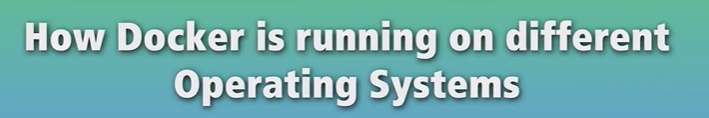
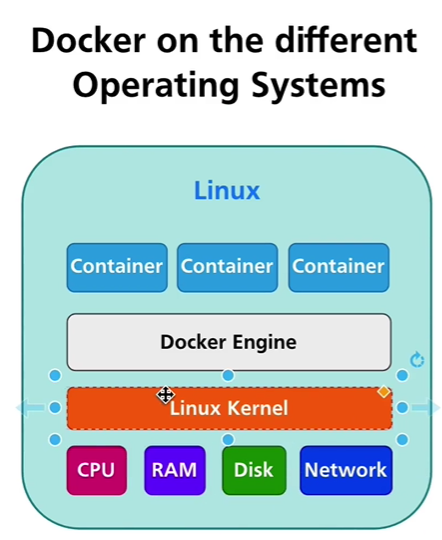
1. 
2. We saw that

|  |  |
| --- | --- |
| Virtual Machine | Docker Container |
| Each VM has its own Guest OS. | Each Container relies on the Host OS. |
| Each VM has dedicated resource. | Each Container shares the resources assigned to **Docker Engine**. |

1. Let’s see how Docker containers run on host operating system.  
   
   1. In the above diagram, Linux is the Host OS.
   2. That host Linux OS has some resources such as CPU, RAM, Disk, Network along with Linux kernel.
   3. On this OS, Docker Engine is installed.
   4. Due to Docker engine, now we can create containers inside Linux machine.
   5. Now containers would be sharing resources allocated to Docker engine and Docker engine may have full resources of the underlying machine or restricted resources.
   6. In case of dedicated resources assigned to Docker Engine, the resource quantity must be less than that of Host OS.
   7. In Linux and windows, it’s possible to group resources per process or per group of processes but no possible in MAC.  
      
   8. The main idea behind all of this Docker setup is that it’s possible to group resources per process inside a container or per group of processes inside a container.
   9. **For example**, you’re able to run 10 different containers with node.js processes and every container can run different node.js version or you can run 10 different web servers one in each container where each web server is of different version.  
      Again this is possible due to separation of resources per process inside of the Linux
   10. But no question how it’s possible to do the same on MAC OS or windows.
   11. It is possible if you create very tiny Linux machine on MAC OS or Windows computer.
   12. When we install Docker on MAC OS or windows, a tiny Linux machine is installed and that is responsible to run containers.
2. 