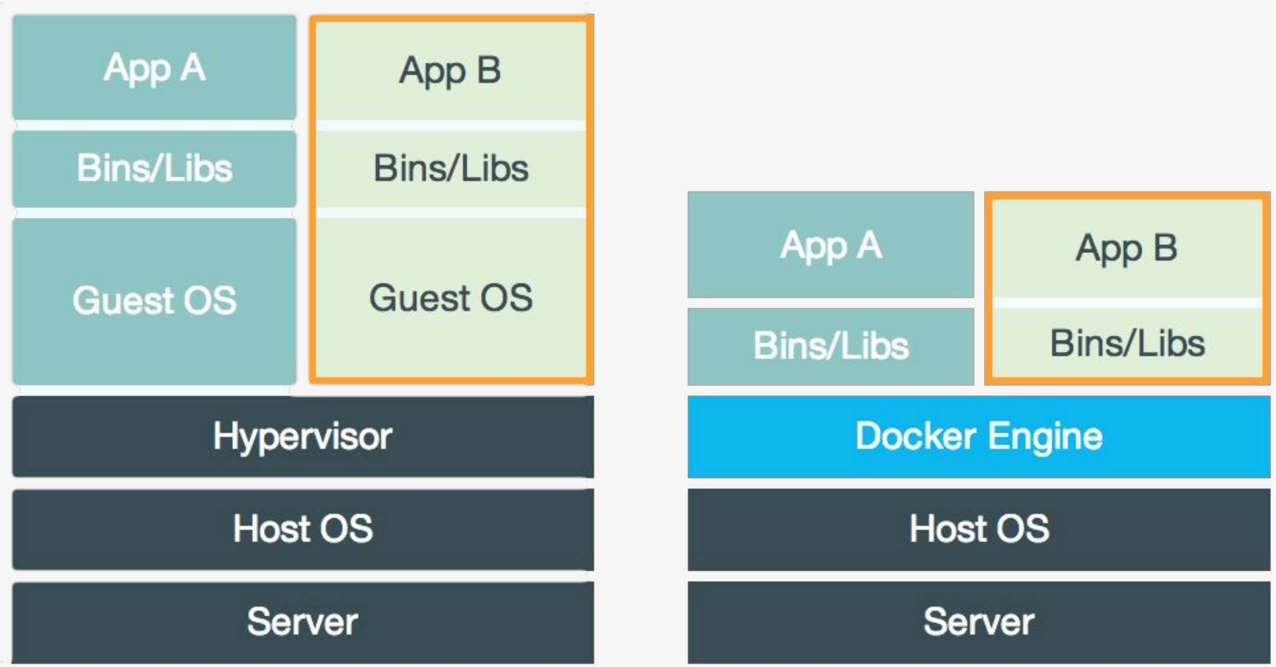
GOOGLE KUBERNETES ENGINE

What is Docker?

- Open Source Project
- It is a tool that packages up an application and all its dependencies in a "virtual container" so that it can be run on any Linux system or distribution.
- Does Docker run only Linux? Docker can "emulate" Linux within it's container space, but client to do so can be install on Linux, Windows and Mac OSX system.

Virtual Machine vs Containers

Container is simply the encapsulation process on the underlying system.



Docker Architecture

- Docker is a client-server application where both the daemon and client can be run on the same system or you can connect a Docker client with a remote Docker daemon.
- Docker clients and daemons communicate via sockets or thru RESTful APIs
  - REST – it is a stateless transfer over HTTP of a web page containing an XML file that describes and includes the desired content. Making a HTTP call and getting info back in a formatted file in this case XML.
- The main components of Docker are
  - Daemon
  - Client
  - Docker.io Registry

Application Virtualization

- This is not something new and has been done before. Many companies have been working on concept of application virtualization.
  - FreeBSD – Jails
  - Sun Solaris – Zones
  - Google – lmctfy (Let Me Contain that for you)
  - OpenVZ

All have adopted Docker Now.

As we have application needs underweight the need of hardware which is why we need containers now.

Docker Installation

- We will be using CentOS version 7.0 (You can do it on Ubuntu or Debian – Check documentation)
- Search Packages
  - sudo yum search docker
- Create a repository to pull docker from
  - cd /etc/yum.repos.d/
  - sudo vi docker.repo

```
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/$releasever/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
```

- Update the OS repositories
  - sudo yum update
- Install Docker
  - sudo yum install docker-engine

Test Docker Installation

- Start the Docker service
  - sudo systemctl enable docker
  - sudo systemctl start docker
  - sudo systemctl status docker
- Grant Permissions to docker.sock group
  - sudo usermod username –G docker
- Test and check for any Docker images
  - docker images
- Let’s run a container
  - docker run hello-world
- Check Docker version
  - docker version

Docker Hub

- Public registry/repository that is maintained by Docker Inc. containing a large number of images that you can download and use to build containers.

  – https://hub.docker.com

Why do we need Container Orchestration?

- What to do when it fails?
- How to connect them to other containers and persistent storage?
- How do you scale running services on the containers?
- How do you balance the load after the containers?

"In simple words, The process of organising multiple containers in this manner is known as container orchestration."

