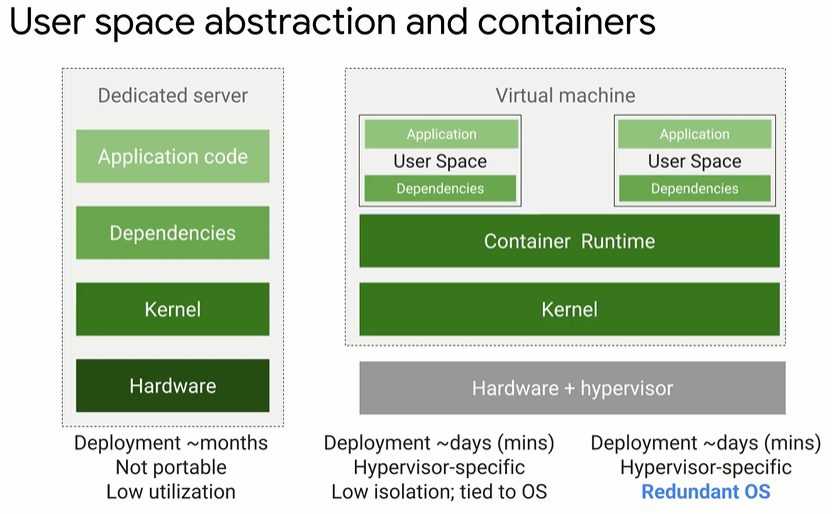
**Architecting with Google Kubernetes Engine – Foundations**

**Introduction to Containers and Kubernetes**

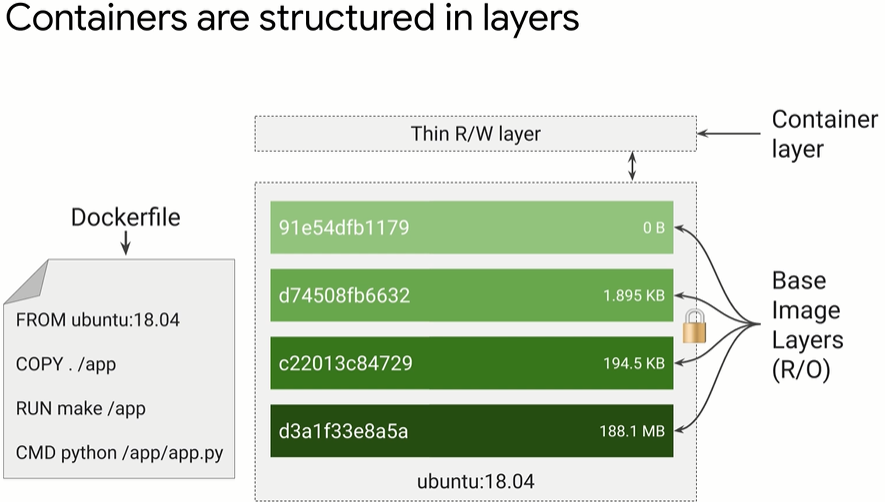
Containers



* User space is all the code that resides above the kernel includes application and dependencies
* Containers are isolated user spaces for running application code
* Lightweight (no OS)
* Easy to create and shutdown

Image – application and it’s dependencies

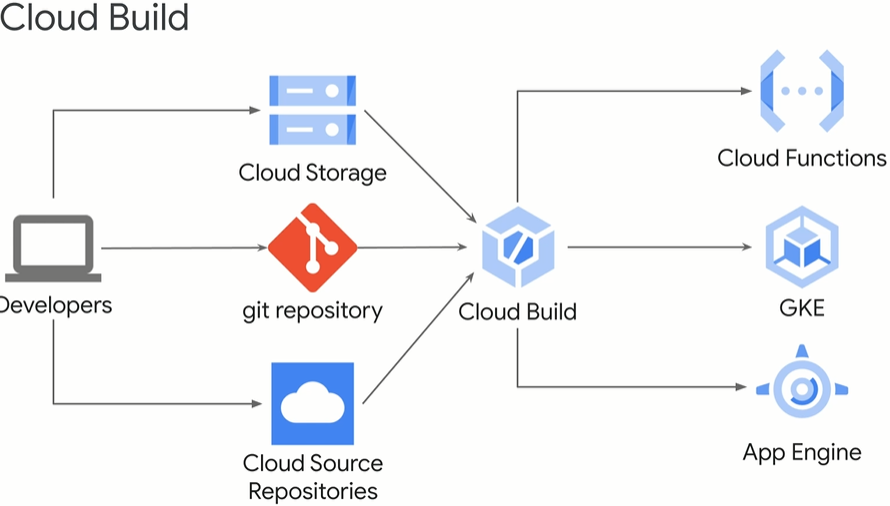
Container – running instance of an image



