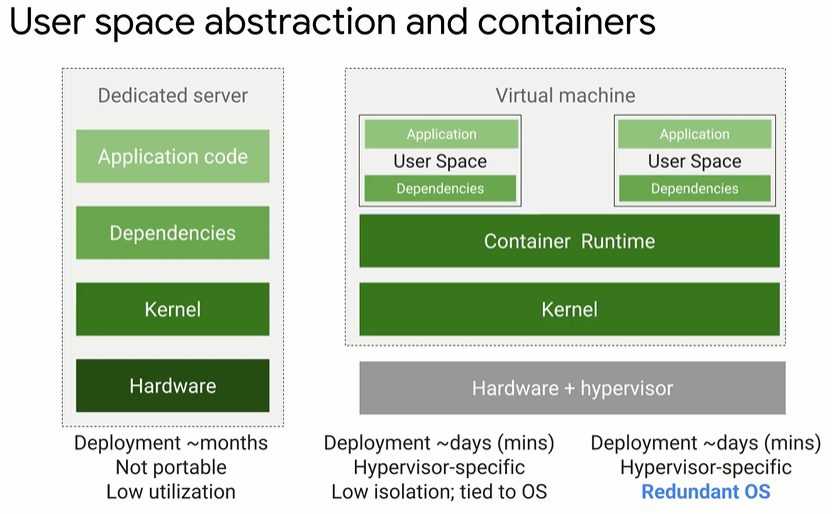
**Architecting with Google Kubernetes Engine – Foundations**

**Introduction to Containers and Kubernetes**

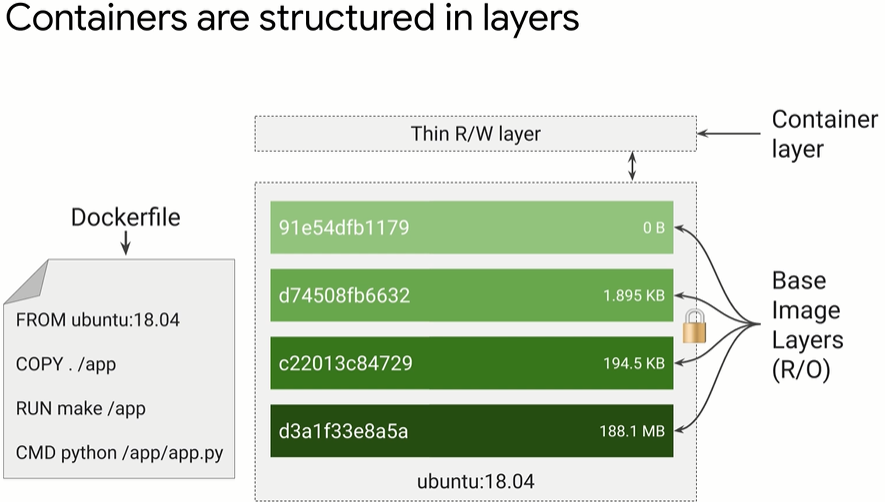
Containers



* 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

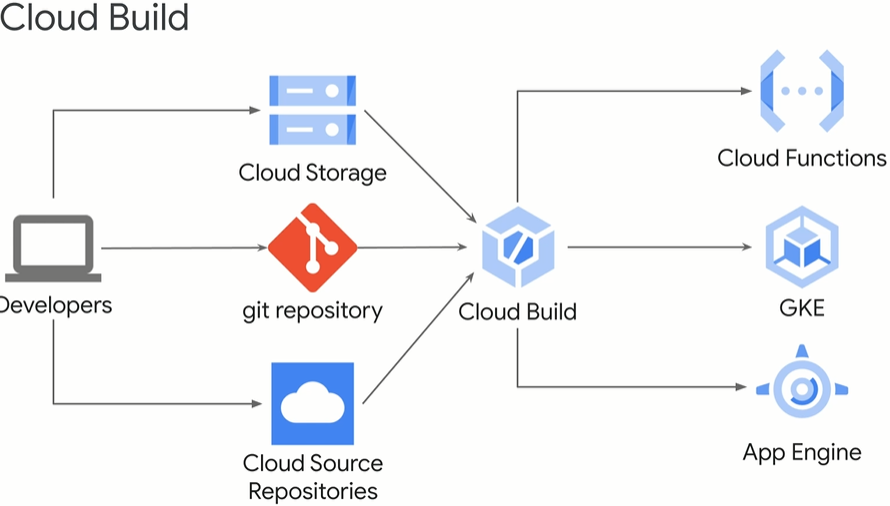
Container – running instance of an image



