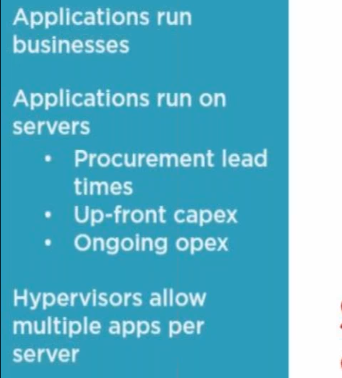
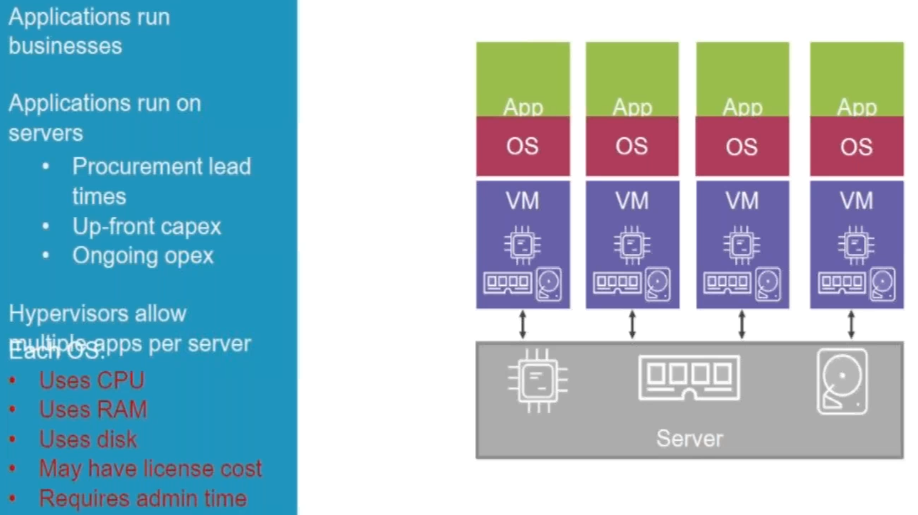
**Docker**

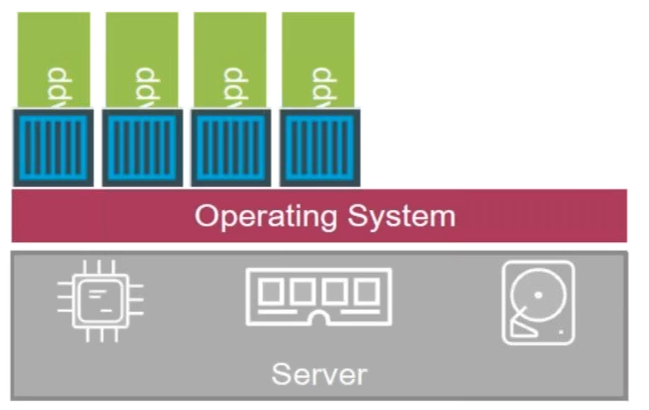


* In physical server, we can have virtual machine running and with the remaining memory, we can have another virtual machine for another purpose
* Rather than free memory, we can use virtual machine

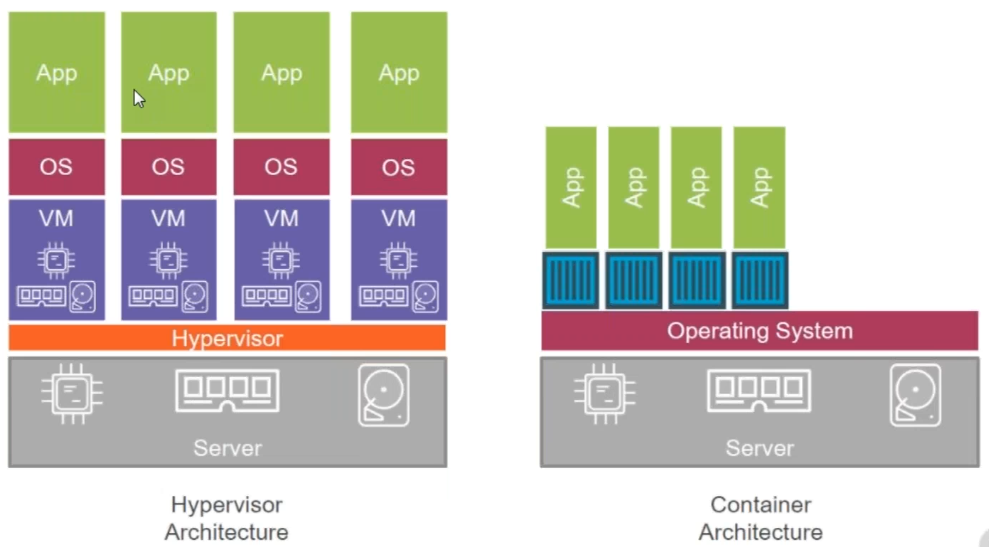


* As above, wee need 4 licensed OS for 4 applications for virtualisation
* Here, even the OS consumes some memory
* That’s why we better use containers
* So, the applications will not have OS overheads
* It will be using the underlying OS of server
* So, we get all the benefits of isolation and remove the overheads of OS

