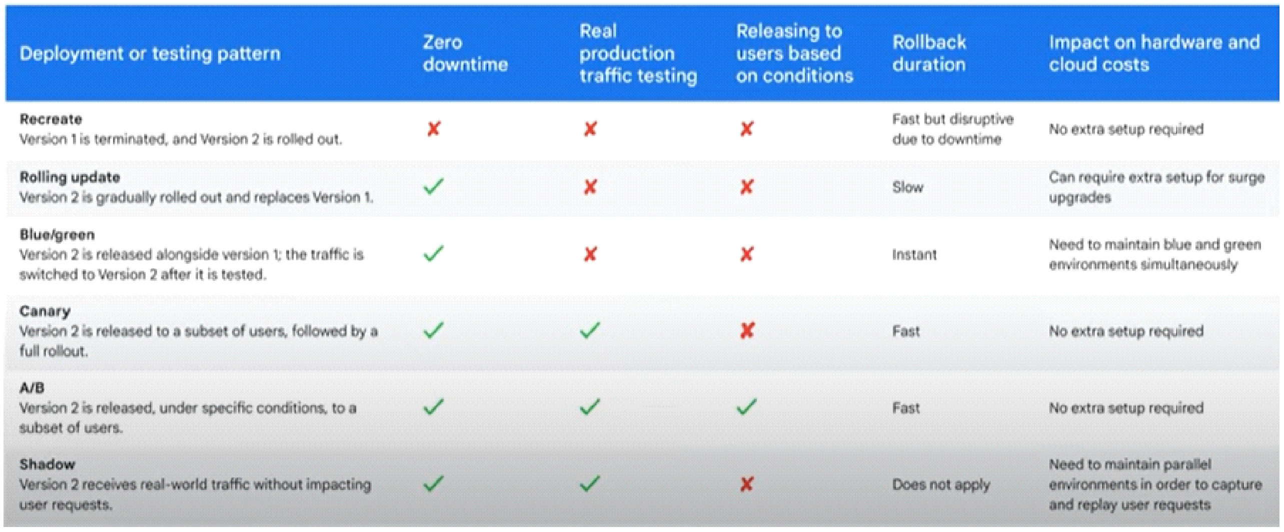
We can update the deployment by

- Changing/updating the config file and running kubectl apply
- Or using kubectl set image deployment deploymentname image image:tag
- Kubectl edit deployment/name - opens a vim editor that updates the deployment
- Using gcp console

Service is a load balancing frontend for pods
Like for frontend there will be a pod and the service load balances the backend for that frontend

For updating the deployments we can use

- Updating the deployments and use the create a new replicaset and creating a pod and deleting the old pod strategy
- (**BLUE/GREEN STRATEGY)**Another is a creating a completely diffeent deployment and switching traffic using service
  The service will have a selector for apps with versions when applying a update as a new version to deployments we create a new deployment version and patch the service to direct the traffic to this new version (kubectl patch service my-app-service -p {"spec":{"selector":{"version":"v2"}}}
- (**CANARY DEPLOYMENT)** in this we don't specify the version but only the app name and we create a new deployment first and scale the deployment with 10 replicas and delte the old deployment(scale down) this means that the traffic will be directed to all the pods with the name specified in the service



Deployment or testing pattern	Zero downtime	Real production traffic testing	Releasing to users based on conditions	Rollback duration	Impact on hardware and cloud costs
Recreate Version 1 is terminated, and Version 2 is rolled out.	✗	✗	✗	Fast but disruptive due to downtime	No extra setup required
Rolling update Version 2 is gradually rolled out and replaces Version 1.	✓	✗	✗	Slow	Can require extra setup for surge upgrades
Blue/green Version 2 is released alongside version 1; the traffic is switched to Version 2 after it is tested.	✓	✗	✗	Instant	Need to maintain blue and green environments simultaneously
Canary Version 2 is released to a subset of users, followed by a full rollout.	✓	✓	✗	Fast	No extra setup required
A/B Version 2 is released, under specific conditions, to a subset of users.	✓	✓	✓	Fast	No extra setup required
Shadow Version 2 receives real-world traffic without impacting user requests.	✓	✓	✗	Does not apply	Need to maintain parallel environments in order to capture and replay user requests

To rollback use

- kubectl rollout undo deployment deploymentname
- Kuectl rollout undo deployment name --to-revision=2
- Kubectl rollout history deployment name --revision=2

We can also change the revision history limit

Kubectl rollout pause deployment name - to pause the deployment
Kubectl rollout resume deployment name - to resume the deployment

Kubectl rollout status deployment name - to view the status of the deployment
Kubectl delete deployment name - to delete the deployment

source <(kubectl completion bash) - for condiguring kubectl bash completion

Before rollout undo the history will be in 1,2
After roolout undo the history will be 2,3 where 3 is the previous verison and 2 is the latest version

sessionAffinity: ClientIP(this in service manifest makes the redendancy of the user requesting to point to a specific pod when doing deployment)

<u>Pod networking:</u>

- Inside pod the networking happens through localhost since all the containers in the pod share the same ip and storage etc..
- But for pods to communicate between each other there is a node networking namespace which has the VM nic that forwards the traffic to other pods
- The nodes get their ip addresses form the ip ranges asssigned to your vpc like each node has its own vms and those vm's have their ip's
- Cluster wide service ip range => 4000
- Pod IP range for the entire cluster => /14 approx

<u>Persistent volumes/storage:</u>

- Volumes
- PersistentVolumes

Volumes are a directory which is accessible to all the containers in a pod
Some volumes are ephemeral too, persistent
Persistent Volumes manage durable storage in a cluster, and are independent of the pod's lifecycle
Can be provisioned manually thourhg cluster admin and dynamically through PersistentVolumeClaims

Ephemeral volume type:

- emptyDir - ephemeral : shared pod's lifecycle
- ConfigMap - object can be referenced in a volume
- Secret - stores sensitive info such as passwords
- downwardAPI - makes data about pods data available to containers

emptyDir volume manifest:

```
apiVersion: v1
…….
Spec:
      Containers:
          - Name:web
            Image: nginx
            volumeMounts:
            -mountmath: /cache
            Name:cache-volume
      Volumes:
           - name:cache-volume
            emptyDir: {}
```

PersistentVolumes abstraction:

- PersistentVolume
  Managed by kubernetes , independent of a pods lifecycle,manually or dynamically provisioned , persistent disks are used by gke as persistentvolumes
- PersistentVolumeClaim
  A pod using PVC submits persistent volume request

Persistentvolume manifest:

```
apiVersion: v1
Kind:PersistentVolume
Metadata:
      Name: pd-volume
Spec:
      storageClassName: "standard"
      Capacity:
        Storage: 100G
    accessModes:
    -ReadWriteOnce:
    gbePersistentDisk:
        pdName:demo-disk
        Fstype:ext4
```

Also we have the PersistentVolumeClaim manifest
That has the claim of requests/storage needed and storage
Class needed and after that manifest is done we provide
that volume claim in Pod manifest

Storage class manifest:

```
Kind: StorageClass
apiVersion: storage.k8s.io/v1
Metadata:
    Name: standard
Provisioner: kubernetes.io/gce-pd
Parameters:
    Type: pd-standard
    Replication-type: none
```

