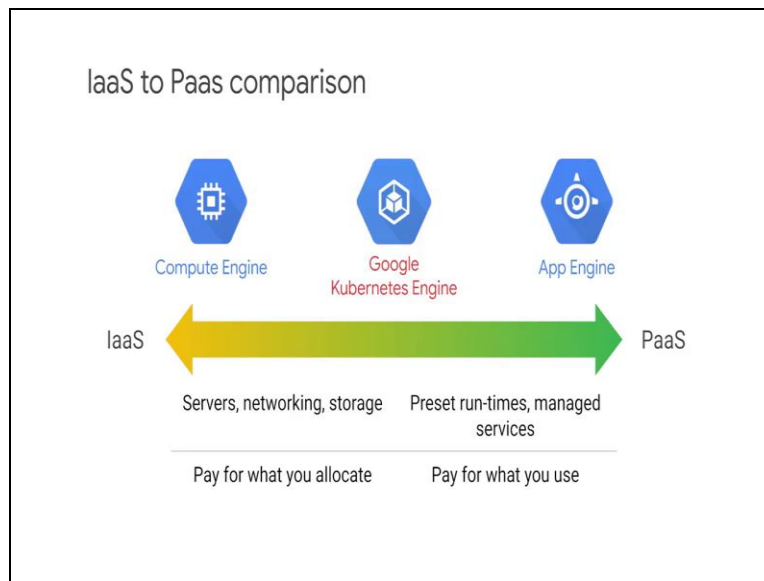


**Google Cloud Computing Foundation Course**  
**Sowmya Kannan**  
**Google Cloud**

**Lecture-23**  
**Containerizing and Orchestrating Apps with GKE**

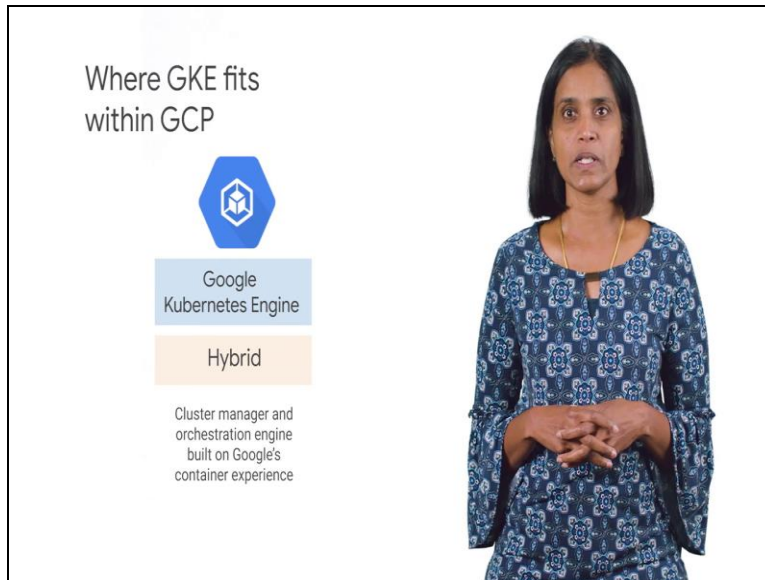
**(Refer Slide Time: 00:06)**



In this final topic you will learn how to leverage Google Kubernetes engine you have already discovered the spectrum between infrastructure as a service and platform as a service and you have learned about compute engine which is the infrastructure as a service offering of TCP with access to servers file systems and networking. Now you will see an introduction to containers and GKE which is a hybrid that conceptually sits between the two.

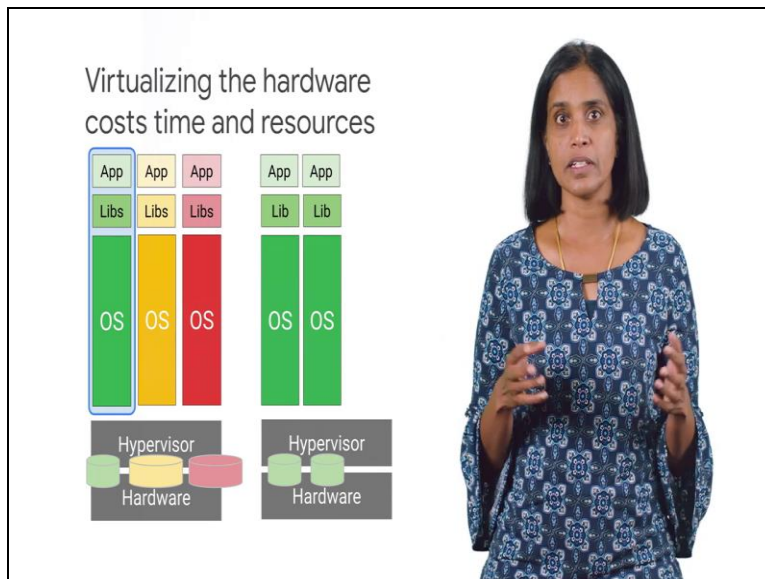
It offers the managed infrastructure of infrastructure as a service with the developer orientation of platform as a service. GKE is ideal for those that have been challenged when deploying or maintaining a fleet of VMs and it is been determined that containers are the solution.