^each layer set of differences from the one before it. Organise layers least likely to most likely to change. The container layer is a R/W layer that is ephemeral (i.e. when container is delete the data in container layer is lost forever). As such, if you want to store data permanently must do it outside of the running container. When updating a container, it pulls down the layers it needs (i.e. only any differences)

Cloud Build

* Managed services for build containers (integrated with Cloud IAM)



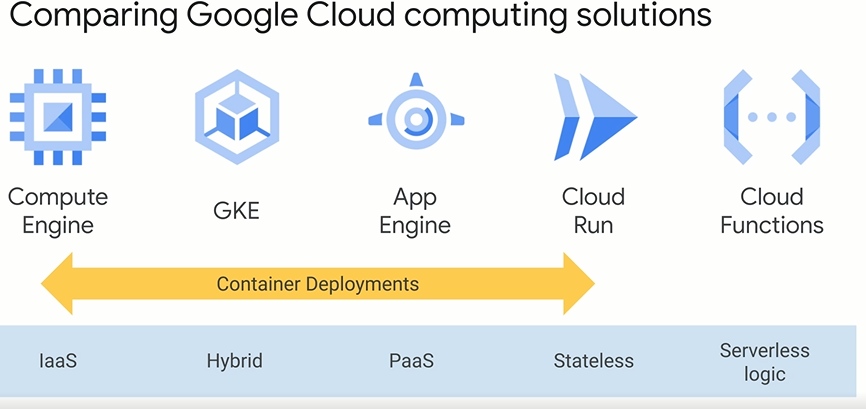
Kubernetes

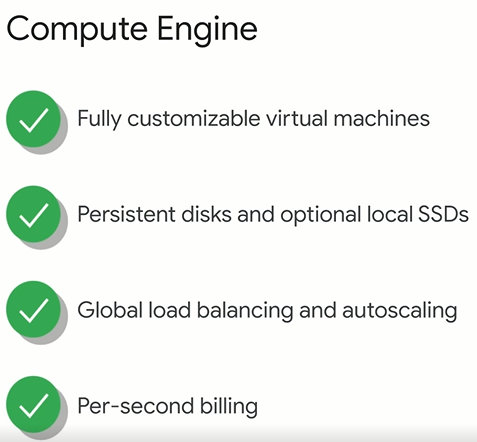
* Used to managed container infrastructure
* Open source
* Manages deployment, scaling, load balancing, logging, monitoring of containerised applications
* Declarative configuration – describe desired state rather than issue series of commands to achieve desired state
* Possible to use imperative configuration for quick temp fixes

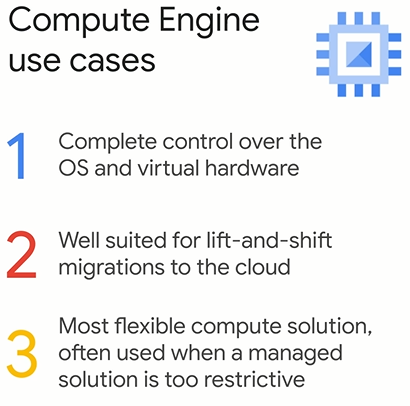
Google Kubernetes Engine (GKE)