Logging, Monitoring, Observability in Google Cloud

Monday, September 12, 2022 10:40 AM

<u>Operations-based tools</u>:
- Monitoring
- Logging
- Error reporting
- Service monitoring

Logging:
- Collect - Automatic logging on all app engine,cloud run,gke,gce VM's and organized by project,additional log parsing through custom fluentd config
- Analyze - analyze log data in real time with logs explorer,pubsub,dataflow,bq,analyze archived logs in gcs
- Export - to gcs,pubsub.bq or logs-based metric to monitoring
- Retain - data access logs are retained for 1-3650 days and admin logs for 400 days

Available logs:
- Cloud audit logs (who did what,where,when,admin activity,data access,system event,access transaparency)
- Agent logs (fluentd agent,system software,common 3rdparty apps)
- Network logs (VPC flow,firewall rules,NAT gateway,Load balancer)

Error reporting:
Errors in applications
Service monitoring:
- Understand and troubleshoot intra-service dependencies
- Support for app engine,anthos service mesh,istio
- Know when you are meeting or breaking SLO's
- Know when you have error budget to spend

<u>Application performance tools:</u>
- Debugger(realtime app debugging,collaborate by sharing debug sessions,debug snapshots,logpoints,conditional debigging,intergrations with IDE's,used with vcs)
- Trace(for distributed latency analysis)
- Profiler(improve performance and reduce costs,low-impact production CPU and heap profiling,understand your app call patterns)

Debugging:
- Triggers(for outage,dataloss,monitoring failure)
- Response(signal analysis,escalation,customer response)
- Postmortem(documentation,root cause analysis,prevention actions)

Metrics help measure success like if we take the business perspective then the common metrics are return on investment,earnings before interest and taxes,employee turnover,customer churn and in software commmon metrics are pageviews,user registrations,clickthrough,checkouts

S - Specific
M - Measurable
A - Achievable
R - Relevant
T - Tiembound

Four main signals
- Latency(time to return results)
- Traffic(number of requests)
- Saturation(how full the service is and frequently tied to degrading performance) - memory utilization,quotas
- Errors(exceptions,failed requests,dropped connections)

Service level Indicator(SLI) is a measure of service reliability
Servicer level objective(SLO) is a reliability target for an SLI

Error budget is an SLO that implies an acceptable level of reliability

The poor SLI's have a bad graph like zigzag wavy pattenrs that does not show the real thing whereas the good SLI shows the correct graph with clean and neat slope of curves(signal-to-noise ratio is good)
SLI => (GOOD EVENTS/VALID EVENTS)*100%
SLI Menu:
- Request/response => availaility,latency,quality

- Data processing => coverage,correctness,freshness,throughput
- Storage => throughput,latency
- Measurement strategies => app-level metrics,logs processing,synthetic clients/data,front-end-infra metrics,client-side instrmnttion

An error budget is generally expressed as a percentage, and that percentage will hover somewhere near, but not at, 100%
If the SLO is 90% requests must return in 200ms then error budget is 100%-90% = 10.0% 10%

Evaluating alerts:
- Precision - adversevly affected by false positives
  The proportion of events detected that were significant
- Recall - recall is adversely affected by issuing alerts
  The proportion of alerts detected that were relevant to the sum of relevant alerts and missed alerts
- Detection time - raising alerts too fast may result in poor precision
- Reset time - how long alerts fire is resolved

Specifying a maximum duration of time based metrics improves the precision
Supported notification channels:
- Email
- Sms
- Slack
- Mobile app gcp
- Pagerduty
- Webhooks
- Pubsub

Resource groups can be created in the cloud monitoring to alert based on group of resources rather than single resource
There are two types of SLO:
- Request based uses a ratio of good requests to total requests
- Window based uses a ratio of the no of good vs bad measurement intervals

We can set SLO's via service monitoring page
Compliance period in SLO are rolling and calendar

Monitoring IAM roles:
- Viewer
- Editor
- Admin
- Metric Writer
- dashboardEditor
- dashBoardViewer

Aligner in chart creation is to break data into regular time buckets

To generate loads
sudo apt-get update
sudo apt-get install apache2-utils
URL=http://[worker-1-server-vm ip]
ab -n 100000 -c 100 $URL/ - 100 requests/second for upto 100000

The OS Monitoring agent collects os based metrics that is based on open source collectd
The agent metrics include collectd,statd
App engine standard has monitoring built in
App engine flex agent pre-installed and configuredGKE Nodes, monitoring configurable and enabled by default
Anthos GKE onprem,agent collects system but not app metrics
Cloud functions,run integrated monitoring supprt provided
When installing agent permissions are needed then we can create a service account and copy the key file to the environment variable or a
etc/google/auth/dddddd.json in linux and C://program files/google/auth/ddd/json

The logging agent is based on fluentd log data collector, also can add own fluentd config

We can also template this installing of logging,monitoring agents by putting them into a vm image

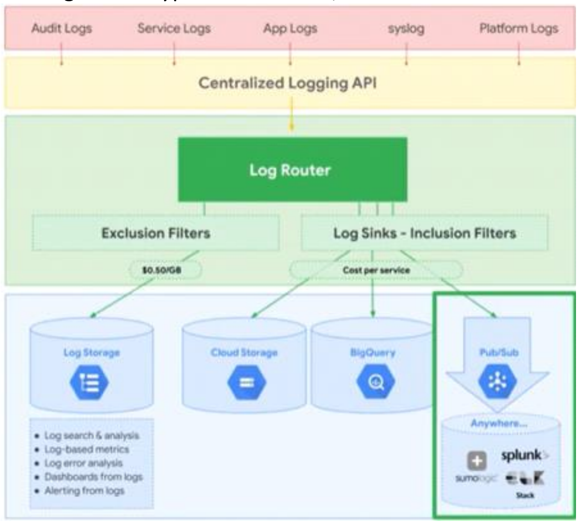
	Base OS install/GCE public image
Period 1:	Hardened OS image
Period 2:	Platform image(frequently building this)
Period 3:	App image(frequently building this)

The hashicorp Packer can automate image builds

Prometheus is an optional monitoring tool for Kubernetes

OpenSensus is a open source library for help to capture manipulate export traces and metrics and supports java,python ,etcc..
If we deploy our app in vm then we need to install logging agent in order to capture it in cloud logging

In the basic logs filter wildcard characters,searching the timetsamp field , range operators,boolean operators are not supported
The log metric types are counter,distribution



For apps in gce or gke we should use error reporting writer role to the app

VPC FLOW LOGS:

- We can enable vpc flow logs per VPC subnet
- It contains details like source ipaddress,port , destination ip address ,port , IANA protocol number
- Also bytes/packets sent,vpc,geographical details

Firewall rules logs can be enabled via creating,editing firewall rules
Gcloud compute firewall-rules update name --enable-logging
--no-enable-logging
Cloud NAT logging allows you to log NAT connections from or to TCP,UDP traffic only,50-100 entries per second per Vcpu
Packet mirroring - clones VPC instance traffic and forwards for examination and this happens At NIC not as part of VPC and it provides access to full traffic flow for regulatory or performance analysis

Network intelligent center:
Centralized network monitoring and visibility

- Topology: view VPC toplogy and associated metrics
- Connectivity tests: evaluate connectitvity to and from VPC resources
- Performance dashboard - packet loss and latecy (VPC)
- FIREWALL INSIGHTS: VISIBILITY INTO FIREWALL ISSUES

Audit logs:

- Admin activity(always enabled and it is retention is 400 days)
- Data access(need to be enabled because of issues with VPC access and it is rentention 30 days default and 3650 if extended)
- System event(always enabled and retention 400 days)

For need to enable data access logging we can do it via gcloud like gcloud projects get-iam-policy project > policy.yaml
And edit the auditotConfigs section for logType:ADMIN_READ,DATA_READ,DATA_WRITE and set-iam-policy and also we can use it at service level or allServices

Incident management roles:

- Incident commander(L1)
- Communications lead(L2)
- Operations lead(L2)
- Primary responder(L3)
- Secondary responder(L3)
- Users and stakeholders

Mitigation is the the action of reducing the severity, seriousness, or painfulness of something.

