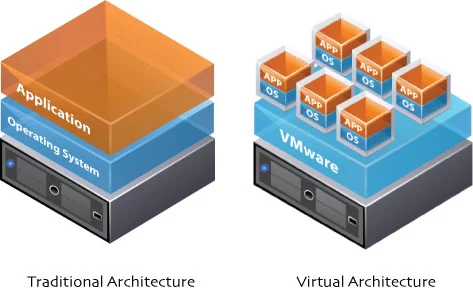
# Containers, Virtual Machine, JVM

## Virtualization

* + Virtualization is a technique on how to separate service from the underlying physical delivery of that service.It is the process of creating a virtual version of something like computer hardware.

OR

* + Virtualization is the "creation of a virtual (rather than actual) version of something, such as a server, a desktop, a storage device, an operating system or network resources".



* + Use
    - Virtualization multiple operating systems and applications can run on the same Machine and its same hardware at the same time increasing the utilization and flexibility of hardware.
    - cost-effective, hardware reducing, energy saving techniques
    - More flexible and efficient allocation of resources.
    - Enhance development productivity.
    - It lowers the cost of IT infrastructure.
    - Remote access and rapid scalability.
    - High availability and disaster recovery.
    - Pay per use of the IT infrastructure on demand.
    - Enables running multiple operating systems

## Types

* + - 1.Hardware Virtualization.
    - 2.Native, full Virtualization.
    - 3.Para-Virtualization.
    - 4.OS level Virtualization.
      * Container
      * Jails
      * Chroot
      * Zones
      * Open-V → Virtuozzo

|  |  |  |  |
| --- | --- | --- | --- |
| Hardware Virtualization | Native, full Virtualization | Para-Virtualization | OS level Virtualization |
| The VM has its own hardware and allows a guest OS to run in isolation | VM simulates enough hardware to allow an unmodified guest OS to be run in isolation | It offers a special API that can be used by modifying the guest OS | Virtualizing a physical server at the OS level, enabling multiple isolated and secure virtualized servers to run on a single physical server |
| E.g  Intel VT(IVT)  AMD Virtualization(AMD-V)   * Parallel Desktop for MAC * Parallel workstation | VirtualBox, Virtual Pc, Vmware, QEMU | XEN | Linux Vserver  Solaris Containers  FreeBSD Jails  Chroot  CGroup |

* Building a lightweight machine by → Container tech
  + Lightweight threads
* Software protocol → building containers very fast and easily
* JVM(Java virtual machine)
  + Providing anything related to Java language

## Layers of Abstraction

* To share a physical computer among multiple users by → Abstraction
  + Providing a generic computing resource should look like
  + Then the physical computer would provide to users
    - Those systems will provide in an abstract way
      * Compute,
      * storage,
      * Network

## Hypervisor

* + A hypervisor is a function which abstracts -- isolates -- operating systems and applications from the underlying computer hardware
  + The hypervisor program installed on the computer allowed the sharing of its memory.
    - E.g. VMware, VirtualBox

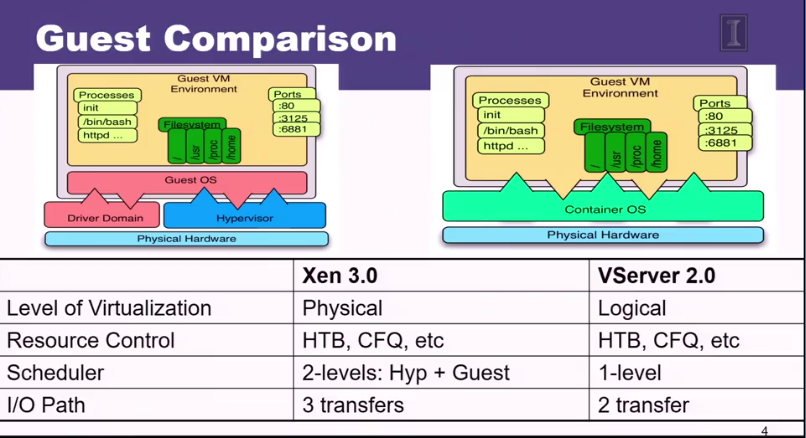
## Hypercalls

* + The Parallels Workstation operating system calls its equivalent a "hypercall".