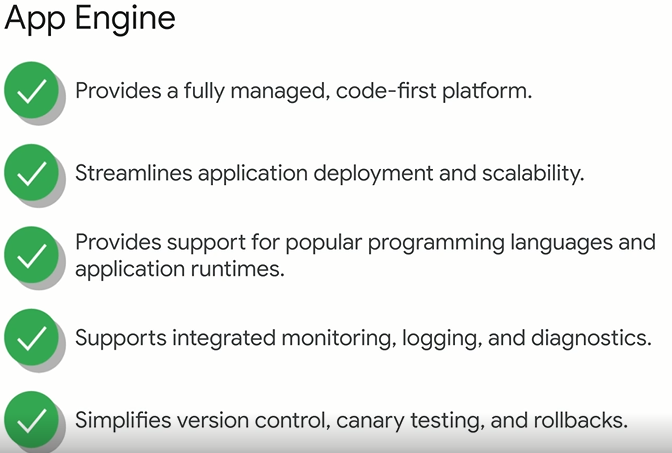
* Managed service for Kubernetes
* Fully managed
* Container optimised OS
* Auto upgrade (clusters to latest version of Kubernetes)
* Auto repair (nodes)
* Cluster scaling
* IAM
* Integrated logging and monitoring (integreates with cloud monitoring)
* Integrated networks
* Cloud console (insights into GKE cluster and resources, view inspect delete resources)