Trace is a collecitons of spans and span wraps metrics about an app's unit of work(context,timing etc..)

Free allotments in monitoring:

- Logging:50gb/project
- Monitoring:150mb /billing account
- Monitoring api calls:1 mi/project
- Trace ingestion:first 2.5 mil spans/project

- Trace spans scanned: 25 mil spans/project

After free stuff:

- Loggin:$0.50/gb
- Trace ingestions: $.20/million spans
- Logs storage : %0.01/gb

Logs generated in logs explorer is free but exporting costs

The load baalncer and vpc flow logs can be considered when excluding from logs

Reliable Google Cloud Infrastructure: Design and Process

Tuesday, September 13, 2022 7:59 PM

When we design a system we need to define some qualitative requirements like **WHO,WHAT,WHY,WHEN,HOW** questions from users point of view

Personas describe a typical person who plays a role (eg:like a user)

When defining user stories for a case study we need to think of the INVEST criteria
I(independent)
N(Negotiable)
V(Valuable)
E(Estimatable)
S(Small)
T(Testable)
Also the stories need to be like **as a** traveller **I want** to search flights **so that** I can find the best place.

Key Performance Indicators(KPI's) is a metric that can be used to measure success like return on investment,employee turnover,earnings before interest and taxes and in software oage views,user registrations,checkouts etc.. And this should comply with **SMART**

<u>Microservices design and architecture:</u>
Using microservices provide increased complexity when communicating between services,also llatency also sometimes gets increased, concerns about securing inter service traffic,multiple deployments

Should avoid storing shared state in-memory on your servers instead use a backend storage service for state(firestore,sql) and memorystore like services for caching

12 factor app best practices:
- Codebase(VCS)
- Dependencies(gcr for isolated dependencies)
- Config(secrets as env's)
- Backing services(db's,caches are accessed via urls for traeting backedn service as attached resources)
- Build,Release,run
- Processes(run in one or more processes - each instance of the app gets its data from a separate db service)
- Port binding(export services via port binding)
- Concurrency
- Disposability(maximize robustness with fast startup,gracefull shutdown)
- Dev/prod parity(keep all env as similar as possible)
- Logs(treat logs as event streams)
- Admin processes(run admin management tasks as on e- off processes(cronjob,appengine,cloudscheduler)

Grpc protocol(mostly used for internally) supports streaming unlike REST (used for global)

Google's REST api is in the following form:
Service endpoint: https://compute.googleapis.com
Collections: instances,instanceTemplates
Verbs: insert,list,get

OpenAPI (based on swagger)is an industry standard for exposing api's to clients because it will be in a human readable format so that they can understand without the need of a source code
Cloud endpoints,Cloud Apigee for using in gcp

gRPC is for fast binary commmunication between services and us based on http2 ,used in gke using envoy proxy

CI/CD:
- We can enable binary authorization on GKE cluster and add a policy that requires signed images
- When an image is built by gcloud build an attestor verifies that it was from a trusted source repo

- Also gcr includes a vulnerability scanner that scans containers

Cloud memorystore is a vertically scaling service like a cloud sql
For eg : storage,sql,spanner,firestore can handle reads with consistentcy
And bigtable,memorystore replicas can handlle consistent writes

Spanner,bigtable are too expensive for small data

__Networking,designing hybrid architecture:__
- A single vm can have max 8 interfaces(network)
- If load balancer use public IP's secure them using SSL(self managed,google managed SSL certificates)
- For low latency and decreased egress cost leverage cloud CDN
- Https,tcp,udp all support internetfacing/internal config but only https,tcp load balancing supports single/multiregion udp only supports singleregion
- **Network Intelligence Center** can be used to visualize network topology and test network connectivity

HA VPN :
- Automatically chooses 2 external IP addresses,supports multiple tunnels,and tunnels connected to HA VPN gateway must use dynamic routing

1) HA VPN to peer VPN gateway topology:
- Two vpn tunnels are created one from HA VPN gateway to one gateway in onprem
- Another from the same HA VPN gateway to another vpn gateway in onprem
- So in this two vpn tunnels will be there and two bgp routes exchaned through the internet

2) HA VPN gateway to AWS peer gateway topology:
- Here two gateways are from aws virtual private gateway with two tunnels each and in the gcp for referencing the aws another interface will be there

Cloud run is allowing us to use kubernetes(fullymanaged) or cloud run for anthos
Cloud appp engine can serve microservices like one more services having one or more versions in a project and 1 or more instances in each version(traffic splitting supported)
Can use memcache for caching and cloud tasks for scheduling tasks in app engine

__Key Performance metrics:__
- Availability - fault tolerance,backup,health checks
- Durability - replication,regular backups,restore from backups
- Scalability - monitor usage,capacity autoscaling

- to avoid single points of failure follow A spare spare N+2 approach to have one unit for upgrade or testing and survive another failing
- Divide business logic into services based on failure domains
- Deploy to multiple zones or regions
- Circuit breaker pattern does something like if a service has a failure and no requests from the clients should reach the service which is a worse case the circuit breaker will break the chain of transferring it to the service
- In GKE,leverage lstio to implement circuit breakers

Lazy deletion:
- When user deletes the data by mistake
- We can store that to trash for 30 days approx and application delets the data and processes it for soft deletion after 60 days approx and app purges the data after all this only
- This way will if user wants to restore the data can do it through support or aadmins

Always use security command center to see the security checks and security related data

The **Identity Aware proxy** simplifies authorization to app engine standard,app engine flexible,gke,compute engine

Identity Platform provides authentication as a service - providers include openidconnect,saml,google,twitter,fb,microsoft,linkedin etc..