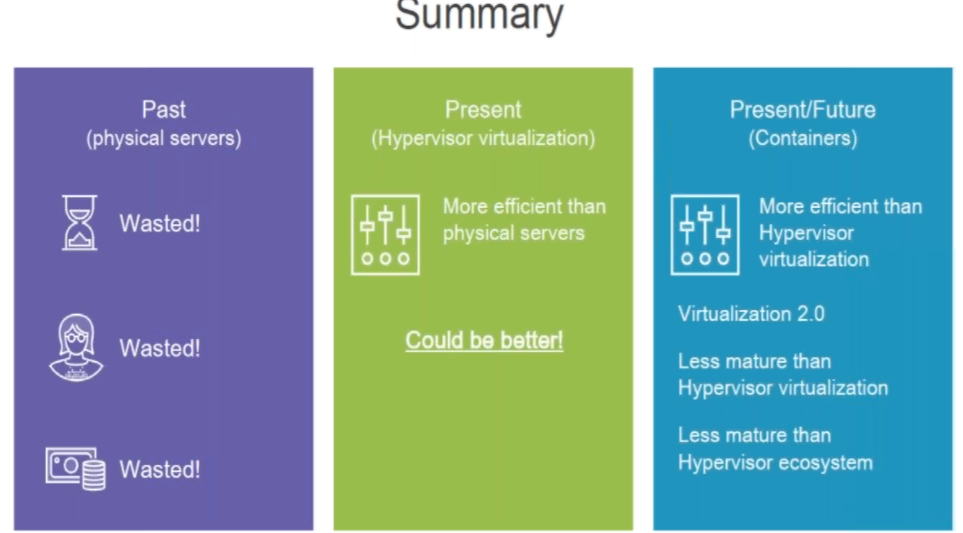
**Containers:**



* As above, blue box is the isolated area which is private area for application
* For containers, it looks like an OS for application, but it is using the OS of server

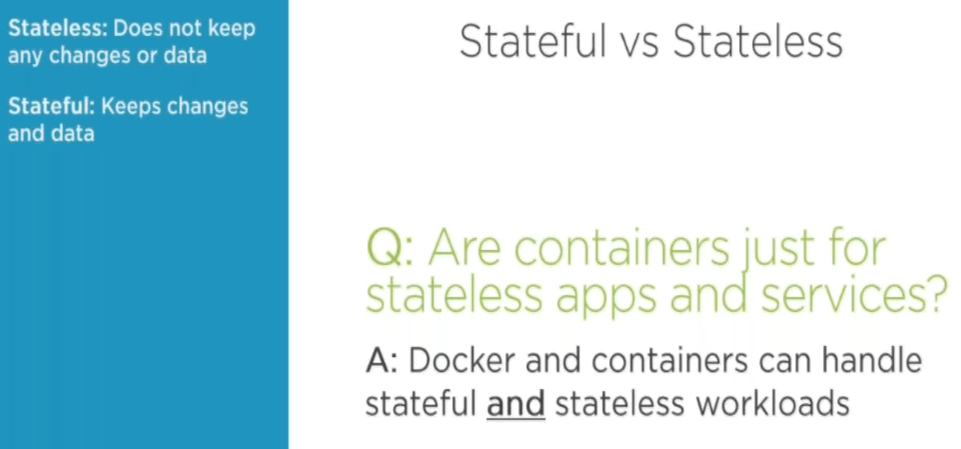


* Microservice breaks the application into multiple services and each service they host it on containers
* So, if we want, we can increase the size of particular service
* With docker, we can get benefits of micro services