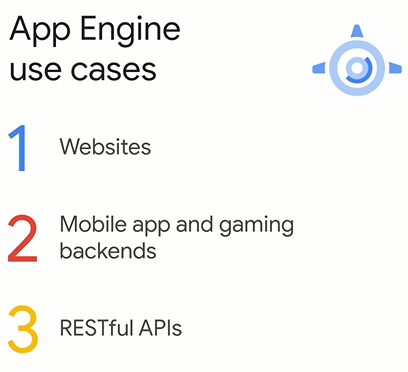
Compute Option Details

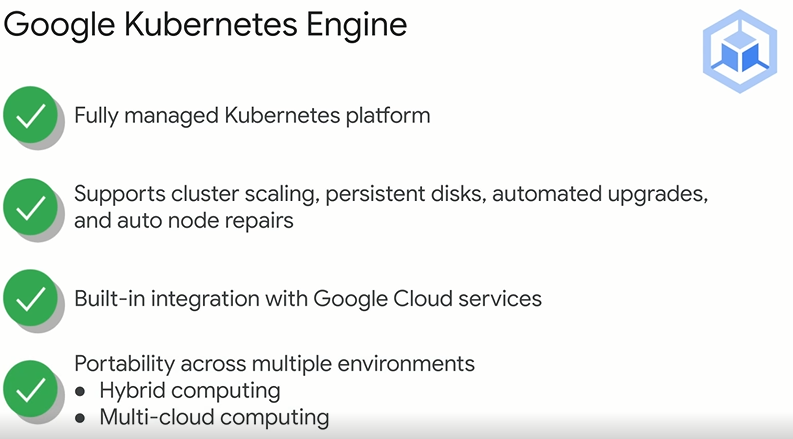


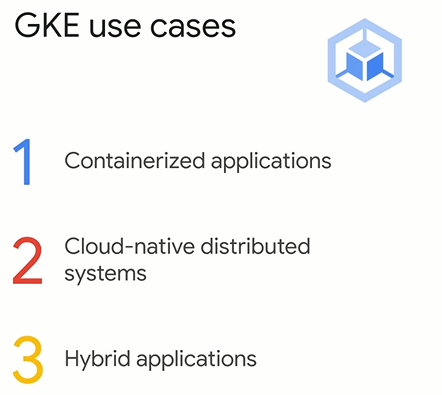


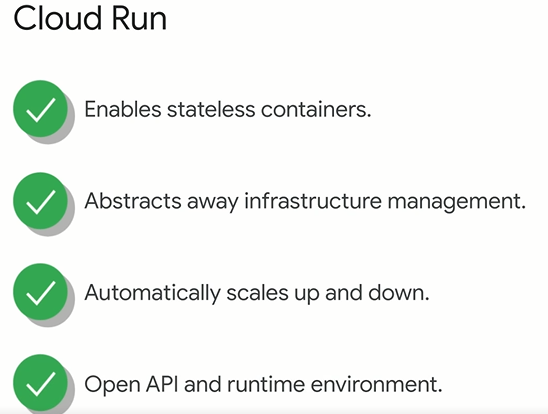


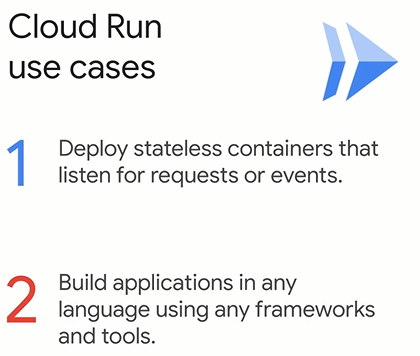


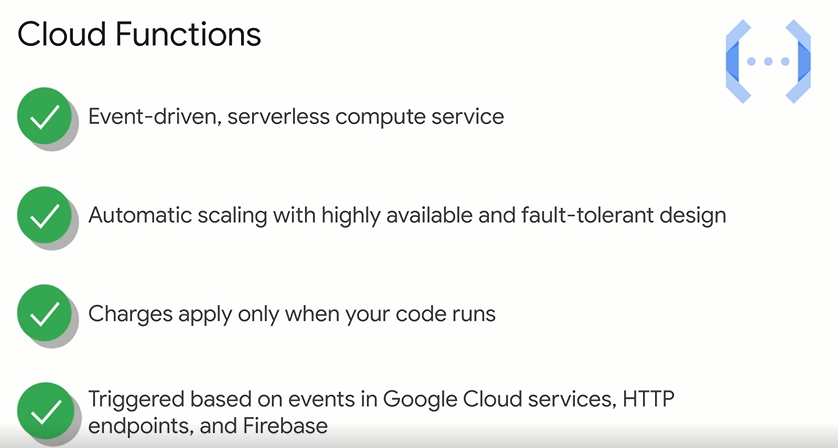


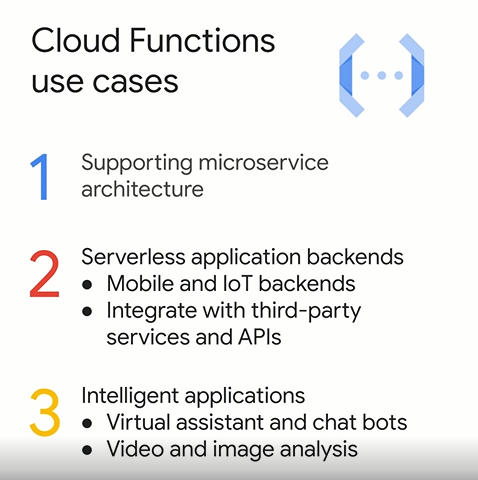


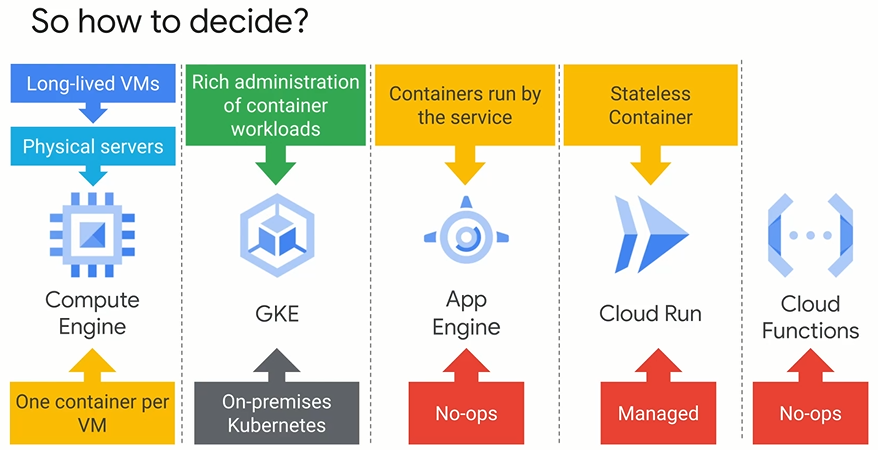






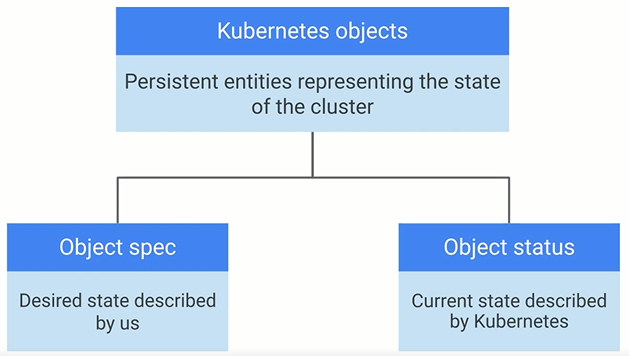