By default in a machine(vm's) ingress is denied and egress is allowed

Also restrict access to services to TLS/HTTPS only

Cloud armor supports layer 7 web app firewall (WAF) rules

- Flexible rules language allows to allow or deny traffic using request headers,geolocation etc..
- Ex: inIpRange(origin.ip,'9.9.9.0/24')
  Request.headers['cookie'].contains('80=BLAH')
  Origin.region_code == 'AU'
  evaluatePreconfiguredExpr('xss-canary')

Data encryption key(DEK) uses AES-256 symmetric key
Keys are encrypted by Key Encryption Keys(KEK)
Google controls root keys in cloud KMS
Keys are automatically periodically rotated
On the fly decryption by authorized user access with no visible performance impact

Clloud CDN CAN HELP reduce the risks of ddos attacks

Cloud armor,cloud cdn,global htto load balancer reduce the risk of ddos attacks

Rolling updates allow you to deploy new versions with no downtime
This is a feature of instance groups ,just change the instance template
Rolling updates in gke are default just need to change the docker image
Completely automated in app engine

We can use blue/green deployment:

- Blue is the current verison
- Green is the new environment in which new veriosn is tested
- Migrate client traffic to green if tested and if failover rollback to blue
- In GCE use DNS to migrate requests from one load balancer to another
- In gke configure service to route to the new pods using labels
- In app engine traffic splitting feature

Canary releases can be used prior to a rolling updte to reduce the risk

- The current serviced veriosn continues to run
- Deploy an instance of the new verison and give it a portion of requests
- Monitor for errors
- In gce create new instance grouo and attach it as a additional backend in load balancer
- In gke craete new pod with same labels as existing pods the service will automatically route a portion of requests to it
- In app engine use traffic splitting feature

In firestore 1 gb approx is free but in another nosql like bigtable 1 gb is cost 1400/month$

For profiler to start getting data from app we need to condfigure the app for starting a profiler session inside the code

```
import googlecloudprofiler

try:
    googlecloudprofiler.start(verbose=3)
except (ValueError, NotImplementedError) as exc:
    print(exc)
```

PRACTICALS

Saturday, September 17, 2022 8:27 PM

- If we delete a project and restore it it will behave differently like it will forget that we are the admin and ask permissions for small things

- To grant the service account or a user of another project we can assign that principal(mail/serviceaccount name) to that resource like for particular bucket/bq dataset
- If need to grant to all the resources in another project for a particular service then we can go to iam of another project and add the service account of another project and give it a role of our choice

In cloud build:

- Series of steps in yaml/json file whenever the code is pushed

  Eg:

```
steps:
  - name: ubuntu
    args:
      - ls
      - '-trl'
    dir: folder
  - name: ubuntu
    args:
      - echo
      - '${_COMMIT_ADDITIONS}'
  - name: ubuntu
    args:
      - '-c'
      - >
        echo ${_COMMIT_ADDITIONS} | rev | cut -d "/" -f1 | rev >
        /workspace/first.txt
    entrypoint: bash
  - name: ubuntu
    args:
      - cat
      - /workspace/first.txt
  - name: gcr.io/cloud-builders/gsutil
    args:
      - cp
      - '${_COMMIT_ADDITIONS}'
      - >-
        gs://asia-east2-airflow-access-t-46ecebe6-bucket/dags/${_COMMIT_ADDITIONS}
```

- Also can use substitution variable using inbuilt variables or user defined(_COMMIT_ADDITIONS) in this case
- _COMMIT_ADDITIONS=$(commit.files['*'].filename)- this will give the filename that was committed when used subtitution variable
- This yaml file will take only the committed file and copies that to cloud storage bucket or we can also use python to test that file

Sole-tenant nodes are physical servers dedicated to hosting only VMs in your project. They're often used for running workloads with extra security, compliance, performance, or licensing requirements.

A network endpoint group (NEG) is a collection of IP addresses that you can apply networking functions to, like load balancing, firewalls and logging. If you're creating an HTTP(S) load balancer with backend containers, your NEG will be created automatically.

FQDN- fully qualified domain name

Network endpoint group (Private Service Connect) is a type of neg where we can target a particular service like kms,logging,pubsub,spanner,cloud run that’s it and there are other neg types like zonal and internet

VPC network peering is used to peer the networks one another
Eg: create peer in one project and do the same in another project both will be connected and it should not overlap ip's and if we ping the ip of anopther project network from the vm of this project it will succed

ssh-keygen -t rsa -f ~/.ssh/KEY_FILENAME -C USERNAME -b 2048 - for generating key pair ssh
If we want to connect the vm from another project with internal ip address only then we can copy the public key present in .ssh/KEY_FILENAME.pub and paste it in the metadata -> SSH keys section and we should not enable os-login=TRUE
Then we can use this command(ssh -i KEY_FILENAME USERNAME@TARGET_INTERNALIP_FROM_ANOTHERPROJECT) now we can connect to the vm
CAN GIVE THE PUB SSH KEY EITHER TO METADATA PAGE OR VM ITSELF
hb

AutoML Tables enforces the following limits on training data:

Maximum size of 100 GB.
Between 1,000 and 200,000,000 rows.
Between 2 and 1,000 columns.
At most 5 dataset imports can be running concurrently.
If you are using CSV, each file must be less than 10 GB.

<u>Spanner</u>:(regional/multiregional)
In Spanner when importing need to specify different database not in the same database

Backups should be created the expiration time which should be 6 hours more from the same time of creation and exports only supported whne in a database

Can create a change streams to capture realtime changes in a database
CREATE CHANGE STREAM SingerAlbumStream FOR (Singers, Albums|ALL); - for specifying what table to watch or all the tables in the database or (TABLE_NAME(col1,col2))

We can use dataflow or any other service to stream the change streams data to bq or gcs to get only the changed data(good for cdc app)

An interleaved table is a table that you declare to be an interleaved child of another table because you want the rows of the child table to be physically stored with the associated parent row. As mentioned earlier, the parent table primary key must be the first part of the child table composite primary key

Monitoring ,Query insights,key visualizer are for insights

This instance cannot be deleted while it contains backups. Delete the individual backups and then try again.

No instance downtime when the instance is edited

<u>Cloud SQL(regional/zonal) (costless than spanner)</u>
Creating instance:
- Machine type configurations(shared core,lightweight,standard,large memory)
- Storage type(ssd,hdd) ,capacity(fixed or enable storage increase to expand storage as it increae)
- Network(private ip-cantdisable once enabled(internal address which is use by internally),public

ip(used outside of gcp externally using proxy or authorized network)
- Backups(point in time recovery,automate backups in schedule(can choose regional,continent level back up store)
- Instance deletion protection
- Maintenance windows
- Flags,labels for customization for everything

Can create a clone of the same instance
When creating an user account we can either allow all form all hosts or restrict from a specific ip address
And also we can assign a principal that can access as this user(SQL)

If we want to secure connections from public IP can use SSL,TLS protocol to allow only SSL connecitons and also we can create upto 10 ssl certificates in the instance and the certificates expire after 10 y

Export will happen only in gcs bucket(same as source while importing except we can select the database/detectautomatically) with SQL,CSV format for entire instance or for specific databases(also can offload export to temporary instance option runs serverless if we want the instance to not have stress because of this)

Can create read replica in order to treat the primary instance as the write one and for reads this will be used(like one instance id for write and another id for replica)

All operations performed are visible in operations tab

When editing an instance we can change the regional to zonal,storage capacity,machine type,add authoried network(check,uncheck private ip),all others coming down

Downtime is there after editing instance/failover

<u>Redis intance(basic(zonal)/standard(regional)</u>
- Only supported upto 300gb
- Can provide read replicas for additional high availability(1,to 5 replicas)
- Connections to this instance can be using direct vpc peering(ip range) ,private service access(sharedvpc)

