GCP

Google Cloud

Professional Cloud DevOps Engineer

Google Certified Professional Cloud DevOps Engineer

Professional Cloud DevOps Engineer

- Pay attention for 5 minutes, before we dive in.
- Advance certification
  - Expectation
    - Basics of Compute Engine,
    - Kubernetes, Docker
- Learn by Doing
- 20/80

GCP certifications

https://cloud.google.com/certification/cloud-devops-engineer

Cloud Cost for this course

- $0 – for GCP account
- GCP Free trial
- $300 for next 3 months https://cloud.google.com/free
- Length: Two hours
- Registration fee: $200 (plus tax where applicable)
- Languages: English
- Exam format: Multiple choice and multiple select,

Google Cloud Devops

BY ANKIT MISTRY

Google DevOps

- Apply site reliability engineering principles to a service
- Build and implement CI/CD pipelines for a service
- Implement service monitoring strategies
- Optimize service performance
- Manage service incidents

GCP Basics

- Google Cloud Overview
- Create GCP Account
- GCP Console Walkthrough
- GCP Regions & Zones
- Creating GCP Projects
- Google Cloud Shell

SRE - Site Reliability engineering

BY ANKIT MISTRY

SRE

- History of Software Development Cycle
- DevOps & SRE
- Role of SRE
- Eliminating Toil
- Blameless Postmortem
- SLI, SLO & SLA
- Error Budgets

History

- History of Software Development Cycle
- DevOps & SRE
- Role of SRE
- Eliminating Toil
- Blameless Postmortem
- SLI, SLO & SLA
- Error Budgets

History

DEVELOPERS

- Developer write code
- Update software
- Adding new feature
- Don’t bother about stability
- They want to push code faster to Prod

OPERATORS

- Operator know how to deploy & monitor application
- Operators don’t know how to write code
- They know how to assemble code
- Solve Production issue
- How to scale Application
- They love less updates

Devops

- DevOps is a set of practices, guidelines and culture
  - which designed to reduce the gap between software development and software operations.
- If Both team work together, productivity will increase
- DevOps established five goals.
  - Reduce organizational silos
  - Accept failure as normal
  - Implement gradual changes
  - Leverage tooling and automation
  - Measure everything

SRE

- There is problem with devops
  - Goal of Devops is broad.
  - Devops does not define how to implement it.
- This is How SRE comes.
- Devops is Philosophy where SRE is implementation of Devops ‘s Philosophy.
- class SRE implements DevOps
- SRE Practices
  - SRE Role
  - blameless postmortems
  - error budget
  - reduce toil
  - track service level metrics , SLIs, SLOs, and SLAs.

SRE Role

- Specific job role
- Old operator role -> SRE Role
- A Site Reliability Engineer is basically the result of asking a software engineer to design an operations team
- SRE requires experience in both development as well as operations
- SRE spends half of their time doing ops-related work
  - production issues, attending call, performing manual interventions
- SRE spends other half of their time in development task, Scaling system, automation
- Compared to old operator, both SRE & Developer share responsibility of Prod Server
- SREs build the tools that developers use to compile, test, and deploy their code. (CI/CD Pipeline)
- Developers and SREs work together to fix issue

Blameless postmortems

- One of goal of DevOps is accept failure as normal
- Failure is un-avoided, However good system you design.
- Once you change system, risk is involved
- If your rate of change is zero, risk is also zero. But that means you are stopping growth.
- Need to balance between change & risk.
- You can take it as opportunity to grow business, if Things break, fix it.
- Fix will teach you lot of thing, minimize future issue.
- In SRE, you can accomplish with Blameless postmortems

Blameless postmortems (Cntd…)

- Idea behind Blameless postmortems
  - is to analyze system failure
  - Root cause behind it.
  - Discuss about what has happened exactly
  - What action need to be performed.
- Not to look for someone who can be blamed.
- Assumption is – everyone had good intentions
- Some postmortems question need to be asked.
  - When incident begin & end?
  - How incident get notified
  - Who are all involved
  - Which system are affected
  - What is root cause of failure
  - How to avoid in future

Blameless postmortems (Cntd…)

- Accept that With Human error are involved.
- Blameless postmortems is
  - Honest Communication with other team member so that similar incident can be avoided in future

Toil

- One of goal of DevOps is leverage tooling and automation
- There is lots of task are manual, laborious.
- Task like Password Change, Copy Files, Creating new Folders, Restart Servers
- These type of task are considered as Toil.
- Identifying Toil is important.
- Not all Task are Toil.
- There are task which is laborious but not necessary is toil.
- Toil is related
  - Prod system
  - Manual, repetitive & automatable task

Toil (Cntd…)

- SRE want to reduce Toil by automation.
- Task like
  - Automate CI/CD Pipeline
  - Schedule Jobs
  - Write some Automation scripts
  - Automate testing
  - No manual Provisioning hardware
- If Repetitive task automated, It should be automated
- Due to Automation, more resource can work something more interesting
- SRE should spend significant amount of time in reducing toil.

Error budget

- One of goal of DevOps is implement gradual change
- Why outage occurs
  - Added new feature, change, new hardware, security patches
- More change leads to less stable system
- How to balance between change & stability
- We have to define metric for high system reliability.
- It is business Problem
- how much can the service fail before it begins to have a significant negative impact?
- How quickly do we need to be able to release new features?
- Depending on target, need to define error budget

Error budget (Cntd…)

- Anytime your service is down, time require to recover it will be consumed from error budget
- After you define error budget
  - as long as you are within error budget, you are good to go for more changes
  - Once you run out of error budget, need to hold all future changes for deployment & make system stable first
- Larger error budget
  - means more downtime for service acceptable,
  - frequent changes possible.
- Less error budget,
  - means less downtime for service acceptable,
  - lesser changes allowed.
- Error budget make sure smaller & gradual changes deployed.