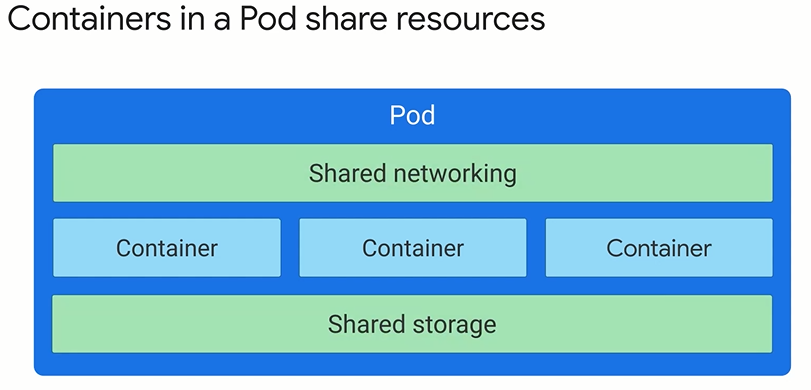


**Kubernetes Architecture**

Kubernetes Concepts



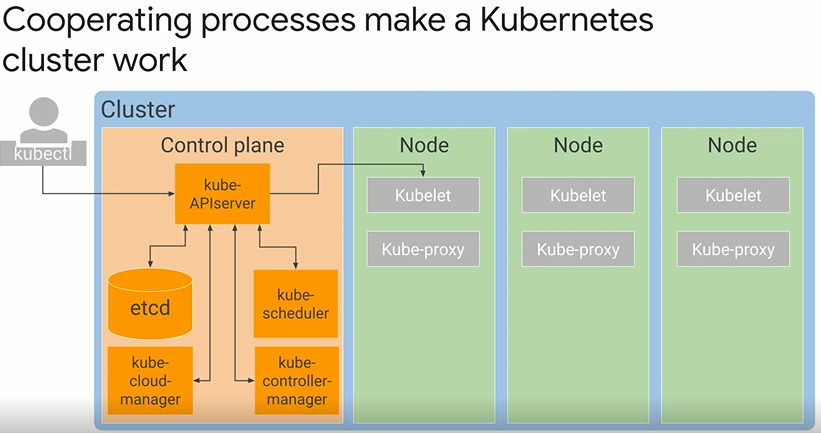
Pod is the smallest deploy Kubernetes object. Accommodates one or more containers.



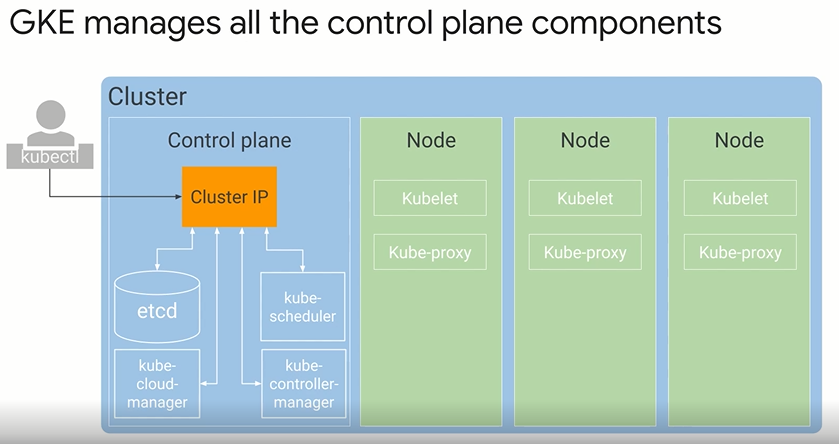
* Each pod is assigned unique IP address
* Every container within pod shares network name space
* Container in same pod can communicate using localhost