Private services access connections:
Are per VPC network and can be used across all managed services such as Memorystore, Tensorflow and SQL.
Are between your VPC network and network owned by Google using a VPC peering, enabling your instances and services to communicate exclusively by using internal IP addresses.
Create an isolated project for you on the service-producer side, meaning no other customers share it. You will be billed for only the resources you provision.
- Can schedule redis snapshots

Security
AUTH(SSL)
In-transit encryption
CMEK

<u>Firestore(serverless)(multiregional,regional)</u>
- In native mode only 10000 writes per second allowed but in datastoe mode no limit
- Realtime updates,mobile/web client library with offline data persistenc e is there in native only
- All runtimes are supported in datastore mode but in native some runtimes are not supported
- In native mode it is referred as collections(table),documents(rows) also inside the documents we can create another collection, in datastore mode if we add a new column when creating an entity(row) in kind(table) then it fills - in all other entities and adds this additional column value by inserting an anpther column in entity schema
- **Indexes:**in native mode we can create composite ,single indexes but in datastore mode need to connect to app engine and through that only we can create indexes

- In key visualizer to scan there is a eligibility like maximum 300 document write traffix should happen like that

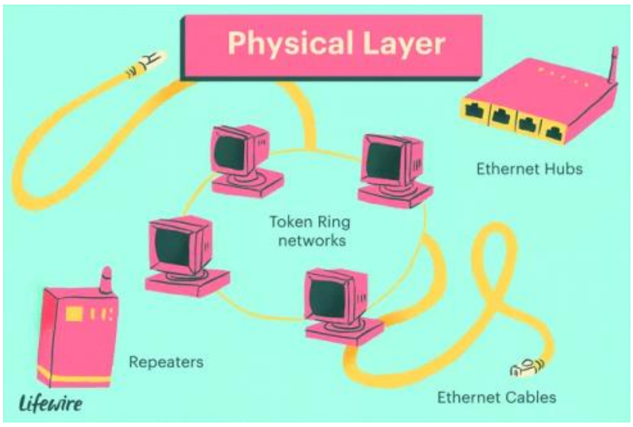
Running commands in background:
- ./script.sh &
- Jobs / ps to see the process
- Kill pid / jobno to kill that process
- Fg jobno - to bring the bg to fg to kill it manually
- After terminal session exit if any process is running we can see that using ps -eaf
- We have to use supervisor in order to run our webapp as daemon and use firewall rules to allow tcp port of the port we use in the app

<u>Bigtable(regional)(but multiregional when added replication cluster)</u>
- Backups,application profiles
- Tables:
- Key visualizer

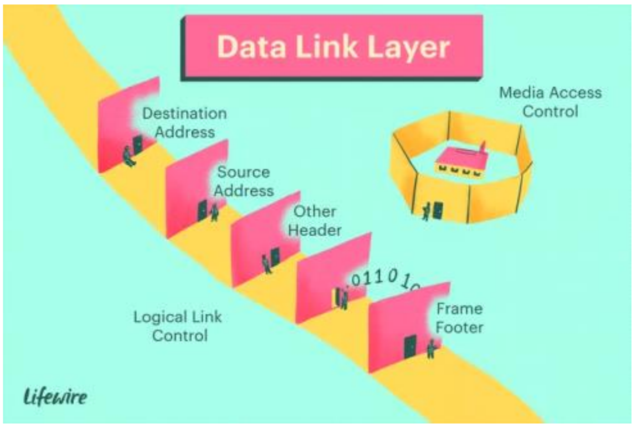
ABOUT NETWORKINGS / OSI MODEL

Sunday, September 18, 2022 3:31 PM

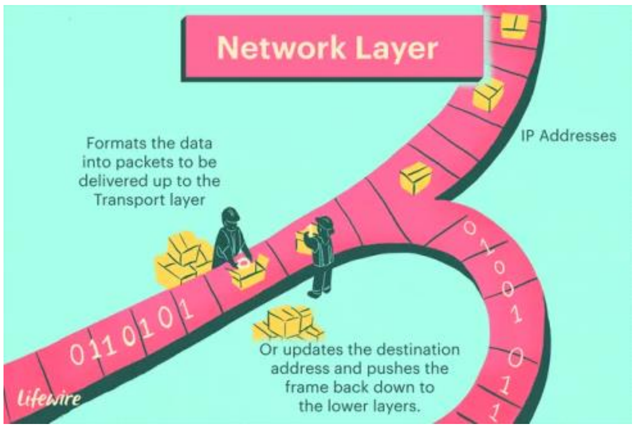
OPEN SYSTEM INTERCONNECTION



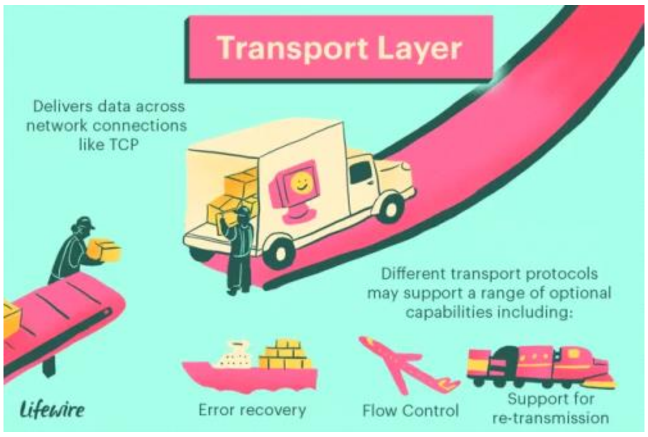
->



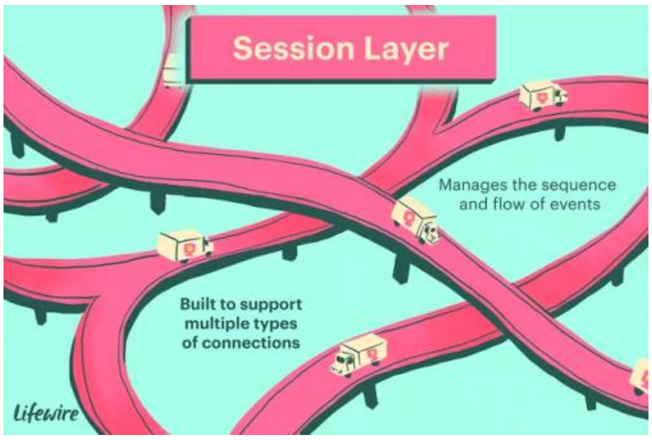
|
V


Network Layer
Formats the data into packets to be delivered up to the Transport layer
IP Addresses
0110101
Or updates the destination address and pushes the frame back down to the lower layers.
Lifewire