**(Refer Slide Time: 00:56)**



It is also ideal when organizations have containerized the workloads and need a system on which to run and manage them and do not have dependencies on kernel changes or on a specific non Linux operating system. With GKE there is no need to ever touch a server or infrastructure. So, how does containerization work?

**(Refer Slide Time: 01:26)**

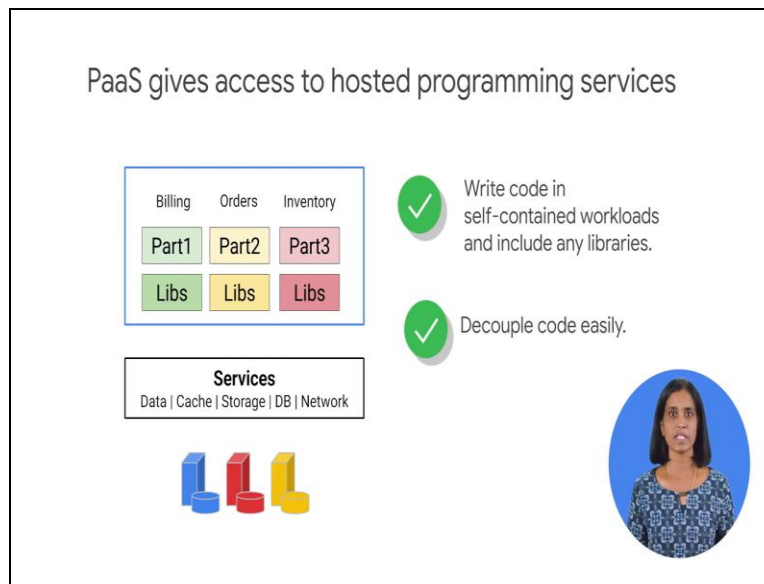


Infrastructure as a service allows you to share compute resources with other developers by virtualizing the hardware using virtual machines. Each developer can deploy their own operating system access the hardware and build their applications in a self-contained environment with access to their own runtimes and libraries as well as their own partitions of RAM, file systems,

networking, interfaces and so on. You have your tools of choice on your own configurable system.

So you can install your favorite runtime, web server, database or middleware configure the underlying system resources such as disk space disk i/o or networking and build as you like. Flexibility comes with a cost the smallest unit of compute is an app with its via the guest OS may be large even gigabytes in size and takes minutes to boot. As demand for your application increases you have to copy an entire VM and boot the guest OS for each instance of your app which can be slow and costly.

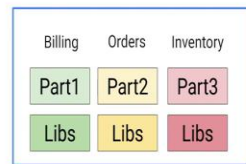
**(Refer Slide Time: 02:44)**



A platform as a service provides hosted services and an environment that can scale were closed independently. All you do is write your code in self-contained workloads that use these services and include any dependent libraries. Workloads do not need to represent entire applications they are easier to decouple because they are not tied to the underlying hardware operating system or a lot of the software stacks that you use to manage.

**(Refer Slide Time: 03:24)**

## PaaS gives access to hosted programming services



Write code in self-contained workloads and include any libraries.



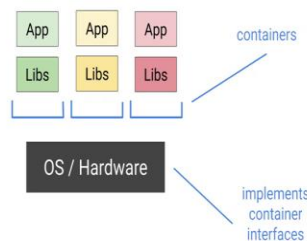
Decouple code easily.



As demand for your app increases the platform scales your app seamlessly and independently by workload and infrastructure. The scales rapidly and encourages you to build your applications as decoupled microservices that run more efficiently but you would not be able to fine tune the underlying architecture to save cost.