## Xen

* + The Xen Cloud Platform (or XCP) is a turnkey open source virtualization solution that provides out-of-the box virtualization and cloud computing.

## Vserver 2.0



## Solaris Zone

* + A Solaris Zone is a partitioned virtual OS environment working in a Solaris OS space. There are two types of Solaris Zones: Global Zones and Non-Global Zones.
  + Zones are provided separate virtualized operating system environments that are derived from global zone.

## Types of Zones

* + Global zone
    - Solaris 10 installation contain a Global zone
    - This is the only zones which is bootable from the system hardware.
  + Local Zones
    - It contains a subset of the complete operating system, and can contains non-shared packages.
    - It doesn’t have awareness of the other zones.
* Zones Resource Capping
  + A resource cap is an upper bound placed on the consumption of a resource, such as physical memory. Per-project physical memory caps are supported

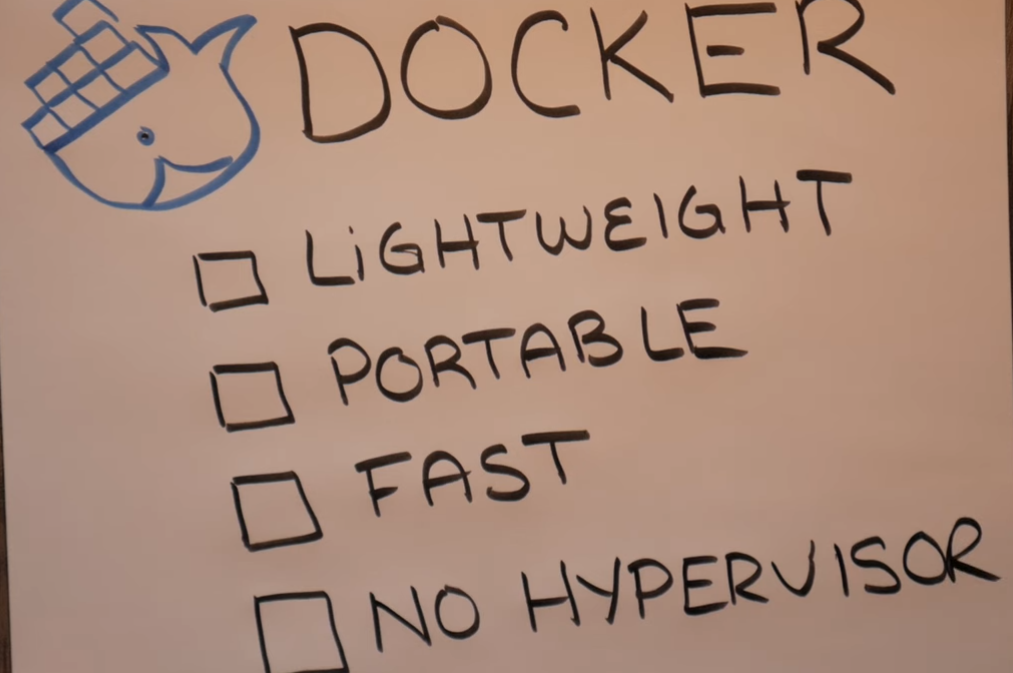
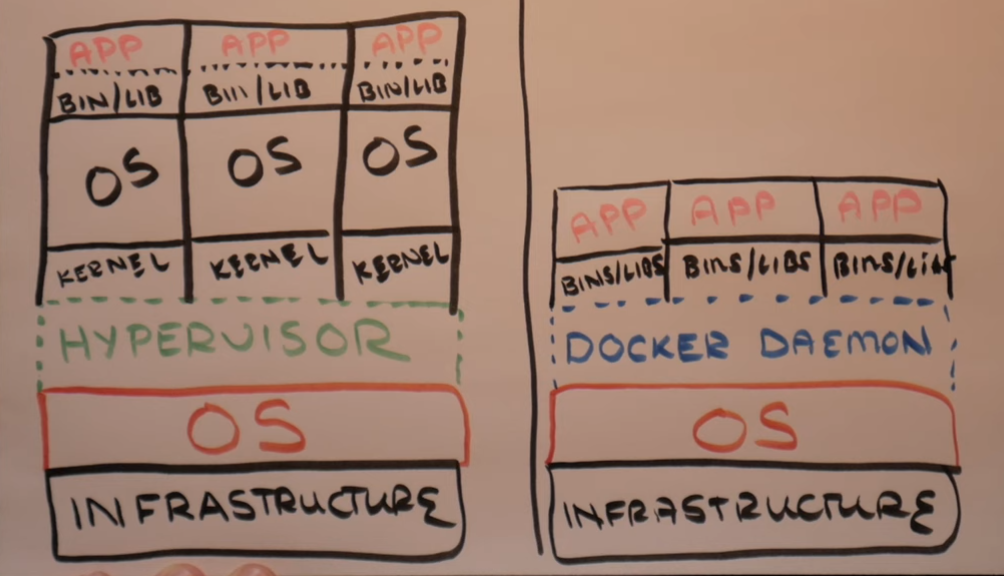
## Linux Container