^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)



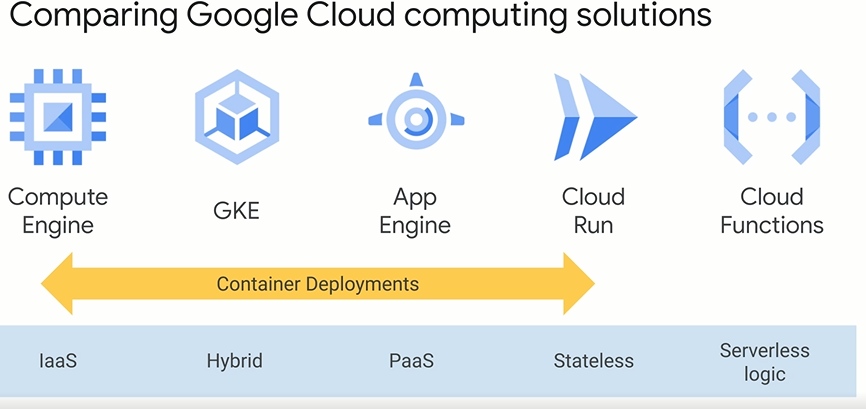
Kubernetes

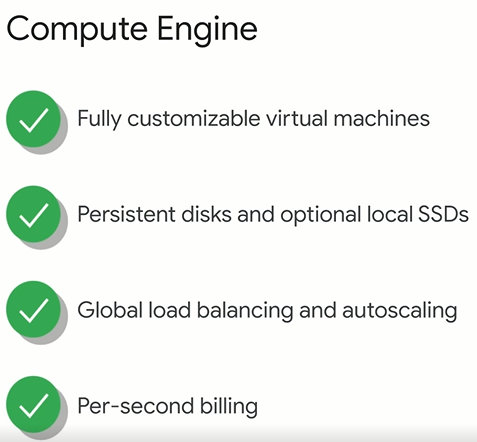
* Used to managed container infrastructure
* Open source
* Manages deployment, scaling, load balancing, logging, monitoring of containerised applications
* Declarative configuration – describe desired state rather than issue series of commands to achieve desired state
* Possible to use imperative configuration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

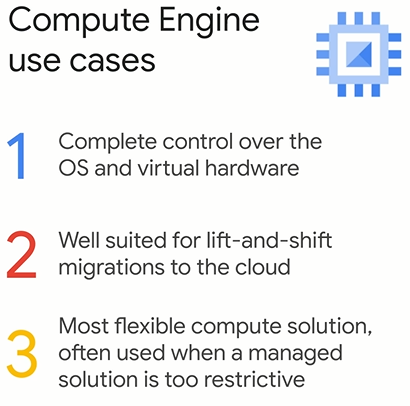
Google Kubernetes Engine (GKE)