SLI, SLO & SLA

BY ANKIT MISTRY

SLO

- Service level objective
- It is internal objective of team
- SLO is something everyone in org want to achieve
- Error Budget is directly related to SLO
- It kind of complement to Error Budget
- Error – 3% means service is down 3% at max
- SLO – 97% means service should be up for 97%
- Error Budget + SLO = 100%
- Define SLO with respect to latency, Availability, Response Time

SLI

- Service level indicator
- Indicator internal to team
- SLI needs to be compared against SLO
- SLI are metrics which track over time (generally 5 minutes interval)
- SLI ranges from 0 to 100%

$$SLI = \frac{Total\ Good\ Event}{Total\ Valid\ Event}\ X\ 100$$

- Let's say SLO – 96%
  - 96% of request should be serve within 300 ms latency.
- If Current SLI is 95% or anything less than 96%, system is under performing.
- SLI help us to find which service are not performing as per SLO
- Good SLI leads customer happy

SLI (Cntd…)

- Good SLI leads customer happy
- If Changes to SLI does not impact customer, SLI definition is not worth
- Different signal to track
  - Latency
  - Traffic
  - Errors
  - Saturation
  - Availability of system
- Selecting right SLO & SLI will lead to success

SLA

- Service level agreement
- It is contract with consequences of failing to meet the SLOs they contain
- SLO & SLA are quite similar
- But your SLAs should not be the same as your SLOs
- SLO is an internal objective,
  - If you can not meet SLO, team can slow down changes
- SLAs violations are shared with your customers
  - If you can not meet SLA, compensate need to be provided to customers
- https://cloud.google.com/terms/sla
- SLI should be higher than SLO & SLA, means current indicator shows services are performing as expected

SLA (Cntd…)

- If SLI goes below SLO, slow own
- If SLI goes below SLA, notify customer & compensate
- Higher SLA Good but more likely you will violate it
- Lesser SLA means You will meet but customer will have less confident in your services
- Google recommendation in case very high SLA
  - Down your service for some time

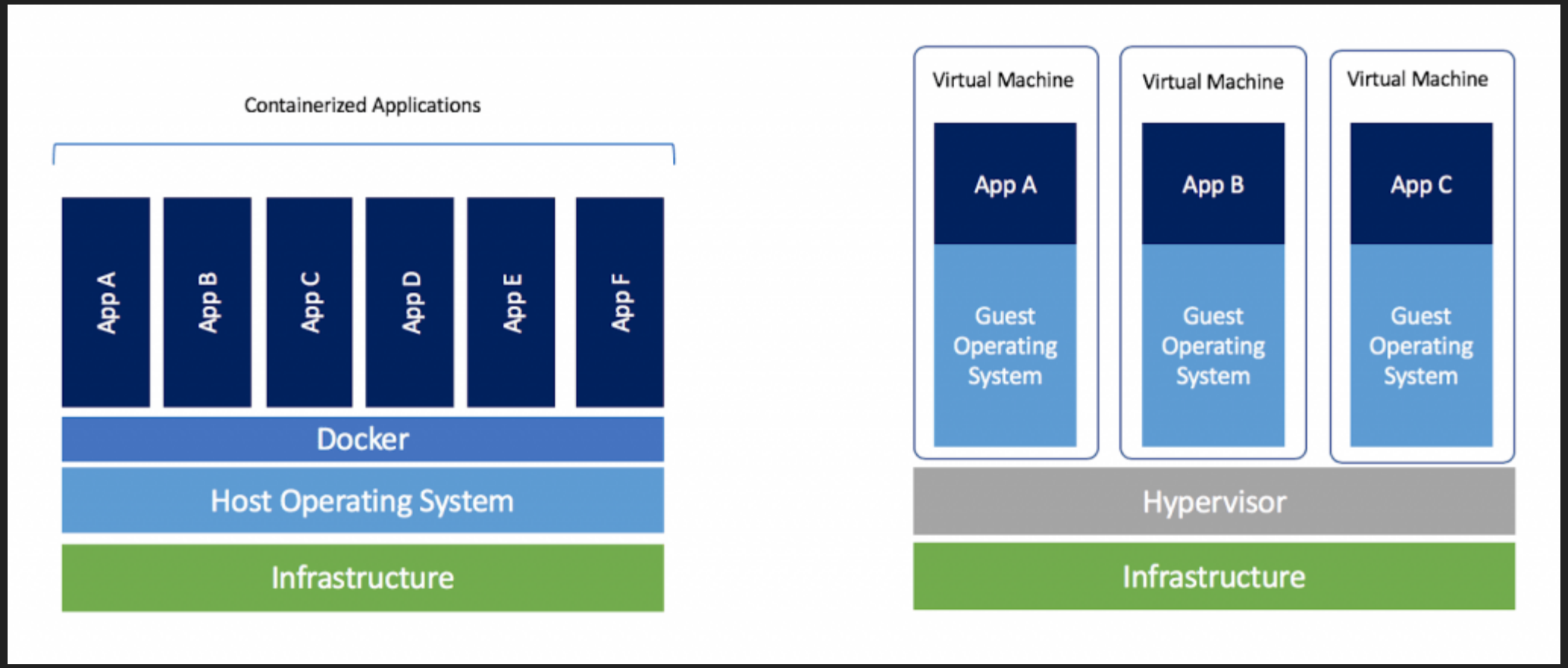
Docker

BY ANKIT MISTRY

Docker

- Docker is software development platform
- Here you packaged app in images
- Container use image to start application
- Containers run on any operating system
- It works exactly same independent of OS, machine, Environment
- Lightweight compared to VM
- Easier to maintain & deploy
- Docker works with any language, runtime, OS

Docker vs VM



Docker workflow

Docker Installation

Dockerfile

Docker Image

Container Registry

AWS ECR

Docker Container

Create Simple Webapp

- Python based Web Application
  - main.py
  - Dockerfile
- Build Docker images
- Push to Container Registry