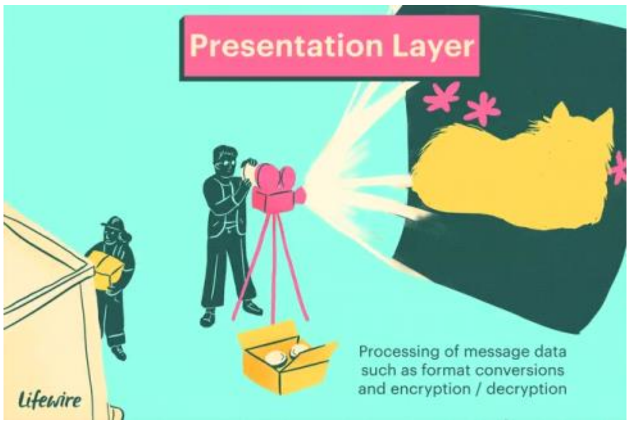
<-


Transport Layer
Delivers data across network connections like TCP
Different transport protocols may support a range of optional capabilities including:
Error recovery
Flow Control
Support for re-transmission
Lifewire

|
V

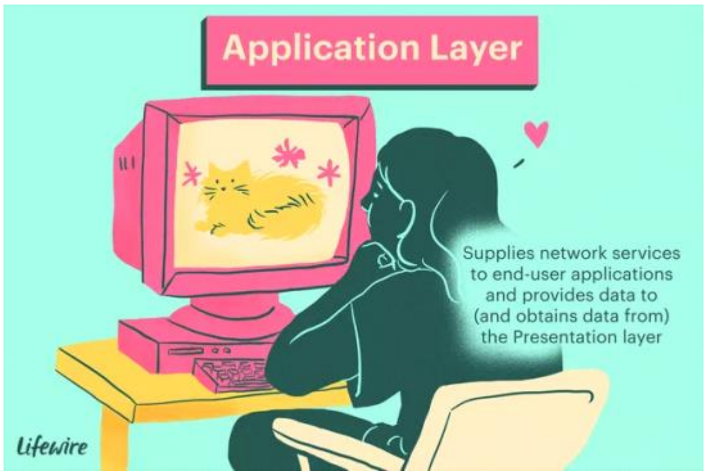

Session Layer
Manages the sequence and flow of events
Built to support multiple types of connections
Lifewire

->


Presentation Layer
Processing of message data such as format conversions and encryption / decryption
Lifewire

|
V

- SSL is secure sockets layer and TLS Is an updated,more secure version of SSL and for the server/app to be ssl secured they need to install the SSL certificate in the server and that is seen by seeing the url like https
- SSL operates on top of TCP protocol(transport layer(tcp,udp)
- SSL starts to work after the tcp connection is established initiating ssl handshake(enables the ssl/tls client to establish the secret keys which they communicate)
- SSL connection error occurs when page being accessed has some security concerns


Application Layer
Supplies network services to end-user applications and provides data to (and obtains data from) the Presentation layer
Lifewire

secure shell(SSH)(public,private key pairs) , SSH file transfer protocol(sftp)

FILE TRANSFER PROTOCOL(in application layer for sending form client to server throuhg tcp/ip) is secured with SSH (sftp) OR SSL/TLS(FTPS)

PRACTICALS

Wednesday, October 12, 2022 4:48 PM

- TO DISABLE horizontal pod autoscaling
  Gcloud containers clusters update clustername --update-addons=HorizontalPodAutoscaling=DISABLED
- Disable kube-dns
  Kubectl scale --replicas=0 deployment/kube-dns-autoscaler --namespace-kube-system
- Limit kube-dns scaling
  Kubectl scale --replicas=0 deployment/kube-dns-autoscaler --namespace=kube-system
  Kubectl scale --replicas=0 deployment/kube-dns --namespace=kube-system

PodDisruptionBudget:
```
apiVersion: policy/v1beta1
Kind: PodDisruptionBudget
Metadata:
        Name: app1-pdb
Spec:
        minAvailable: 2
        Selector:
                matchLabels:
                        App:app1
```

ResourceQuota:
```
apiVersion: v1
Kind: ResourceQuota
Metadata:
        Name: myapp
Spec:
        Hard:
                Requests.cpu: "1"
                Requests.memory: 1Gi
                Limits.cpu: "2"
                Limits.memory: 2Gi
```

LimitRange:
```
apiVersion: v1
Kind: LimitRange
Metadata:
        Name: myapp
Spec:
        limits:
         - default:
                Memory: 512Mi
           defaultRequest:
                Memory: 256Mi
          type: Container
```

gcloud container clusters create scaling-demo --num-nodes=3 --enable-vertical-pod-autoscaling
kubectl get hpa

- Horizontal pod autoscaling adds mmore pods to utilize like a shred cpu or resources across pods
    - Choose the rght buffer size(cpu utilization(75%/45%)
    - Minimize startup time/shutdown time)
    - Accurate ready and live state
      kubectl autoscale deployment php-apache --cpu-percent=50 --min=1 --max=10
- This autoscale command will configure a Horizontal Pod Autoscaler that will maintain between 1 and 10 replicas of the pods controlled by the php-apache deployment. The cpu-percent flag specifies 50% as the target average CPU utilization of requested CPU over all the pods. HPA will adjust the number of replicas (via the deployment) to maintain an average CPU utilization of 50% across all pods.
- Vertical Pod autoscaling increases size of the pod
    - When this is in off state it creates a recommendation chart of the pod behaviours
    - When it is in initial state it does the scaling based on the chart
    - When it is in auto it does the same automatically
    - Handles sudden spikes quicker
    - Set limits,avoid larger bumps
    - Apps should handle restarts

```
cat << EOF > hello-vpa.yaml
apiVersion: autoscaling.k8s.io/v1
kind: VerticalPodAutoscaler
metadata:
  name: hello-server-vpa
spec:
  targetRef:
    apiVersion: "apps/v1"
    kind:       Deployment
    name:       hello-server
  updatePolicy:
    updateMode: "Off"
EOF
```
kubectl get hpa

- Both pod autoscaling must not be enabled like the vertical in off and horizontal done
- Cluster autoscaler does the adding more nodes like adding pods in horizontal pod autoscaling

```
gcloud beta container clusters update scaling-demo --enable-autoscaling --min-nodes 1 --max-nodes 5
gcloud beta container clusters update scaling-demo \
--autoscaling-profile optimize-utilization
```

- Node auto provisioning add node pools like adding node pools for spikes(use preemtible if possible, this takes a while ince it is adding node pool)

```
gcloud container clusters update scaling-demo \
  --enable-autoprovisioning \
  --min-cpu 1 \
  --min-memory 2 \
  --max-cpu 45 \
  --max-memory 160
```

- If we have the deployment /pod in pause state it will create room for other deployments to spin up inside the node
- **For containers shutdown use preStop hook for graceful shutdown,terminationGracePeriodSeconds attribute in maniifest**
- If we use pods for load balancing use network endpoint groups to bypass iptables and for faster access

```
Metadata:
        Annotations:
                Cluster-autoscaler.kubernetes.io/safe-to-evict: "true"
```

For safing to evict

PodDisrupptionBudget:
```
kubectl create poddisruptionbudget kube-dns-pdb --namespace=kube-system --selector k8s-app=kube-dns --max-unavailable 1
kubectl create poddisruptionbudget prometheus-pdb --namespace=kube-system --selector k8s-app=prometheus-to-sd --max-unavailable 1
kubectl create poddisruptionbudget kube-proxy-pdb --namespace=kube-system --selector component=kube-proxy --max-unavailable 1
kubectl create poddisruptionbudget metrics-agent-pdb --namespace=kube-system --selector k8s-app=gke-metrics-agent --max-unavailable 1
kubectl create poddisruptionbudget metrics-server-pdb --namespace=kube-system --selector k8s-app=metrics-server --max-unavailable 1
kubectl create poddisruptionbudget fluentd-pdb --namespace=kube-system --selector k8s-app=fluentd-gke --max-unavailable 1
kubectl create
```