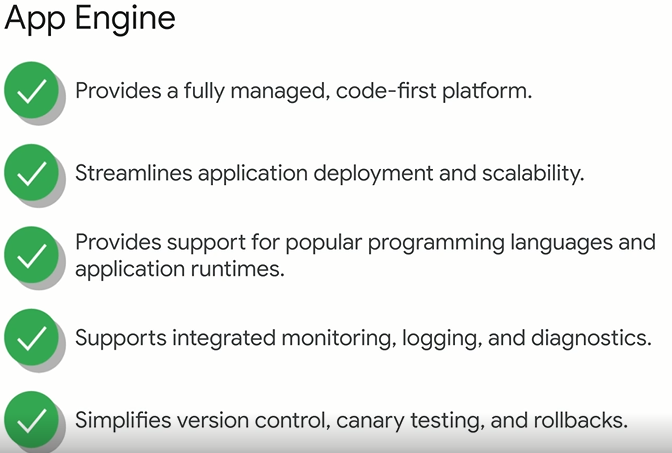
* Managed service for Kubernetes
* Fully managed
* Container optimised OS
* Auto upgrade (clusters to latest version of Kubernetes)
* Auto repair (nodes)
* Cluster scaling
* IAM
* Integrated logging and monitoring (integreates with cloud monitoring)
* Integrated networks
* Cloud console (insights into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