Kubernetes Control Plane

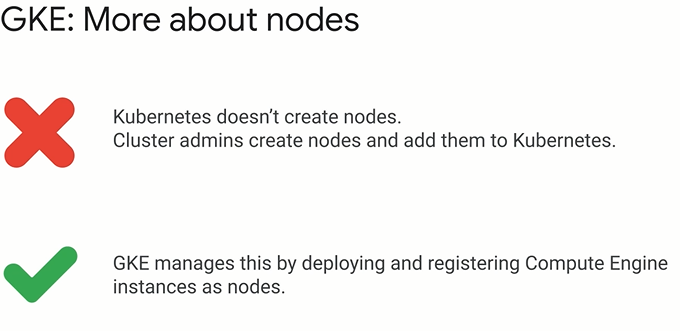
* Cooperating process that make a Kubernetes cluster work
* Kube-APIserver – accept commands that view or change state of cluster (kubectl command connects to the server)
* Etcd – cluster database, reliably stor state of the cluster
* Kube-scheduler – schedules pods onto nodes
* Kube-controller-manager – continually monitors state of cluster through kube-APIserver. When current state doesn’t match desired state, it attempts to make changes to match the desired state
* Kube-cloud-manager – manages controllers that interactive with underlying cloud providers
* Each node runs a small family of control plane components too
  + Kubelet – starts the pod and monitors its lifecycle
  + Kube-proxy – maintain network connectivity among pods in a cluster