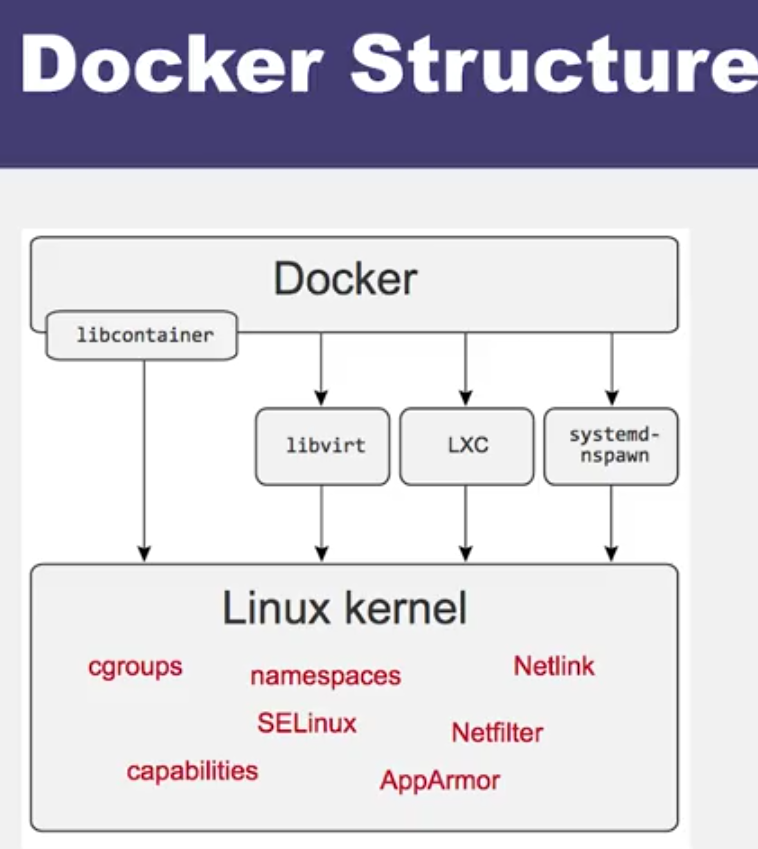
* Container in cloud computing is used to build blocks, which help in producing operational efficiency, version control, developer productivity and environmental consistency. By this, the user is assured of reliability, consistency, and quickness regardless of the distributed platform.
* OpenVZ
  + Each container has its own:
    - Files: chroot()
    - Process tree(PID namespace)
  + Network(net namespace)
  + Devices
  + IPC objects(IPC namespace)

