

LPIC-1 101-500 – Lesson 17

102.6 Linux as a virtualization guest



Virtualization, Containerization and Cloud computing


- Virtualization is a technique to create Virtual Computers or Virtual Machines (VMs) inside a physical computers. The VMs are known as the guest and the physical computer as the host.
- Containerization is a technique to create isolated environments on a host (either physical or virtual) to host applications or minimal operating systems.
- Cloud computing is the use of technologies like Virtualization, Containerization, Storage, Networking as a service. In the last decade their has been a boom in Cloud computing usage with Linux being a driving force with 90% market share on public cloud workloads.

Virtualization

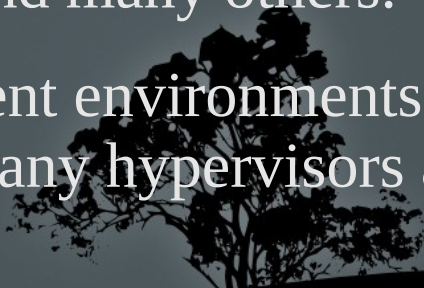
- Emulates almost a complete hardware computer with software.
- Can emulate almost any operating system irrespective of the operating system on the host.
- Performance can be enhanced on hosts supporting Hardware-assisted virtualization.
- Performance on the guest machine can also be improved by using special drivers for the virtual hardware called **guest drivers**.



Virtualization Terminology

- **Image:** a file (usually a virtual disk) that contains the guest OS.
 - **Snapshots** are an easy way to capture the state of a VM (disk and memory) at a certain time and be able to restore it at a later stage.
 - **Cloning** is the process of creating a copy of another machine for backups or experimentation.
 - **Open Virtualization Format:** useful for cloning a VM across different hypervisors. The resultant **OVF** files can be bundled together in an Open Virtualization Archive (**OVA**).
 - **Template:** an image that has been prepared beforehand to create VMs with the same OS, applications, structure, etc.
 - **Migration** is seamlessly relocating a VM from one host to another.
 - **Failover** is when a VM seamlessly migrates to another host when its host fails.
 - **GPU Virtualization** is sharing the resources of a GPU (graphics card) from the host to the guest. Used for compute intensive applications like Machine Learning, Cryptocurrencies etc.
 - **Desktop Virtualization** is when virtualization is used for the emulating of Desktop Machines.
- 

Virtualization systems (*Hypervisors*) and tools in Linux

- **KVM:** The most popular hypervisor in Linux, included the Linux kernel. Usually combined with the **libvirt** user-space tools for easier management.
 - **XEN:** another popular hypervisor also included in the linux kernel.
 - **QEMU:** one of the first hypervisors in Linux which can be integrated with KVM for faster performance. It can also emulate different CPU architectures like x86, ARM, PowerPC and others.
 - **VirtualBox:** another popular hypervisor with a userfriendly GUI, mainly for Desktop virtualization.
 - **libvirt:** a management tool for many linux hypervisors and container platforms like, KVM, XEN, QEMU, LXC and many others.
 - **Vagrant:** A tool for creating virtual development environments using a repository of Linux OS images. Support many hypervisors and container platforms.
- 

Cloning considerations

- When cloning a VM some important unique parameters are cloned as well. Care should be taken so as these parameters be regenerated on the clone.
- Regenerate the ssh host keys:


```
# rm /etc/ssh/ssh_host_*  
# ssh-keygen -A  
# ls -la /etc/ssh/ssh_host_*
```

- Regenerate the Machine ID:


```
# rm /etc/machine-id  
# rm /var/lib/dbus/machine-id  
# dbus-uuidgen --ensure
```

- There are tools like **cloud-init**, that can automate these processes along with cleaning command history, logs and caches.