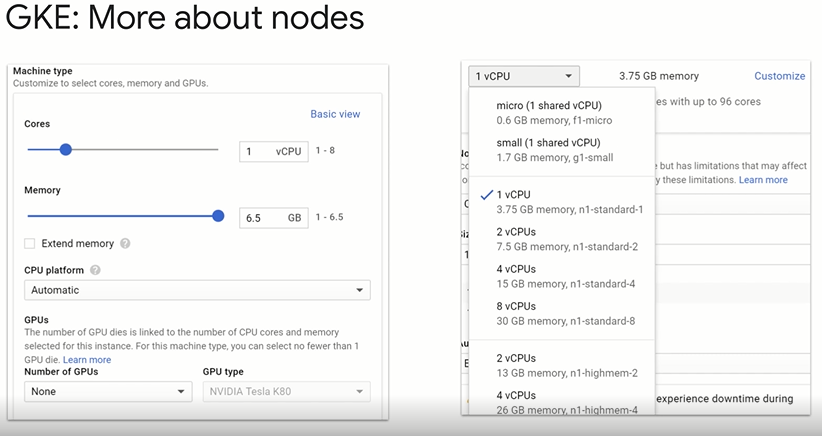


GKE Concepts

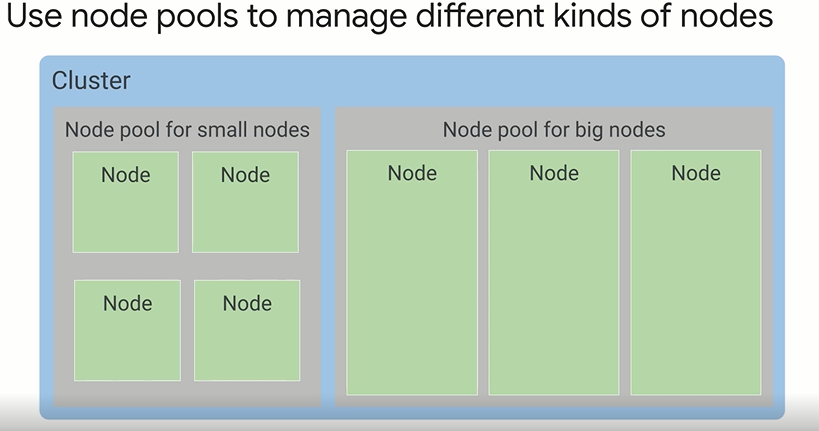


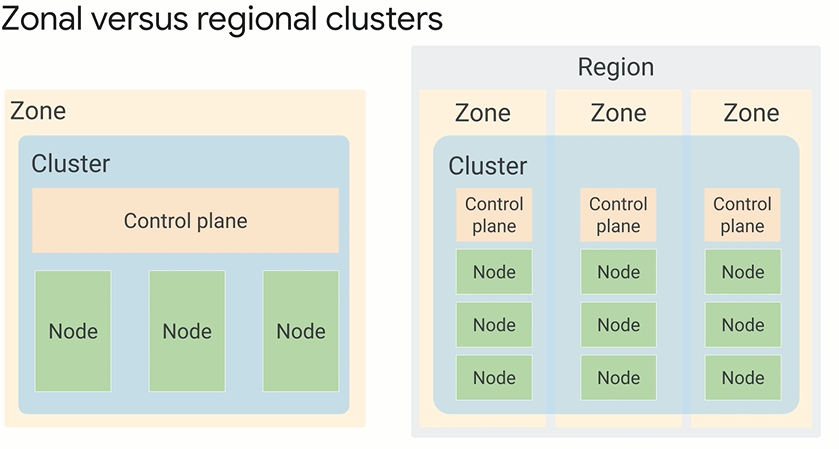
GKE takes responsibility for provisioning and managing all the control plane infrastructure

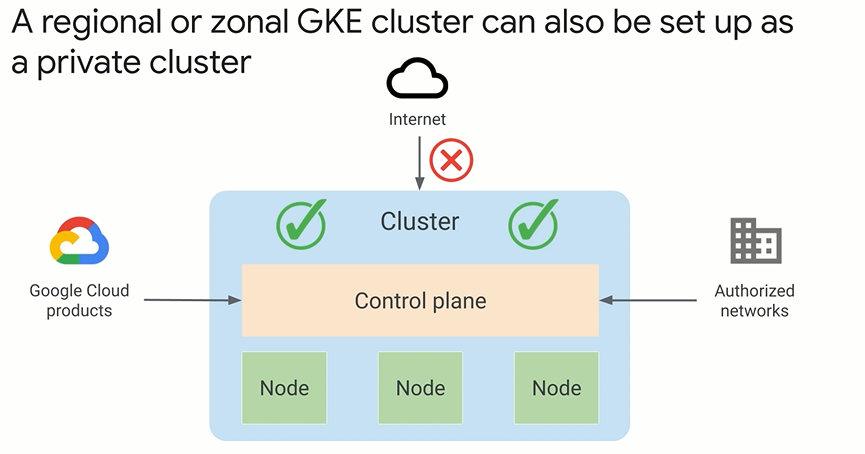




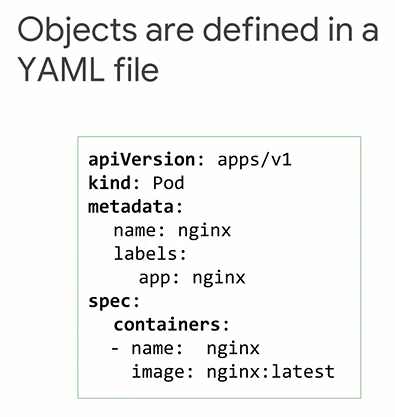
^because nodes run on compute engine you can customise node machine type when creating cluster.

^ node pool – subset of nodes within a cluster than share a configuration such as memory, CPU. Easy way to ensure workloads run on right hardware within a cluster (Only available on GKE)

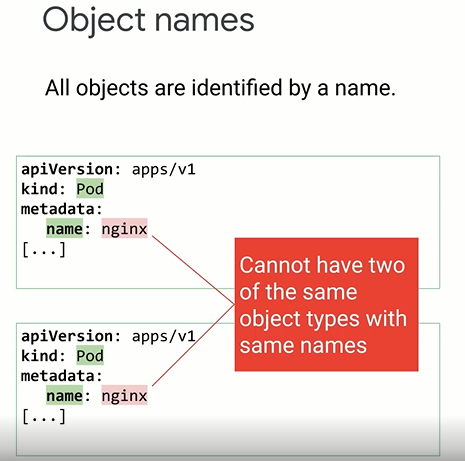
^one you choose zone or regional, it cannot be converted to the other type