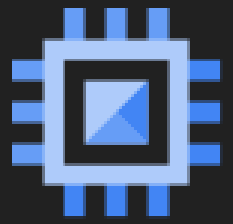
Deploy App

BY ANKIT MISTRY

Two thing to consider

- Where you want to deploy
- What are deployment strategy

Compute Options



Compute Engine

Kubernetes

App Engine

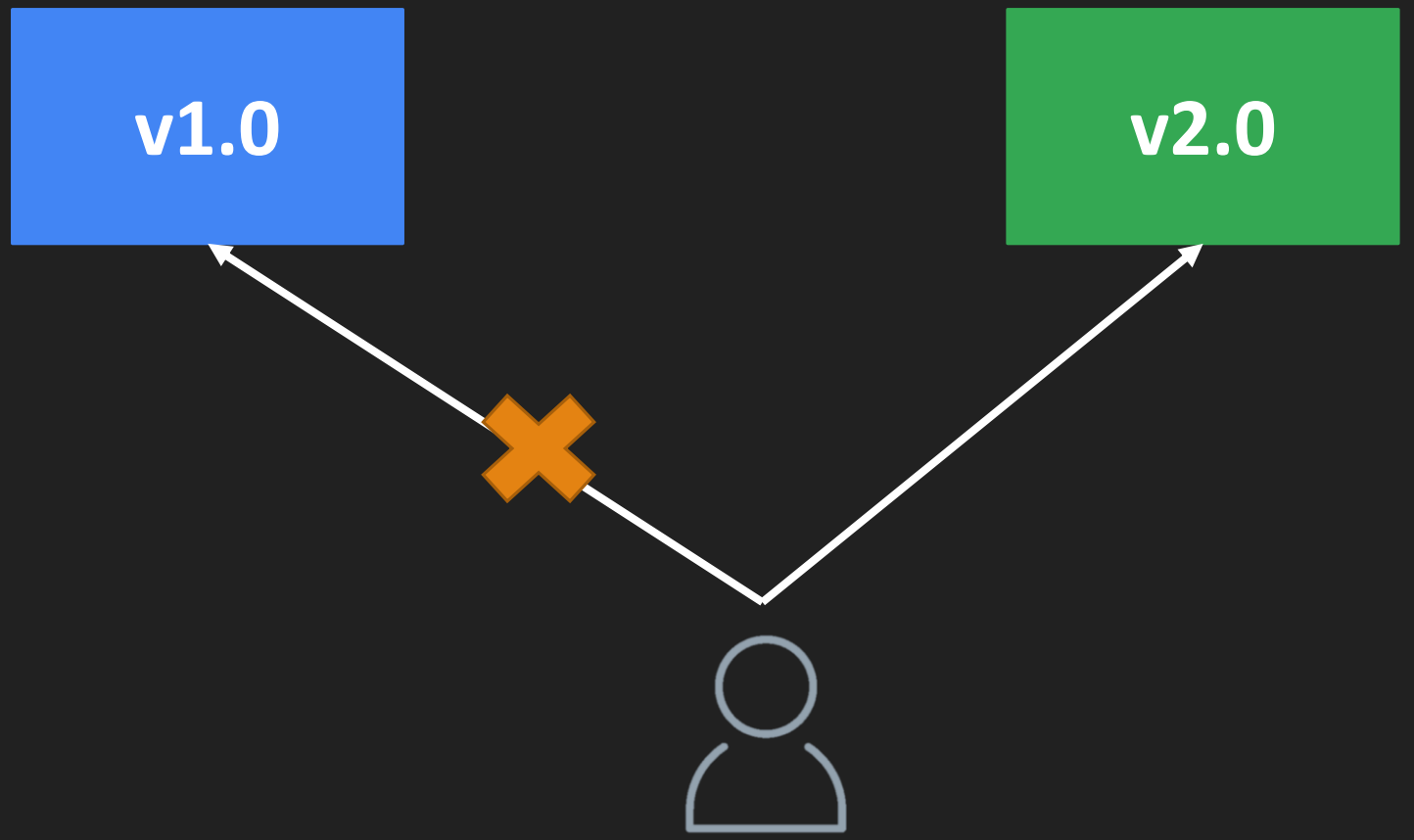
Cloud Run

Cloud Function

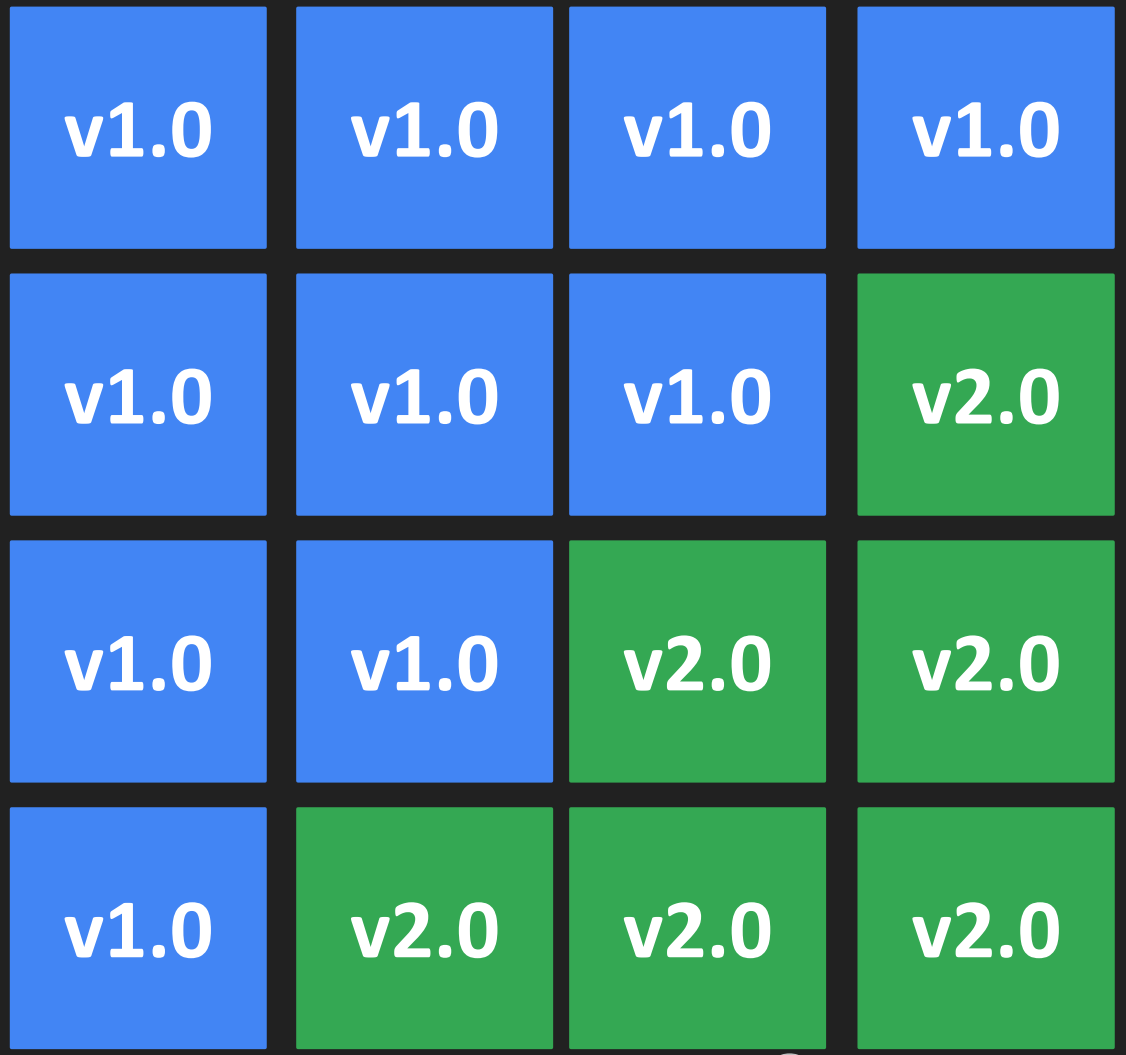
Deployment methods

- Blue/green Deployment
- Rolling Deployment
- Canary Deployment
- Traffic splitting Deployment

Blue/green Deployment



Rolling Deployment



Canary Deployment

v1.0	v1.0	v1.0	v1.0	v1.0	v1.0	v1.0	v1.0
v1.0	v1.0	v1.0	v1.0	v1.0	v2.0	v2.0	v2.0
v2.0	v2.0	v2.0	v2.0	v2.0	v2.0	v2.0	v2.0

Traffic Splitting

- Small Percentage of user will be served new version (ex : 10-20%)
- If everything is fine, Redirect all user to new version.
- Traffic splitting can be used for A/B Testing.