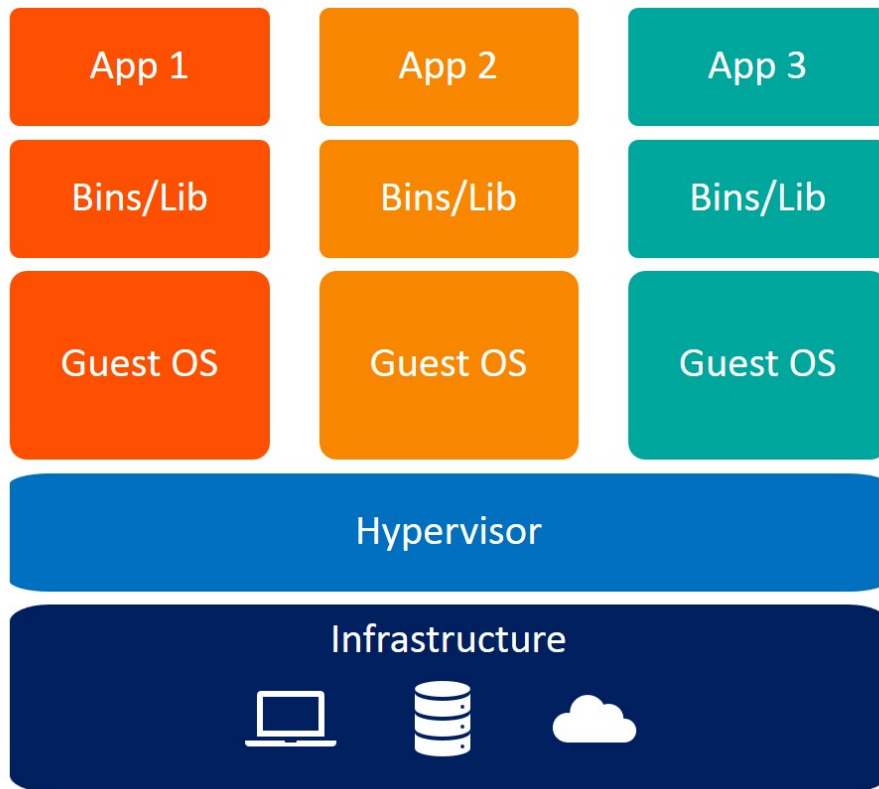
Containerization

- Also called **OS-level virtualization**.
 - Can only emulate the same operating system as the host (**Operating System Containers**), and applications (**Application Containers**) written for operating system of the host.
 - Boast Better Performance than Virtualization.
 - Provides less isolation than Virtualization (less secure).
 - Easier and faster deployment and decommissioning.
 - Provides a consistent environment across different platforms. For example one can develop the system on a different distribution than the target or even Windows or MacOS and run the application on a Linux host.
 - Better redundancy though multiple instances per application plus load-balancing
 - They can run either on Bare Metal (Physical servers) or VMs.
- 

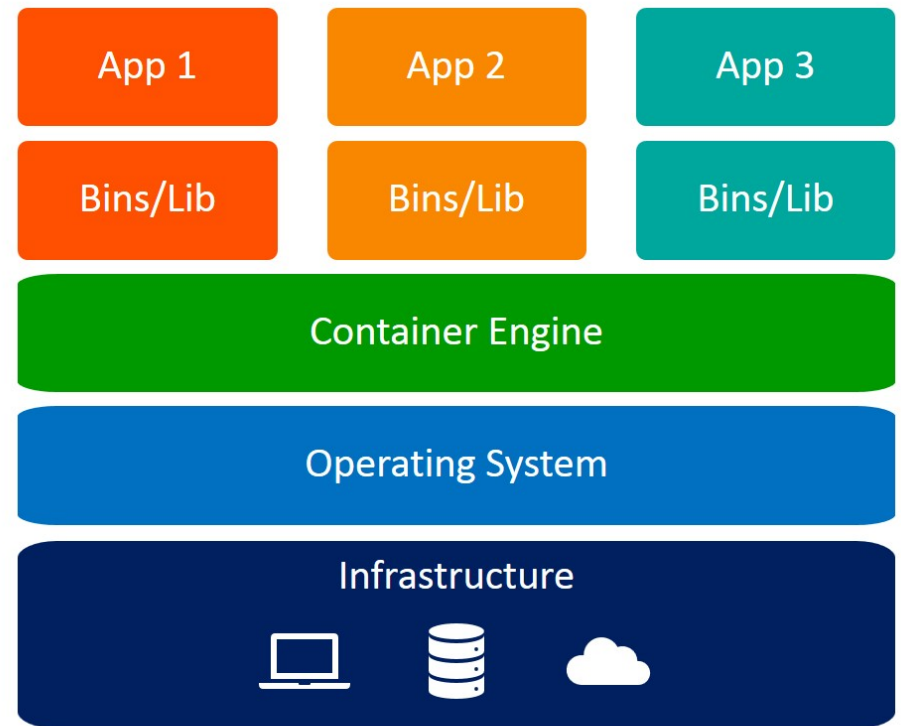
Container Engines in Linux

- **OpenVZ:** One of the first container engines for Linux.
 - **LXC:** A very popular engine mainly for OS containers, included in the linux kernel.
 - **LXD:** developed by Canonical on top of LXC aiming to provide a better user experience.
 - **Docker:** One of the most popular platforms for application containers.
 - **rkt:** Mainly for application containers and a Docker competitor.
- 