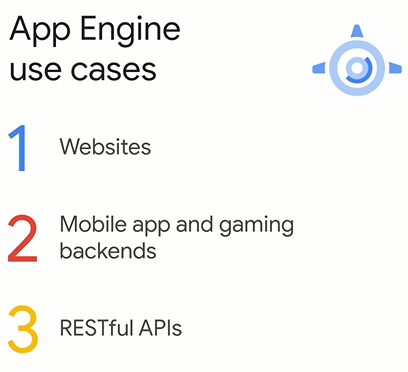
Compute Option Details

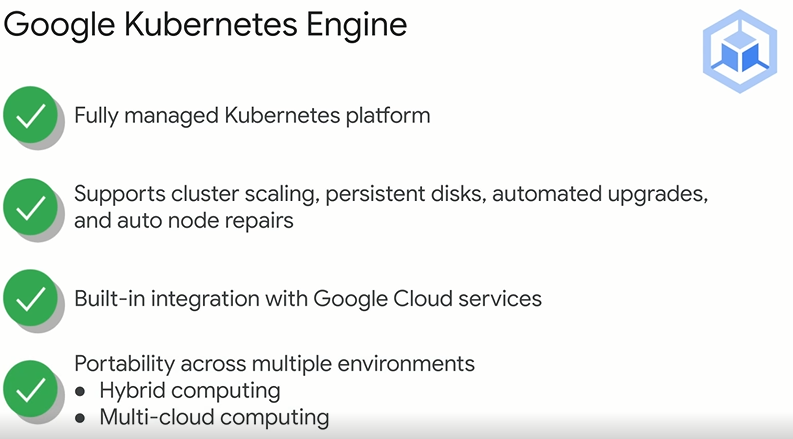


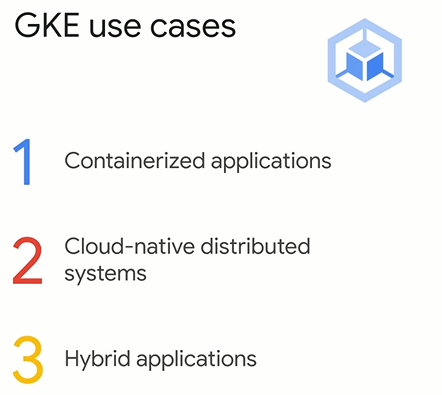


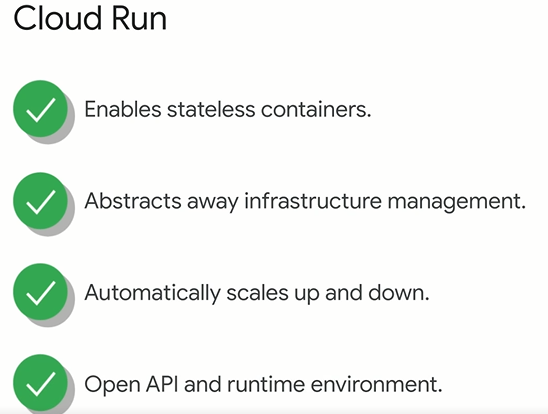


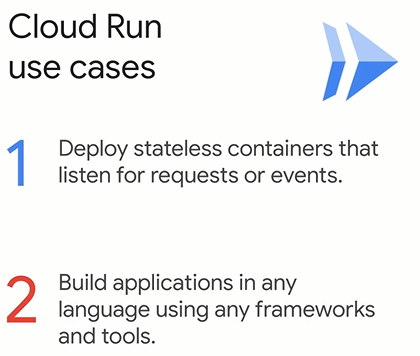


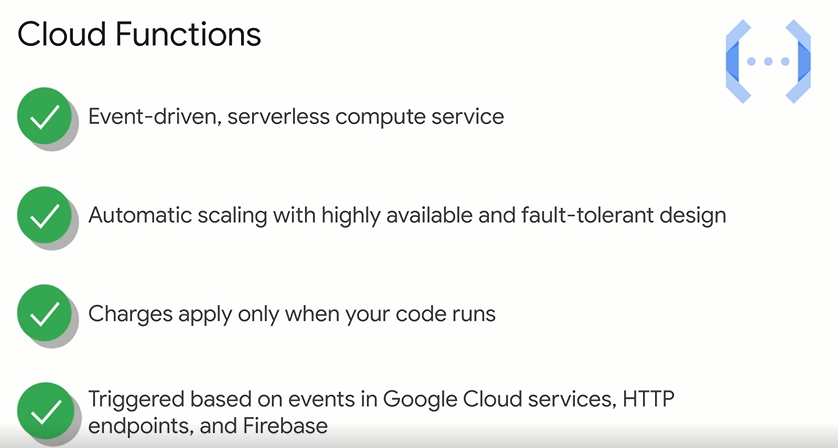


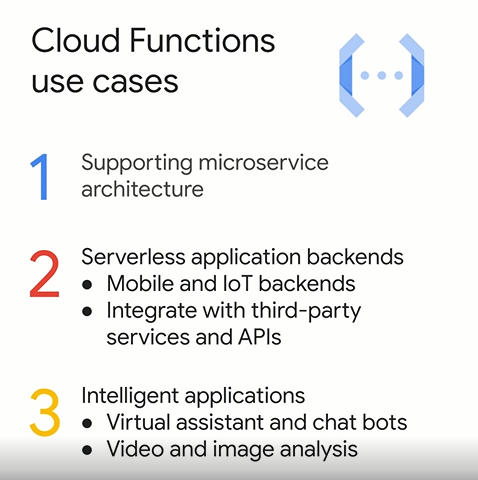


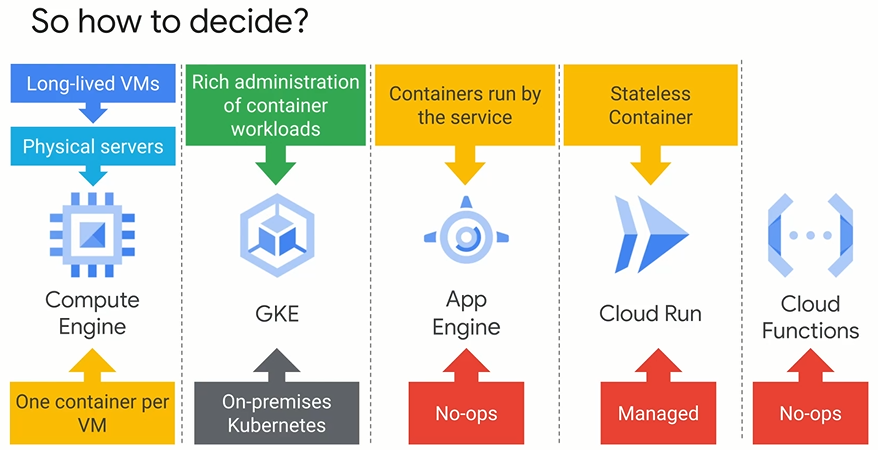












**Kubernetes Architecture**