Deploy Cloud Functions

BY ANKIT MISTRY

Deploy to App Engine

BY ANKIT MISTRY

Deploy to Cloud Run

BY ANKIT MISTRY

Deploy to Kubernetes

BY ANKIT MISTRY

Deploy to Compute Engine

- IAAS – Infrastructure as a service
- General Purpose computing machine
- 2 ways Deployment
  - Containerized App
    - Via Container optimized OS
    - Via Other OS + manual Docker installation
  - Non Containerized App
    - Manual install apache
    - Install Via startup script

Deploy to Instance Group

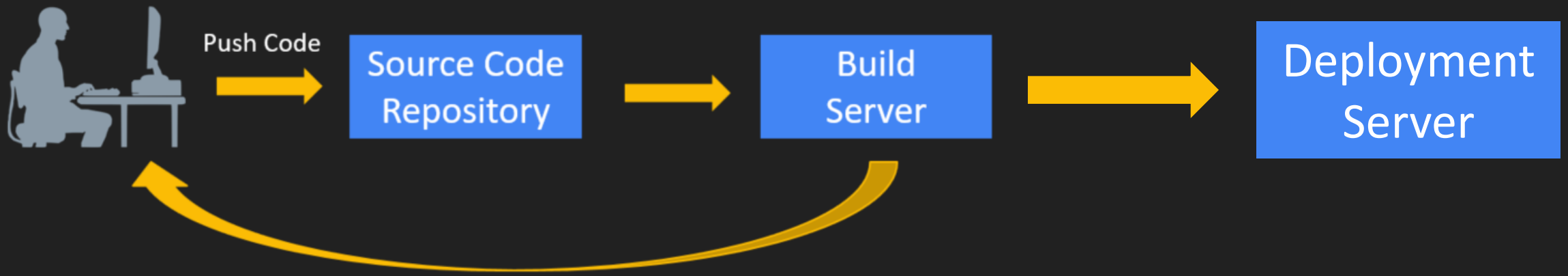
- Instance Template
  - Blue print for all Virtual machine
- Create Instance from template
- Instance group
  - managed
  - unmanaged
- Load balancer

CI/CD Pipeline

BY ANKIT MISTRY

Continuous integration

Continuous Deployment



Continuous Deployment vs Continuous Delivery

- Continuous Deployment
  - Fully automated, no manual intervention
  - Code is continuously build & deploy
- Continuous Delivery
  - Release to Production
  - May involve manual approval
  - It will make sure delivery are often & fast
  - Before Continuous Delivery, frequency of release usually one in 3-month
  - Now, Possible to release 5 times in day