VMs vs Containers




Virtual Machines




Containers

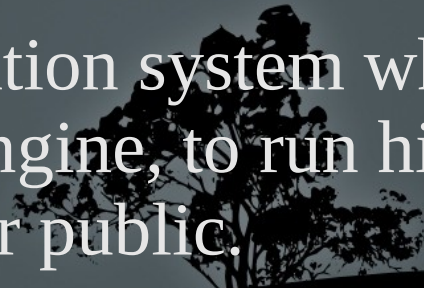
Cloud Computing

- There is a growing trend for companies to host their infrastructure on hosted services on a “Pay as you grow model”. This paradigm is also known as “Infrastructure as a Service” (IaaS).
 - Some fundamental components of IaaS:
 - **Compute:** The virtualization part (CPU, Memory).
 - **Networking:** virtual networks switches and routers.
 - **Block storage:** virtual disks, usually fixed.
 - **Object storage:** accessing storage via HTTP/API which can grow along with the data.
 - **Images:** can be used as templates to launch instances.
- 

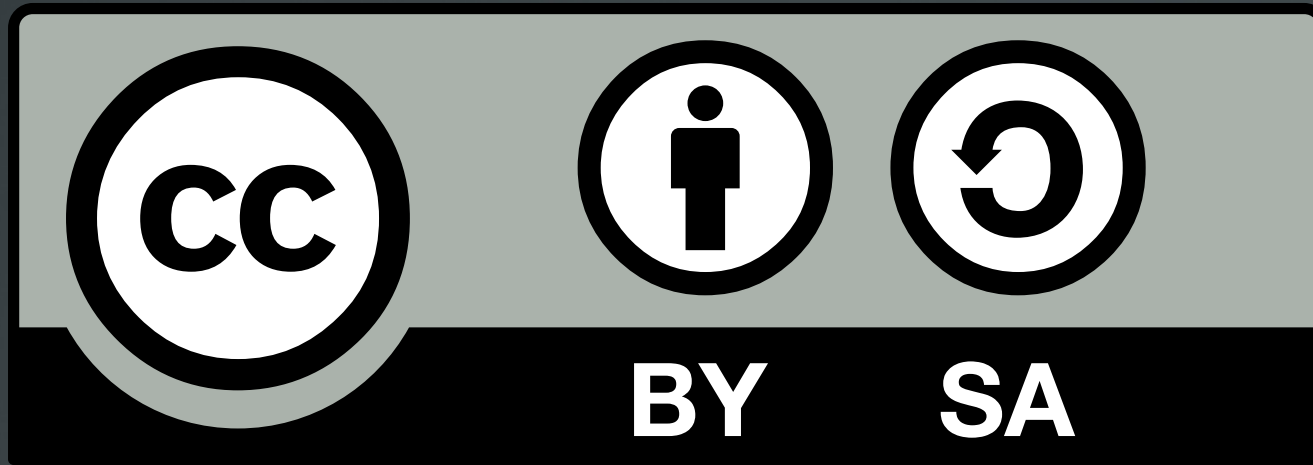
Cloud Computing Terminology

- **Instance:** an application can be composed of many instances (either VMs or Containers).
 - **Elasticity:** also know as **Auto Scaling**. This allows an application to launch more instances as demand grows.
 - **Load Balancing:** most cloud platforms can automatically load balance the traffic among instances.
 - **Public Cloud:** Infrastructure hosted on public services.
 - **Private Cloud:** Infrastructure operated by a single organization.
 - **Hybrid Cloud:** Infrastructure that combines Public and Private environments.
- 

Cloud Computing Solutions

- **AWS:** Public cloud Operated by Amazon.
 - **GCP:** Public cloud Operated by Google.
 - **Azure:** Public cloud Operated by Microsoft.
 - **DigitalOcean:** Public cloud alternative to the above.
 - **OpenStack:** popular open source platform for private and hybrid clouds. Uses VMs and Containers.
 - **Apache CloudStack:** another private and hybrid cloud alternative.
 - **Kubernetes (K8s):** a container orchestration system which uses Docker, rkt or its native CRI-O engine, to run highly scalable applications. Can be private or public.
- 

License



The work titled "LPIC-1 101-500 – Lesson 17" by Theodotos Andreou is distributed with the Creative Commons Attribution ShareAlike 4.0 International License.