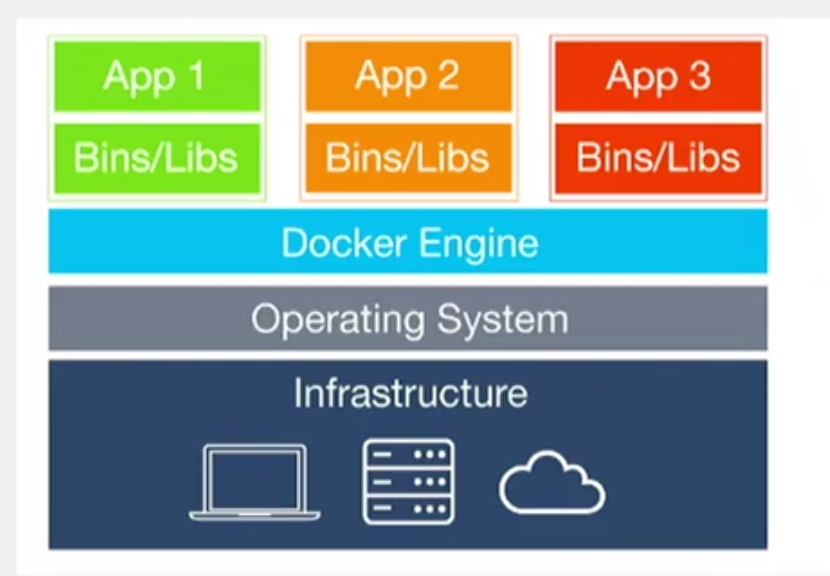
## lxc Linux container

* + Better isolation as compared to a chroot
  + Application and services run at the native speed
  + It is an open source.
  + No patch is required to the kernel.

## Docker

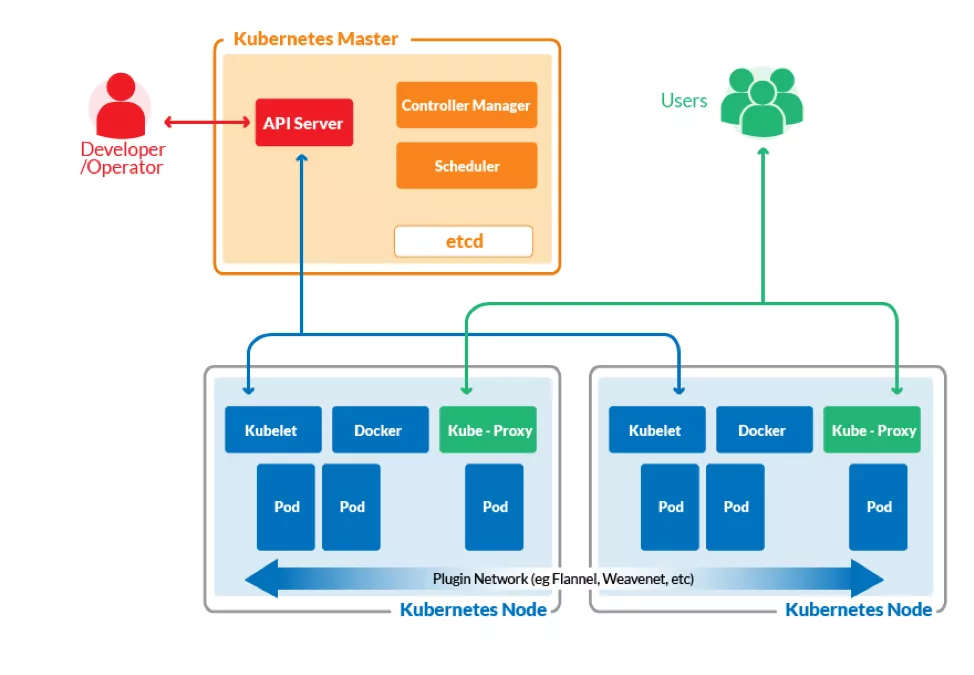
* 
* 
* It is a container that wrap up a piece of software in a complete filesystem that contains everything needed to run: code, runtime, system tools, system libraries.
* It automates the deployment of the application inside software containers.
* Docker is a computer program that performs operating-system-level virtualization.
* Docker is used to run software packages called containers. Containers are isolated from each other and bundle their own application, tools, libraries and configuration files; they can communicate with each other through well-defined channels.