Different Services for CI/CD

BY ANKIT MISTRY

Source Code management

Bitbucket

mercurial

Source Code Management

GitHub

Cloud Source Repository

Build

Jenkins

Build

Teamcity

circleci

Cloud Build

Artifact Storage

Artifact Storage

Container Registry

Artifact Registry

Deployment

Deployment

Compute Engine

Kubernetes

App Engine

Cloud Run

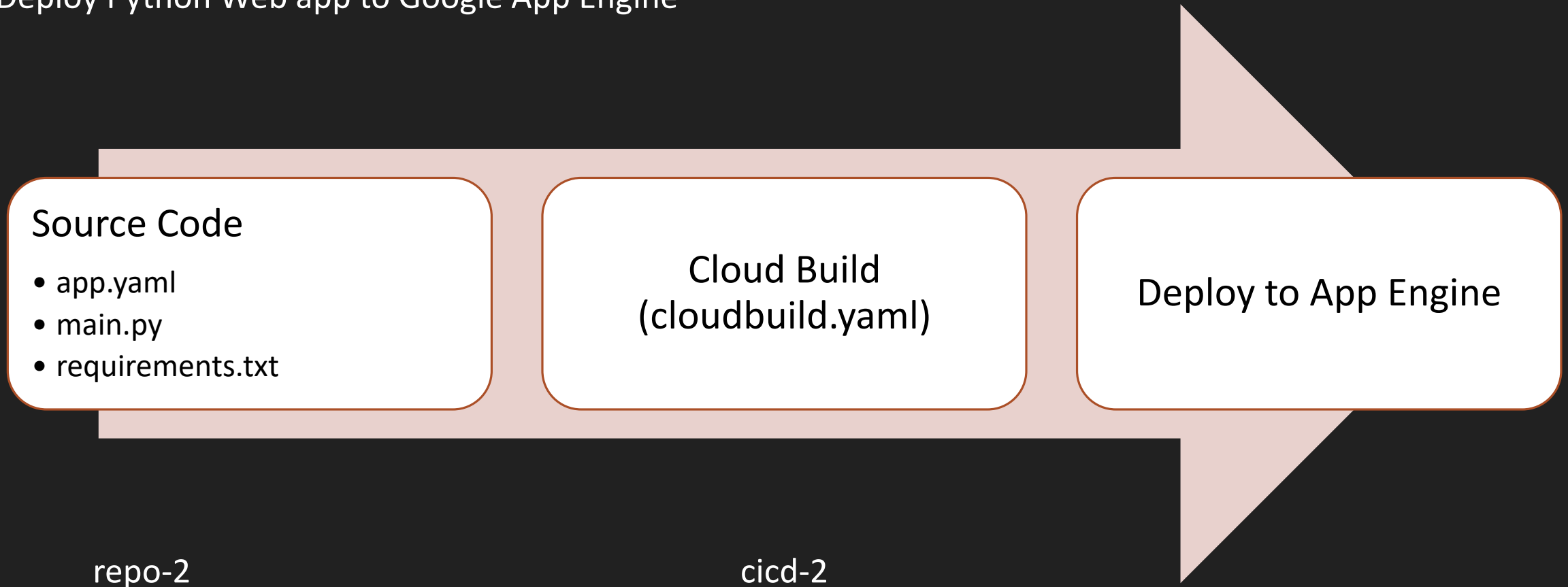
Cloud Function

CI/CD Pipeline - 1

Create Docker Image & Push to Container Registry

CI/CD Pipeline - 2

Deploy Python Web app to Google App Engine

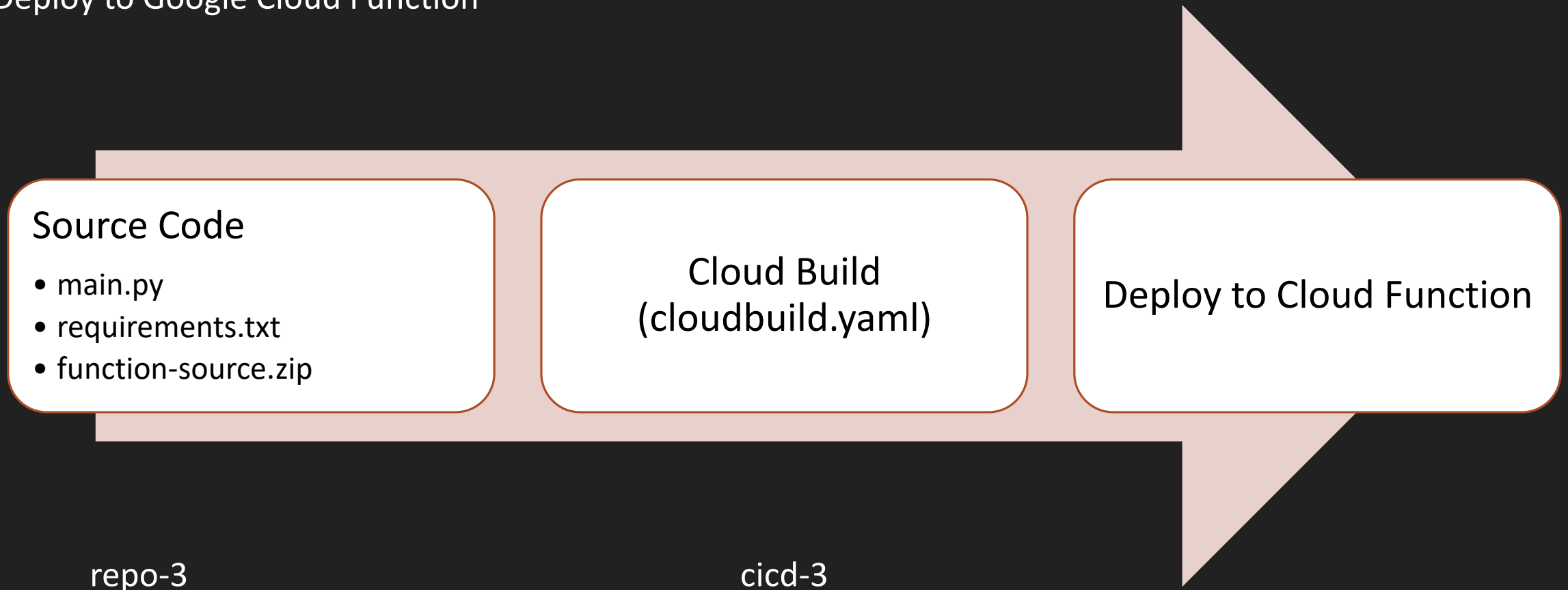


repo-2

cicd-2

CI/CD Pipeline - 3

Deploy to Google Cloud Function

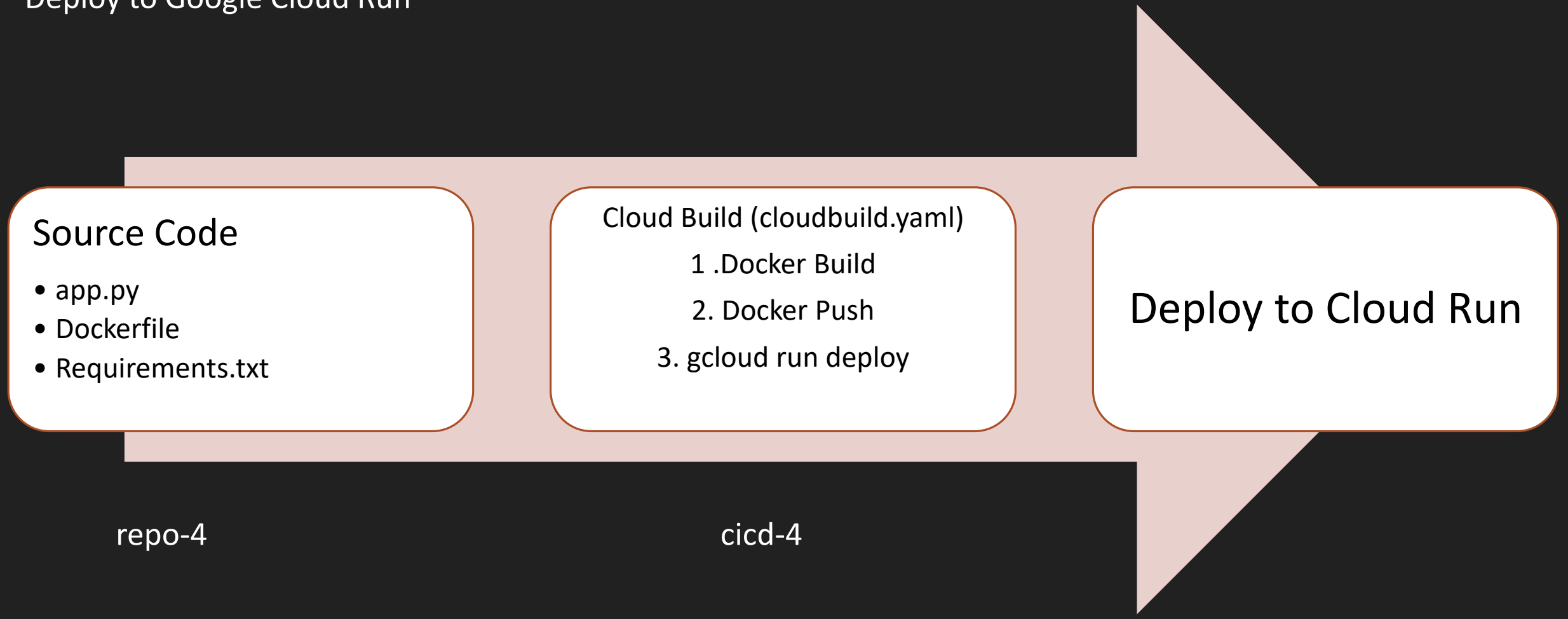


repo-3

cicd-3

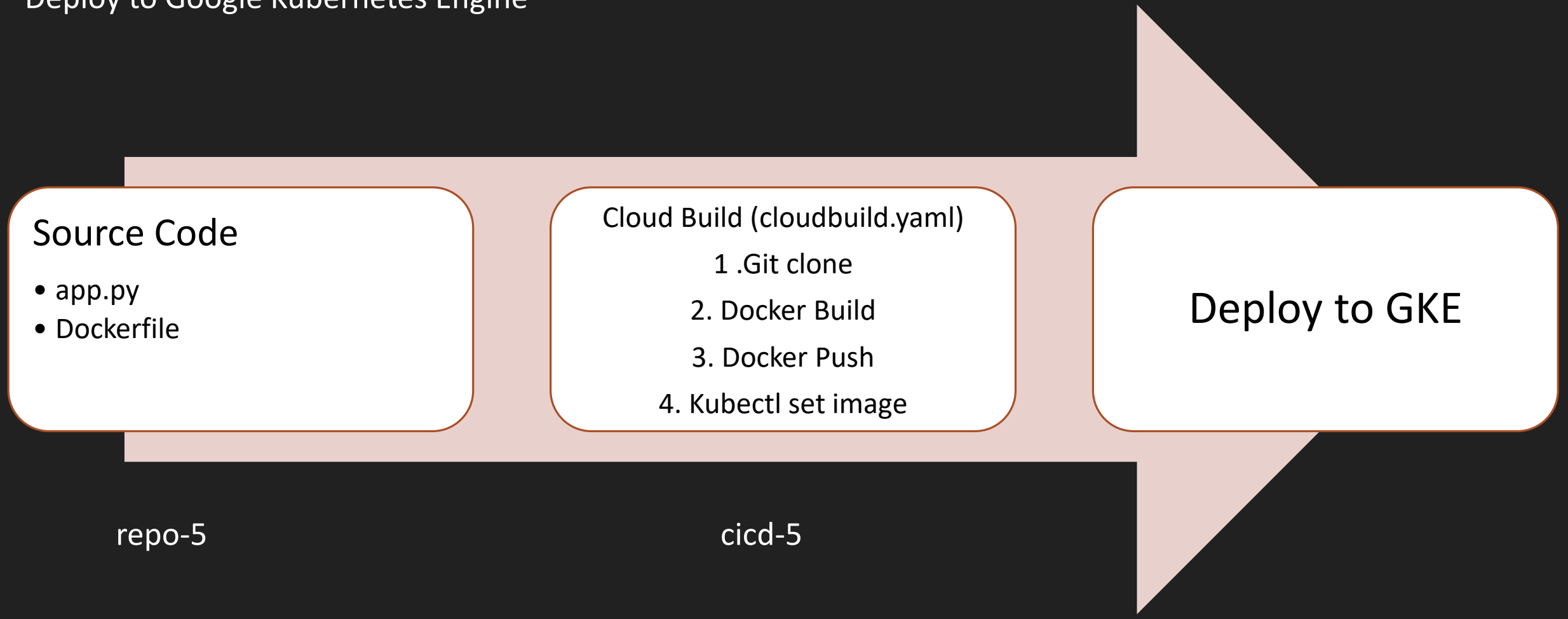
CI/CD Pipeline - 4

Deploy to Google Cloud Run



CI/CD Pipeline - 5

Deploy to Google Kubernetes Engine



GKE CI/CD Steps

- Create GKE Cluster
- Deploy some sample Docker Images
- Create Load balancer based service
- Create Source Code Repos
- Add Python code, Docker file, cloudbuild.yaml
- cloudbuild.yaml
  - clone repo
  - Build Image
  - Push image
  - Update new image
- Create Cloud Build Trigger

Jenkins

- Popular Open source tool for CI/CD
- Alternative to Cloud Build
- Jenkins can be extended with plugins
  - Source code - git
  - Unit testing – junit
  - GCloud SDK
- Installation
  - VM + manually install Jenkins
  - marketplace solution (Preferred)

IAC - Infrastructure as a code

BY ANKIT MISTRY

IAC

- Infrastructure as a code