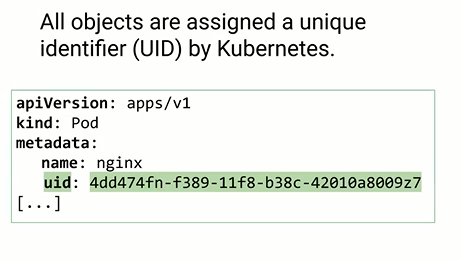
Kubernetes Object Management



^ can also be written in JSON. The above YAML file defines desired state, name and container image for pod. Best practice to use source control to manage YAML files



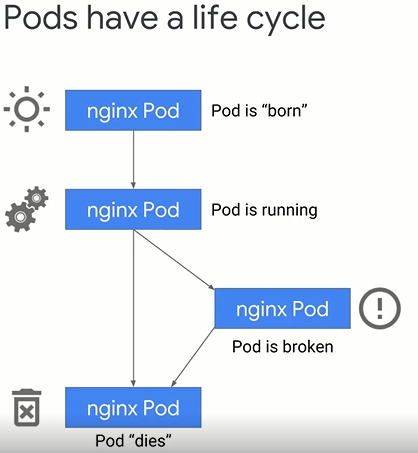
^ if object is delete its name can be resused

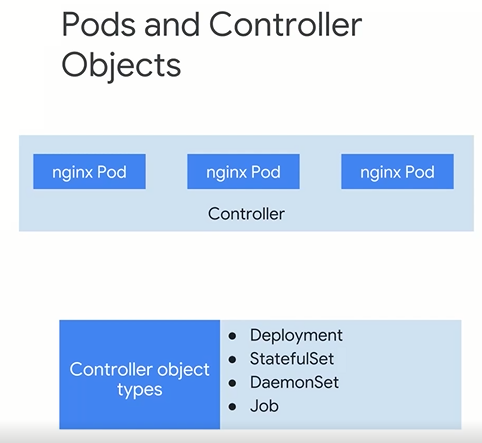




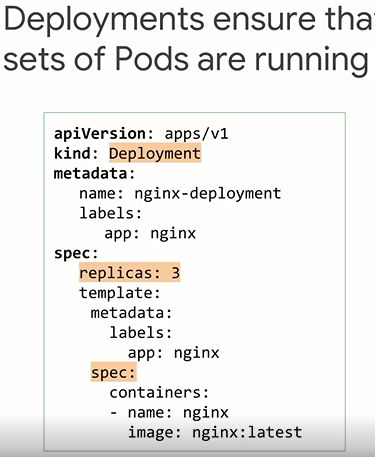
^key value pairs use to tag object during or after creation. Used to organise objects. Can use command below to filter pods



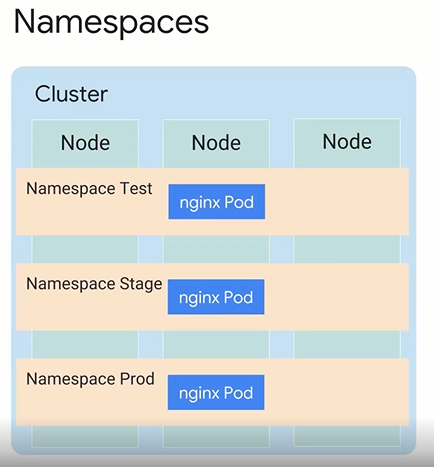




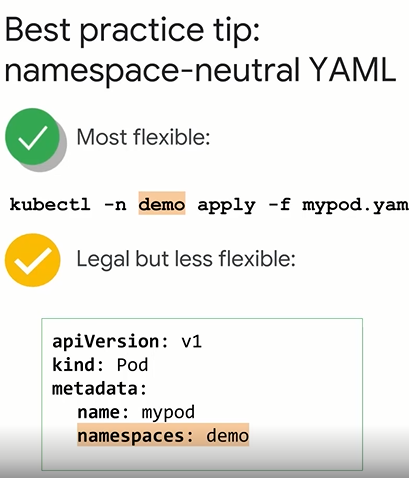
^controller object manages the state of the pod, such as deployment



^deployment ensure a define set of pods running at any given time. E.g how many replica pods. Based on this template controllers maintain the desired state within a cluster



^the name can be the same as long as they exist in different namespaces



^can specify namespace name using command line flag or in YAML. Best to do using command line to make YAML files more flexible

Migrate for Anthos

* Move existing VMs into containers