**(Refer Slide Time: 03:52)**

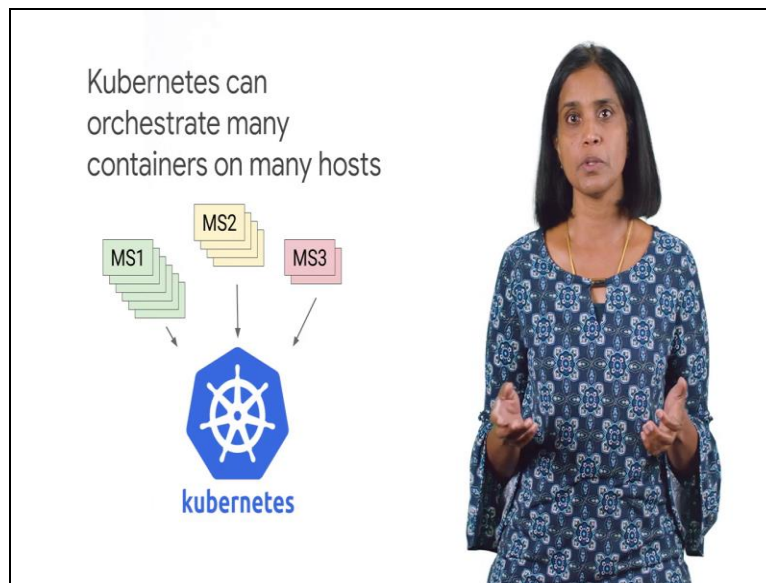
## IaaS flexibility with PaaS scalability



That is where containers come in the idea of a container is to give you the independent scalability of workloads in a platform as a service and an abstraction layer of the operating system and hardware in an infrastructure as a service. It only requires a few system calls to create and it starts as quickly as a process. All you need on each host is an OS kernel that supports containers and a container runtime.

In a sense your virtualized the operating system it scales like platform-as-a-service but gives you nearly the same flexibility as an infrastructure as a service. Containers provide an abstraction layer of the hardware and operating system. An invisible box with configurable access to isolated partitions of the file system Ram and networking as well as a fast startup with only a few system calls.

**(Refer Slide Time: 04:58)**



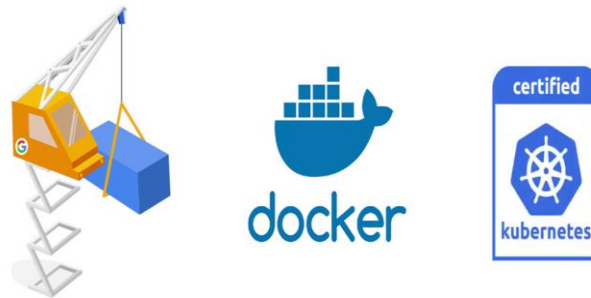
Using a common host configuration you can deploy hundreds of containers on a group of servers. If you want to scale for example a web server you can do so in seconds and deploy any number of containers depending on the size of your workload on a single host or a group of hosts. You will likely want to build your applications using lots of containers each performing their own function like micro services.

If you build them this way and connect them with network connections you can make them modular and deploy them easily and scale independently across a group of hosts. And the host can scale up and down and start and stop containers as demand for your app changes or as hosts fail. With a cluster you can connect containers using network connections build code modularly deployed easily and scaled containers and hosts independently for maximum efficiency and savings.

Kubernetes is an open source container orchestration tool you can use to simplify the management of containerized environments. You can install kubernetes on a group of your own manage service or run it as a hosted service in GCP on a cluster of managed compute engine instances called Google Kubernetes Engine. Kubernetes makes it easy to orchestrate many containers on many hosts scale them as micro services and deploy roll all and rollbacks.

Kubernetes was built by Google to run applications at scale. Kubernetes lets you install the system on local servers in the cloud manage container networking and storage deploy rollouts and rollbacks and monitor and manage container and host health.

**(Refer Slide Time: 07:03)**



Just like shipping containers the software container makes it easier for teams to package manage and ship their code. They write software applications that run in a container. The container provides the operating system needed to run their application the container will run on any container platform. This can save a lot of time and cost compared to running servers or virtual machines. Like a virtual machine imitates a computer a container imitates an operating system.

Everything at Google runs on containers. Gmail, web search, maps, MapReduce, batch processes Google file system, Colossus even cloud functions our VMs in containers. Google launches over 2 billion containers per week. Docker is the tool that puts the application and everything it needs

in the container. Once the application is in a container it can be moved anywhere that will run docker containers any laptop server or cloud provider.

This portability makes code easier to produce manage and troubleshoot and update. For service providers containers make it easy to develop code that can be ported to the customer and back. Kubernetes is an open source container orchestration tool for managing a cluster of docker Linux containers as a single system. It can be run in the cloud and on-premises environments. It is inspired and informed by Google's experiences and internal systems.

**(Refer Slide Time: 08:58)**

Google Kubernetes Engine



GKE is a managed environment for deploying containerized apps. It brings Google's latest innovations in developer productivity resource efficiency automated operations and open source flexibility to accelerate time to market. GKE is a powerful cluster manager and orchestration system for running docker containers in Google cloud. GKE manages containers automatically based on specifications such as CPU and memory.

It is built on the open source kubernetes system making it easy for users to orchestrate container clusters or groups of containers. Because it is built on the open source kubernetes system it provides customers the flexibility to take advantage of on-premises hybrid or public cloud infrastructure.