- Process of managing and provisioning cloud resources with some descriptive language
- Create Shell/Python script for creating VM
- But writing/maintaining such code is tedious task
- Need better language to create resource

```
Create N/W
Wait for above step to finish
Provision Subnet
Create Firewall rule
Wait for above step to finish
Compute engine instance with all parameter
```

```
resource "google_compute_instance"
"first-instance"{
    name = "hello-1"
    zone = "us-central1-a"
    machine_type = "n1-standard-1"
    boot_disk {
        initialize_params {
            image = "debian-
cloud/debian-9"
        }
    }
    network_interface {
        network = "default"
    }
}
```

Tool for IAC

- Cloud Native tool available for infrastructure provisioning
- Azure – Template
- Google – Deployment manager
- AWS - Cloud Formation
- JSON/YAML
- Terraform is cloud agnostic.
- With Multiple provider, resource can be provisioned for multiple cloud.

Terraform

- Terraform is the one of the most popular tool for Infrastructure provisioning
- Free – Open source
- Developed by HashiCorp
- Quick & easy to get started with single binary file
- Master HCL – terraform in short span of time
- Terraform has multiple provider are available.
- Apart from Public cloud, lots pf different other provider are available for network, DNS, Firewall, database
- Write configuration in HCL/JSON.
  - HCL is preferred.
- Terraform is agentless tool
- It is not configuration tool. Work well with Ansible.