- **For containers shutdown use preStop hook for graceful shutdown,terminationGracePeriodSeconds attribute in maniifest**
- If we use pods for load balancing use network endpoint groups to bypass iptables and for faster access

Since these are non-standard ports, you specify named ports to identify these. Named ports are key:value pair metadata representing the service name and the port that it's running on. Named ports can be assigned to an instance group, which indicates that the service is available on all instances in the group. This information is used by the HTTP Load Balancing service that will be configured later.

```
gcloud compute backend-services create fancy-fe-frontend \
  --http-health-checks fancy-fe-frontend-hc \
  --port-name frontend \
  --global
```
(using the named port defined in mig)

An HTTP load balancer is structured as follows:

1. A forwarding rule directs incoming requests to a target HTTP proxy.
2. The target HTTP proxy checks each request against a URL map to determine the appropriate backend service for the request.
3. The backend service directs each request to an appropriate backend based on serving capacity, zone, and instance health of its attached backends. The health of each backend instance is verified using an HTTP health check. If the backend service is configured to use an HTTPS or HTTP/2 health check, the request will be encrypted on its way to the backend instance.
4. Sessions between the load balancer and the instance can use the HTTP, HTTPS, or HTTP/2 protocol. If you use HTTPS or HTTP/2, each instance in the backend services must have an SSL certificate.

```
gcloud compute http-health-checks create fancy-fe-frontend-hc \
  --request-path / \
  --port 8080
```
(for checking based on the path)

```
gcloud compute instance-groups managed set-autoscaling \
  fancy-be-mig \
  --max-num-replicas 2 \
  --target-load-balancing-utilization 0.60
```

```
system --selector k8s-app=fluentd-gke --max-unavailable 1
kubectl create poddisruptionbudget backend-pdb --namespace=kube-system --selector k8s-app=glbc --max-unavailable 1
kubectl create poddisruptionbudget kube-dns-autoscaler-pdb --namespace=kube-system --selector k8s-app=kube-dns-autoscaler --max-unavailable 1
kubectl create poddisruptionbudget stackdriver-pdb --namespace=kube-system --selector app=stackdriver-metadata-agent --max-unavailable 1
kubectl create poddisruptionbudget event-pdb --namespace=kube-system --selector k8s-app=event-exporter --max-unavailable 1
```

```
cat << EOF > pause-pod.yaml
---
apiVersion: scheduling.k8s.io/v1beta1
kind: PriorityClass
metadata:
  name: overprovisioning
value: -1
globalDefault: false
description: "Priority class used by overprovisioning."
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: overprovisioning
  namespace: kube-system
spec:
  replicas: 1
  selector:
    matchLabels:
      run: overprovisioning
  template:
    metadata:
      labels:
        run: overprovisioning
    spec:
      priorityClassName: overprovisioning
      containers:
      - name: reserve-resources
        image: k8s.gcr.io/pause
        resources:
          requests:
            cpu: 1
            memory: 4Gi
EOF
```

ANOTHER PRACTICALS

Thursday, October 13, 2022 10:02 AM

The management interface for Binary Authorization, a service that provides policy-based deployment validation and control for images deployed to Google Kubernetes Engine (GKE), Anthos Service Mesh, Anthos Clusters, and Cloud Run.

Identity-Aware Proxy (IAP) lets you manage who has access to services hosted on App Engine, Compute Engine, or an HTTPS Load Balancer

SSL policies control how load balancers negotiate SSL with clients. Any HTTPS or SSL load balancer that you set up will have a default policy in place, managed by Google.

You can create policies for closer control over SSL/TLS, but they won't be applied until you attach them to an HTTPS or SSL load balancer.
Cloud IDS (Cloud Intrusion Detection System) provides cloud-native network threat detection with industry-leading security.
Detect network-based threats such as malware, spyware, command-and-control attacks

For live migration when on host maintenance occurs it applies to following conditions:
- Confidential VMs must be set to stop and optionally restart. Compute Engine offers a 60-second notice before a Confidential VM is stopped.
- Same for vm's with gpu's attached
- Cloud tpu's preemtible,spot vm's are not supported for live migrayion

For monitoring the notices on live migrate use http://metadata.google.internal/computeMetadata/v1/instance/maintenance-event metadat server to receive the ststaus like NONE if no maintenance scheduled

99% availablity sla means 14 minutes or so in a day
And 99.9 means 1.4 minutes per day
99.99 means sin milliseconds

Secure boot runs only software that is verified by digital signatures of all boot components using UEFI firmware features. If some software cannot be authenticated, the boot process fails. Software is authenticated by verifying the digital signature of boot components are in a secure store of approved keys.

Virtual Trusted Platform Module (vTPM) is a virtual module for storing keys and other secrets. vTPM enables Measured Boot, which takes measurements to create a known good boot baseline, which is known as the integrity policy baseline. That baseline is used for comparisons with subsequent boots to detect any differences.

Integrity monitoring compares the boot measurements with a trusted baseline and returns true if the results match and false otherwise. Logs are created for several types of events, including clearing the secrets store, early boot sequence integrity checks, late boot sequence integrity checks, updates to the baseline policy, and enabling/disabling Shielded VM options.

We can increase the size of the pd by using resize2fs command in linux
Create a tokenizer service and store only tokenized data for PII data's that need to be processed
To improve the cache hit ratio customize the cache keys to omit the protocol from the key

HIPAA is a US health reguation
PCI DSS is a payment card data security regulation(maybe NA)
SOX is a US regulation on some public traded companies
GdpR is a european privacy protection regulation
COPPA a privacy regulation children

Private address space as RFC 1918 for communicating different vpc networks

CAPITAL EXPENDITURE is the funds spent to acquire assets such as computer equipment,vehicles,land
OPERATIONAL EXPENDITURES is an expense paid for the operating budget
OPERATING BUDGET is allocating funds for labor,supplies etc..