Kubernetes

* Definition
  + It provides a “platform for automating deployment, scaling, and operations of application containers across across clusters of hosts.



## Difference between Kubernetes and Docker

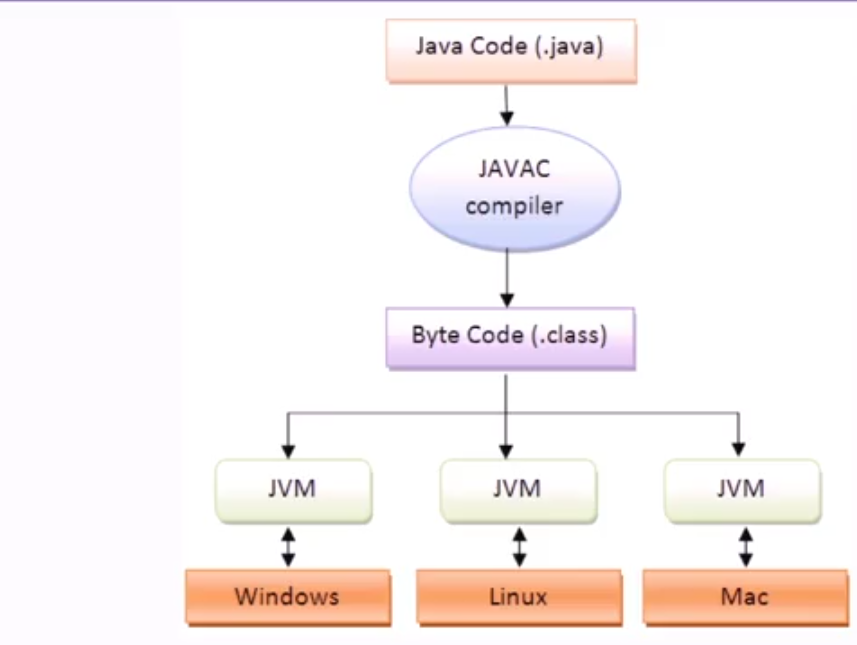
* + Kubernetes is that it’s a largely open source
  + Kubernetes specifies that each pod should be able to freely communicate with every other pod in the cluster in a given namespace, whereas Docker has a concept of creating virtual network topologies and you have to specify which networks you want your containers to connect to.
  + The fundamental differences of Kubernetes vs Docker: The former is meant to run across a cluster while the latter runs on a single node.

## Similarities between kubernetes and docker in cloud computing

* + Their love for microservices-based architecture (more on this later).
  + Their love for open source community. Both are large open source projects.
  + They are largely written in Go programming language, which allows them to be shipped as small lightweight binaries.
  + They use human-readable YAML files to specify application stacks and their deployments.
* Parts of kubernetes
  + Pods
  + Labels and Selectors
  + Controllers
  + Services

JVM

* Definition
  + It is a an abstract computing machine that enables a computer to run a Java program.
    - Specification
      * Document formally describing a JVM implementation.
    - Implementation
      * A computer program meeting the JVM specification requirements.
    - Instance
      * An implementation running in a process that executes a computer program compiled into a Java bytecode.
* JRE(Java Runtime Environment)
  + It is software package that contains what is required to run a Java program.
* JAva execution



* Difference b/w JVM and classic VM



* Why JVM in cloud
  + It is secure for remote execution
  + It is portable distributed applications.
  + It supports elasticity and flexibility.
  + It runs on most hardware.
  + It supports more secure browsing.
  + It is easier to software engineer because of typing and language.
  + It is easier to share(means open source).

IAAS openstack

* OpenStack is a free and open-source software platform for cloud computing, mostly deployed as infrastructure-as-a-service (IaaS)

IAAS providers

* Amazon
* Microsoft
* Google

Serverless Architecture

* Serverless architecture means that developers and operators do not need to spend time setting up and tuning auto-scaling policies or systems; the cloud provider is responsible for scaling the capacity to the demand.