Terraform Installation

BY ANKIT MISTRY

Terraform – Create VM

BY ANKIT MISTRY

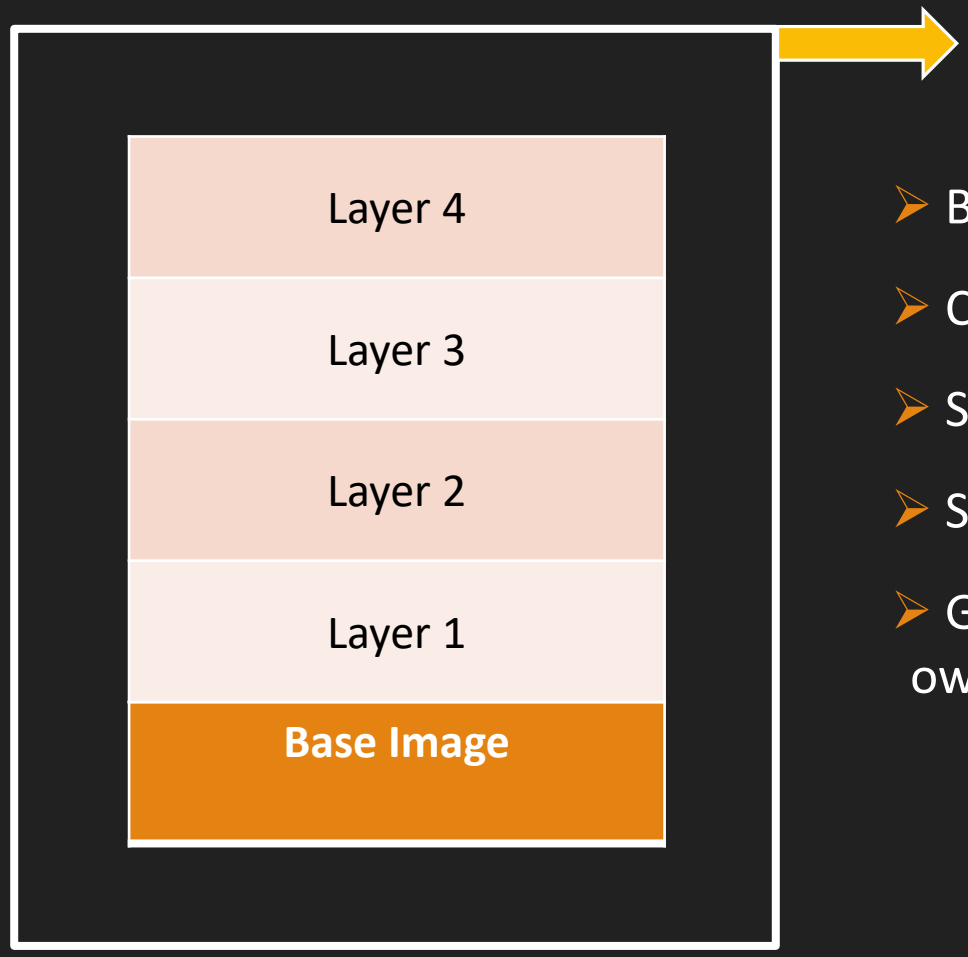
Secure Container Deployment

BY ANKIT MISTRY

Secure Container Deployment

- Google managed base images
- Container analysis
- Binary authorization

Secure Base Image



- Base image is made of Ubuntu/debian based OS
- Choosing right base image is important
- So, How to pick right base image
- Solution is : Google marketplace
- Google maintained these images & deploy for their own app deployment.

Container Scanning

- Container Analysis provides
  - automated vulnerability scanning
  - manual vulnerability scanning
- For containers in Artifact Registry and Container Registry
- Works exactly same for both Registry
- Manual
  - gcloud artifacts docker images scan imageurl –remote
- Automate
  - Let’s see in action

Scanning & Base image Demo

BY ANKIT MISTRY

Binary authorization

- How to Prevent from deployment
- Binary authorization
- Binary Authorization is a deploy-time security control
- It ensures only trusted container images are deployed on Google Kubernetes Engine (GKE) or Cloud Run
- Let’s see in action
  - For Cloud Run
  - For GKE

Cloud Operation Tool

BY ANKIT MISTRY

Operation Tool

- Operation like Monitoring, Logging
- Why Logging – Monitoring is required
- What is Logging
- Kinds of Log – Audit Logs
- Log Collection
- Log Routing
- Log Export
- Cloud Monitoring – Metrics, Dashboard, Uptime check, Alerts
- Cloud Debugger, Trace, Profiler, Error Reporting

Why such tool

- Software Development + Maintenance
- Everyone want their software run smoothly
- But No software is bug free
- issues come at dev stage, Test or Prod level
- How to find root cause behind it
- You need to continuously monitor resources
  - Space is sufficient
  - Is application is slow
  - Is CPU usage going beyond 90%
  - Who did What with Prod (even if by mistake)
- So to know all those answer & many more, such tool is required

Cloud Monitoring

- Monitor various cloud Resources
- Different Metrics can be measured
- Monitor one or more GCP Project or AWS Account
- Workspace
  - Multiple metrics can be added
- Default workspace & custom workspace
- Let’s see in action – Monitoring UI

Cloud Logging

- Log management tool
- Fully managed service
- Store Exabyte scale data
- Log can collected from multiple source
- Search & analyze log
- Let’s explore Logging UI
  - Logs Explorer, Dashboard, Log Metrics, Logs Router

Types of Cloud Audit Logging

who did what, when, where

Admin activity

- By Default Enabled
- Administrative action
- 400 days
- Free
- Create VM, Delete VM
- Can not Configure, Can not Disable

System Event

- By Default Enabled
- Generated by Google System
- 400 days
- Free
- VM Migration, Preemtive VM
- Can not Configure, Can not Disable

Data Access

- By Default **Not** Enabled
- Create, modify Resource Data
- 30 days
- **Not** Free
- Create Object in Bucket
- Can be disable

Policy Denied

- By Default Enabled
- Google Service denies access
- 30 days
- **Not** Free
- Security violation
- Can not be disabled. But can be excluded with Filters