Anthos service mesh can be deployed in three ways
- In cluster control planethe istiod service is run in the control plane to manage security,traffic,config,service discovery
- With managed anthos service mesh google manages the control plane including upgrades
- Anthos mesh for compute engine vm's can manage and secure MIG's and GKE clusters in the same mesh by us

In the multi cluster ingress architecture the config cluster is the GKE cluster running in gcp that is configured with the mutlti ingress resource(running only in gcp environment/deployment)

The anthos config management service controls configuration by applying to configuration specifications to select components of a cluster based on namespaces,labels,annotations.
It also consists of policy controller that enforces business logics on API requests to kubernetes

The minimal deployment option known as attached clusters deployment includes anthos config management, anthos UI, dashboard, anthos service mesh

Cloud tpu's are custom desgined application specific IC's(ASIC)preemtible TPU's are available at 70 percent off standarsd pricing

In cloud storage bucket naming we can use dns names because names can appear in CNAME record in DNS if we use sequential namees then it will create hotspotts to create domain name bucket we have to verify that that is ours

Cloud firestore has three tiers basic/zonal,high scale(for genome analysis)/zonal , enterprise/multiregions
Setfacl also is supported for acess control

Bigquery storage write api provides high throughput ingestion and exactly once deplivery sematics

10.0.0.0/8,172.16.0.0/12,192.168.0.0/16 are considered private address by internete engineering task force
Cloud router provides routing services for dediated,partner inerconnect,ha vpn,supported router appliances

In direct peering works by exhanging bgp routes which define paths bfor transimitting data between

For preparing the source for postgres cloud migration:

```
- sudo apt install postgresql-13-pglogical
  sudo su - postgres -c "gsutil cp gs://cloud-training/gsp918/pg_hba_append.conf ."
  sudo su - postgres -c "gsutil cp gs://cloud-training/gsp918/postgresql_append.conf ."
  sudo su - postgres -c "cat pg_hba_append.conf >> /etc/postgresql/13/main/pg_hba.conf"
  sudo su - postgres -c "cat postgresql_append.conf >> /etc/postgresql/13/main/postgresql.conf"
  sudo systemctl restart postgresql@13-main
- In pg_hba.conf these commands added a rule to allow access to all hosts
  #GSP918 - allow access to all hosts
  host    all all 0.0.0.0/0   md5
- In postgresql.conf, these commands set the minimal configuration for pglogical to configure it to
  listen on all addresses:
  listen_addresses = '*'
- sudo su - postgres
  psql
- \c postgres;
  CREATE EXTENSION pglogical;
  \c orders;
  CREATE EXTENSION pglogical;
  \c gmemegen_db;
  CREATE EXTENSION pglogical;
- \l
  CREATE USER migration_admin PASSWORD 'DMS_1s_cool!';
  ALTER DATABASE orders OWNER TO migration_admin;
  ALTER ROLE migration_admin WITH REPLICATION;
  \c postgres;
- GRANT USAGE ON SCHEMA pglogical TO migration_admin;
  GRANT ALL ON SCHEMA pglogical TO migration_admin;
  GRANT SELECT ON pglogical.tables TO migration_admin;
  GRANT SELECT ON pglogical.depend TO migration_admin;
  GRANT SELECT ON pglogical.local_node TO migration_admin;
  GRANT SELECT ON pglogical.local_sync_status TO migration_admin;
  GRANT SELECT ON pglogical.node TO migration_admin;
  GRANT SELECT ON pglogical.node_interface TO migration_admin;
  GRANT SELECT ON pglogical.queue TO migration_admin;
  GRANT SELECT ON pglogical.replication_set TO migration_admin;
  GRANT SELECT ON pglogical.replication_set_seq TO migration_admin;
  GRANT SELECT ON pglogical.replication_set_table TO migration_admin;
  GRANT SELECT ON pglogical.sequence_state TO migration_admin;
  GRANT SELECT ON pglogical.subscription TO migration_admin;
  \c orders;
- GRANT USAGE ON SCHEMA pglogical TO migration_admin;
  GRANT ALL ON SCHEMA pglogical TO migration_admin;
  GRANT SELECT ON pglogical.tables TO migration_admin;
  GRANT SELECT ON pglogical.depend TO migration_admin;
  GRANT SELECT ON pglogical.local_node TO migration_admin;
  GRANT SELECT ON pglogical.local_sync_status TO migration_admin;
  GRANT SELECT ON pglogical.node TO migration_admin;
  GRANT SELECT ON pglogical.node_interface TO migration_admin;
  GRANT SELECT ON pglogical.queue TO migration_admin;
  GRANT SELECT ON pglogical.replication_set TO migration_admin;
  GRANT SELECT ON pglogical.replication_set_seq TO migration_admin;
  GRANT SELECT ON pglogical.replication_set_table TO migration_admin;
  GRANT SELECT ON pglogical.sequence_state TO migration_admin;
  GRANT SELECT ON pglogical.subscription TO migration_admin;
- GRANT USAGE ON SCHEMA public TO migration_admin;
  GRANT ALL ON SCHEMA public TO migration_admin;
  GRANT SELECT ON public.distribution_centers TO migration_admin;
  GRANT SELECT ON public.inventory_items TO migration_admin;
  GRANT SELECT ON public.order_items TO migration_admin;
  GRANT SELECT ON public.products TO migration_admin;
  GRANT SELECT ON public.users TO migration_admin;

- \c gmemegen_db;
  GRANT USAGE ON SCHEMA pglogical TO migration_admin;
  GRANT ALL ON SCHEMA pglogical TO migration_admin;
  GRANT SELECT ON pglogical.tables TO migration_admin;
  GRANT SELECT ON pglogical.depend TO migration_admin;
  GRANT SELECT ON pglogical.local_node TO migration_admin;
  GRANT SELECT ON pglogical.local_sync_status TO migration_admin;
  GRANT SELECT ON pglogical.node TO migration_admin;
  GRANT SELECT ON pglogical.node_interface TO migration_admin;
  GRANT SELECT ON pglogical.queue TO migration_admin;
  GRANT SELECT ON pglogical.replication_set TO migration_admin;
  GRANT SELECT ON pglogical.replication_set_seq TO migration_admin;
  GRANT SELECT ON pglogical.replication_set_table TO migration_admin;
  GRANT SELECT ON pglogical.sequence_state TO migration_admin;
  GRANT SELECT ON pglogical.subscription TO migration_admin;
-
  GRANT USAGE ON SCHEMA public TO migration_admin;
  GRANT ALL ON SCHEMA public TO migration_admin;
  GRANT SELECT ON public.meme TO migration_admin;

  \c orders;
  \dt
  ALTER TABLE public.distribution_centers OWNER TO migration_admin;
  ALTER TABLE public.inventory_items OWNER TO migration_admin;
  ALTER TABLE public.order_items OWNER TO migration_admin;
  ALTER TABLE public.products OWNER TO migration_admin;
  ALTER TABLE public.users OWNER TO migration_admin;
  \dt
  \q
  exit
```

networks
In CLOUD DNS a records are addresses that map to domain names to ip address, CNAME or canonical records store aliases,MX is a mail exchange record, NS are name server records

Organization constraints examples:

- App Engine Disable Source Code Download, to prevent download of code previously uploaded to App Engine
- Cloud Run Allowed Binary Authorization Policies, which defines the set of Binary Authorization policy names that are allowed to be specified on a Cloud Run resource
- Cloud SQL Restrict Public IP Access on Cloud SQL Instances, to restrict the use of IP addresses on Cloud SQL instances
- Cloud Storage Enforce Public Access Prevention, to prevent public exposure of data by not allowing public access to Cloud Storage data
- Compute Engine Shielded VMS, which requires all new Compute Engine VMs to use Shielded disk images
- IAM Disable Cross-Project Service Account Usage, to prevent service accounts from being used in projects other than the one they were created in

OPERATIONS

Tuesday, October 18, 2022 4:46 PM

For GKE monitoring and stuffs we will use --logging,--monitoring flags while creating or updating clusters to specify the required workloads from which metrics to be collected

Also we can create a resource named PodMonitor in the namespace where the workloads are running and the pods selectors so that metrics will be collected from those pods