Enter Kubernetes

- Open Source container manager
- Automated Deployment, scaling and management or enterprise applications
- Terminology
  - Master: Controls Kubernetes Nodes
  - Node: Machines or Instances that perform tasks and are controlled by Kubernetes master.
  - Pod: Group of 1 or more containers in a node
    - Share an IP Address, hostname, and other resources. Abstracts network and storage away from the container, resulting in easy movement
  - Replication controller: Ensures specified number of pod replicas are running at any one time across nodes
  - Kubectl – CLI tool for kubernetes

Google Kubernetes Engine

- Full managed environment for deploying containerized applications
  - Uses compute engine resources
- As a managed service – details handled by you
  - Set CPU, memory and storage requirements, and GKE will do the rest
- Self healing – resulting in high availability and reliability
- Auto Scaling – scale up and down based on demand
- No vendor lock in
- Custom OS – Container-Optimized OS
  - Docker container runtime and all Kubernetes components

When to pick Kubernetes engine?

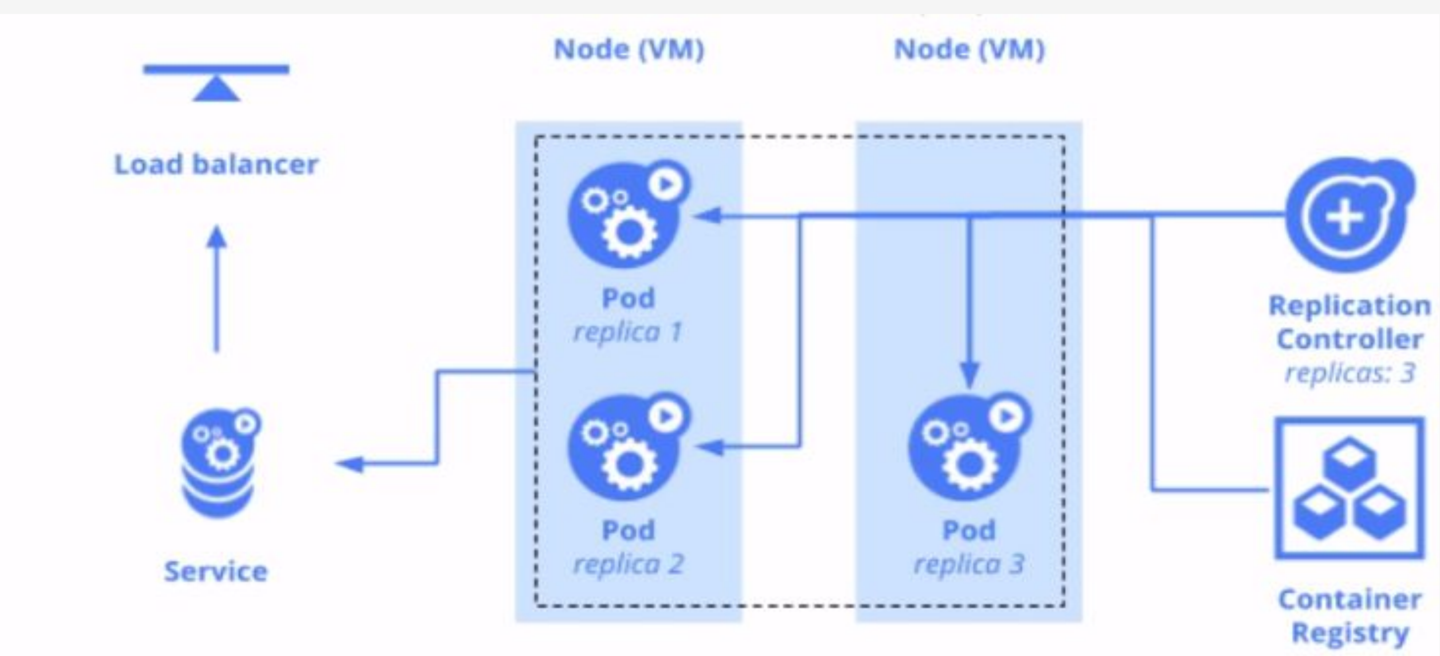
- When to choose over App Engine?
  - Hybrid or multi-cloud deployment
  - Use of protocols beyond HTTP(S)
  - Need multi-container solution – need orchestration

- When to choose GCE over GKE?
  - Needs GPU
  - Non Kubernetes container solution
  - Migration existing on premise VM to cloud
  - Custom OS or Kernel needs

GKE Components

- Container Cluster
  - Group of Compute Engine instances running Kubernetes
  - Contains 1 or more nodes instances and managed Kubernetes master endpoint
- Kubernetes Master
  - Manages the cluster, single endpoint
- Pods
  - Group of one or more containers
  - Share storage and configuration data among containers
  - Pods can contain multiple containers and multiple pods can exist of one node
- Nodes
  - Individual Compute Engine Instances
  - Run service to support Docker
  - Each node contains one or more pod

GKE Components



- Container images are grouped into Pods
- Pods are replicated across nodes
- Replication Controller both kills and duplicates pods
- Services give a single point of access, without worrying what pod is where
- Container registry has images for easy deployment

Deploy an Application

- Create a container cluster
- View cloned source code for changes
  - https://github.com/GoogleCloudPlatformTraining/cp100-bookshelf
- Cloud Shell instance – Remove code placeholders
  - Bookshelf-frontend.yaml
  - Config.py
- Cloud Shell instance – Package your app into a Docker Container
  - Docker build –t gcr.io/project-id/bookshelf .
- Cloud Shell instance – Upload the image into Container Registry
  - gcloud docker -- push gcr.io/project-id/bookshelf
- Deploy your app to the cluster
  - Kubectl create –f bookshelf-frontend.yaml