[Hands-on] Cloud audit Logging

BY ANKIT MISTRY

Explore Audit Log Structure

BY ANKIT MISTRY

Log Collection

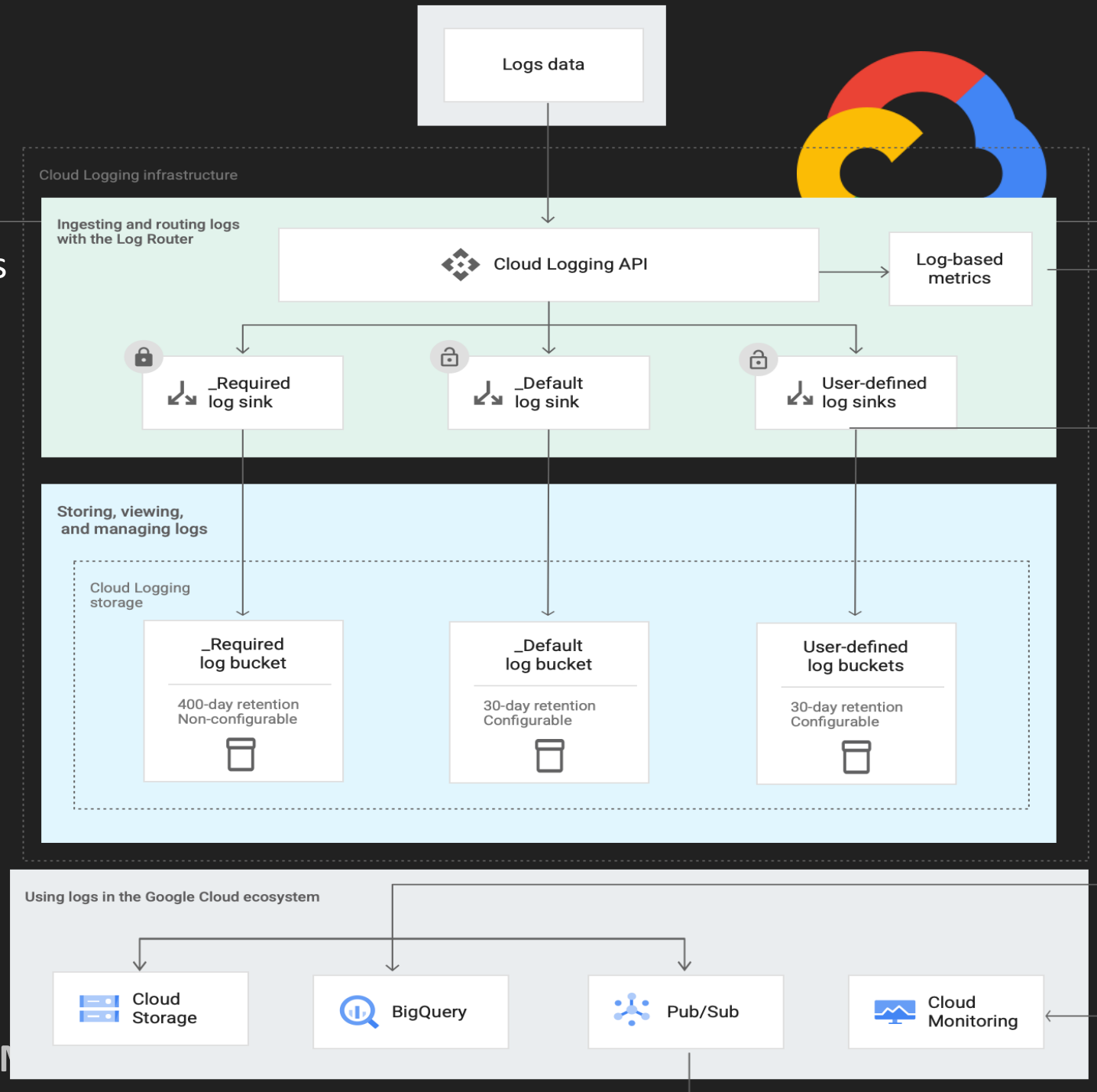
- Log read/write via gcloud SDK
- Automatically
  - Cloud Run, GKE, App Engine
- Logging Agent
  - For Compute Engine on Google cloud / AWS VM
  - Legacy agent/ Ops agent
- Cloud Logging API
  - Python/Java SDK
  - From On-premises

[Hands-on] Log based metrics

BY ANKIT MISTRY

Log Router

- Log arrives at Log Router from various sources
- From Router, diverted to various sink
- Two types of Log Bucket
  - _Required
  - _Default
- Logs can be routed to User defined Bucket
- Sinks
  - BigQuery
  - Cloud Storage
  - PubSub



More Ops/Dev Tool

- Cloud error reporting – detect error
- Cloud Debugger – Find state of running application
- Cloud Trace - latency
- Cloud Profiler – How much resource consumed

Optimize resource utilization

- Resource cost, utilization levels, Billing
- Pre-emptible VMs
- Committed use discounts [CUD], sustained use discounts[SUD]
- TCO considerations

Preemptible VM

- Just like Other virtual machine
- Short lived cheaper virtual machine
- Provision Pre-emptible VM When
  - Workload is fault tolerant
  - Not require 100% high availability
  - Cost is critical
- up to 80% discount
- max life is 24 hours
- Not always available
- Google give you 30 sec warning before auto shutdown
  - Regular VM has higher priority than Preemptible VM
- Let’s see how to configure it

Flat-rate, committed use discounts[CUD], sustained use discounts[SUD]

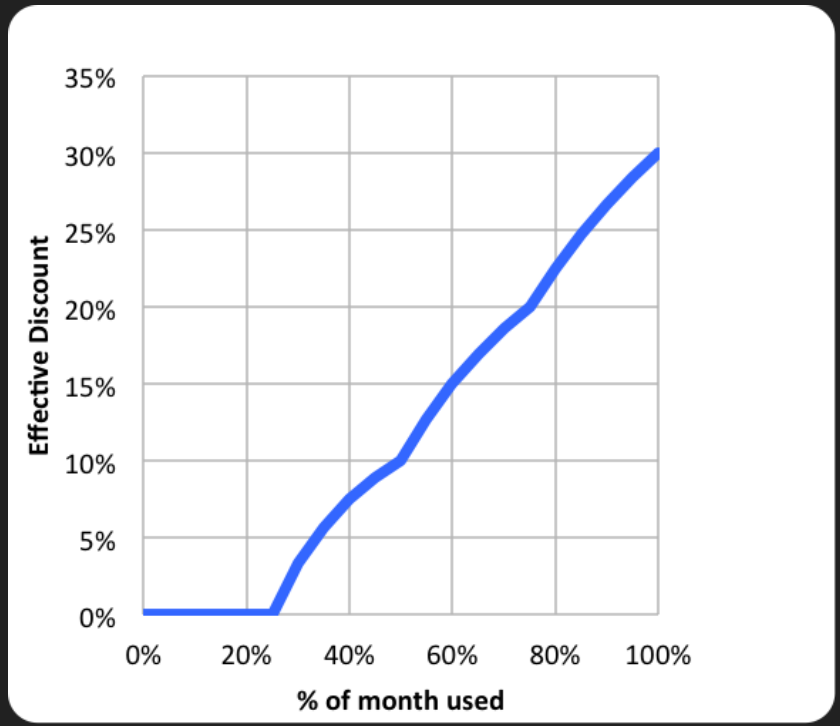
BY ANKIT MISTRY

Flat Rate

- Pay for what you use
- No Special Discount
- In Compute Engine :
  - E2 and A2 category of Machine

Sustained use discounts[CUD]

- Sustained use discounts are automatic discounts for running specific Compute Engine resources a significant portion of the billing month
- Applies to N1, N2 machine types
  - Not applicable to other machine type
- If you use at least 25% of month
- Only on GKE & VM Instances
- Let's see in action



Committed use discounts[CUD]

- Let’s say your workload is predictable
- you can commit for 1 year or 3 year
- Get up to 70% of discount.
- Only on GKE & VM Instances
- Can not cancel commitments
- Let’s see in action

Total cost of operations (TCO)

- TCO = Purchase Cost of Asset + Cost of operation
- When moving to Cloud from on Premises
  - Cost need to consider
  - In GCP, No purchase of asset
  - Provision Resources with no minimum commitment (Expect few service feature)
  - Cost include (Pay as you go model)
  - Operation Cost

THANK YOU