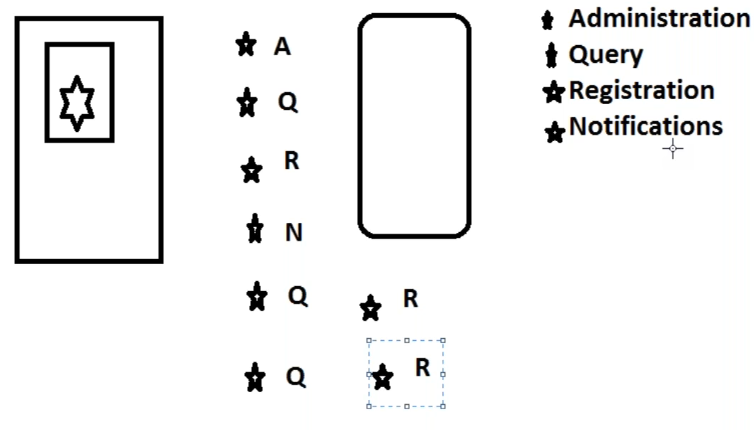


**What kind of work will we do in containers?**

* We can’t install desktop applications, we can server side work like website, webservers etc…
* There are two type of applications



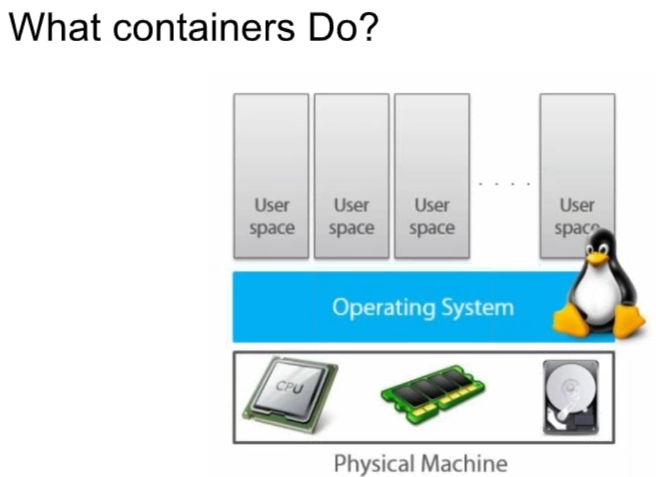
* Stateless do whatever we want but stateful stores the result



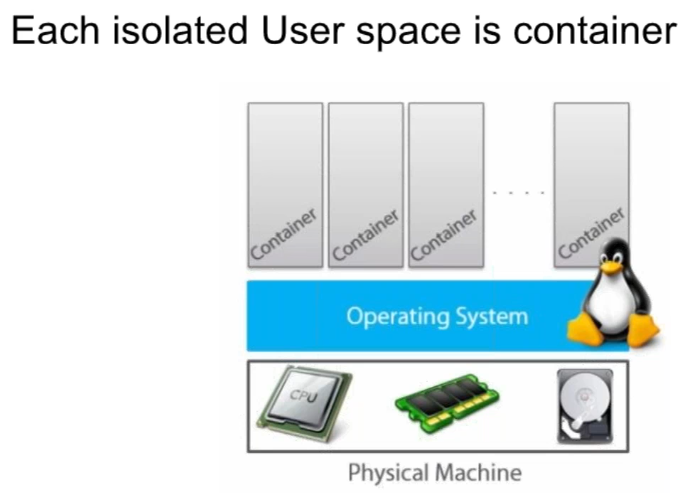
* As above, rather than putting all the stars in OS, we put it each one on container
* We can use docker in local DC also
* Container is a concept and docker is a product or implementation on that
* Chocolatey is used to download the apps
* **Choco search git**
* **Choco install vscode**
* Docker is always about applications
* There is community and enterprise edition in docker



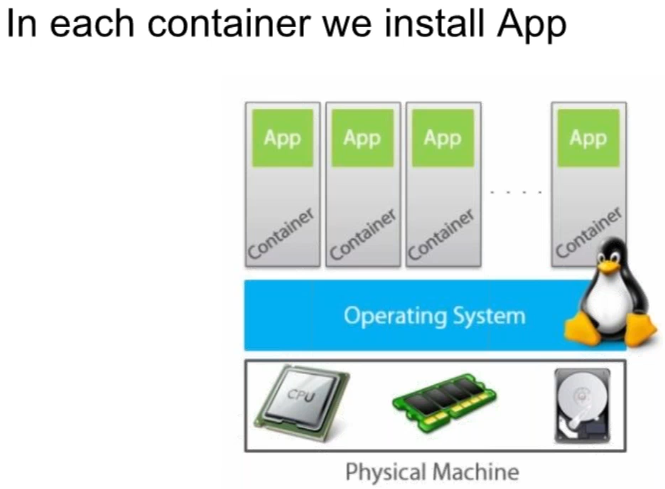
* In physical machine, we have users’ space and in that we will be running applications



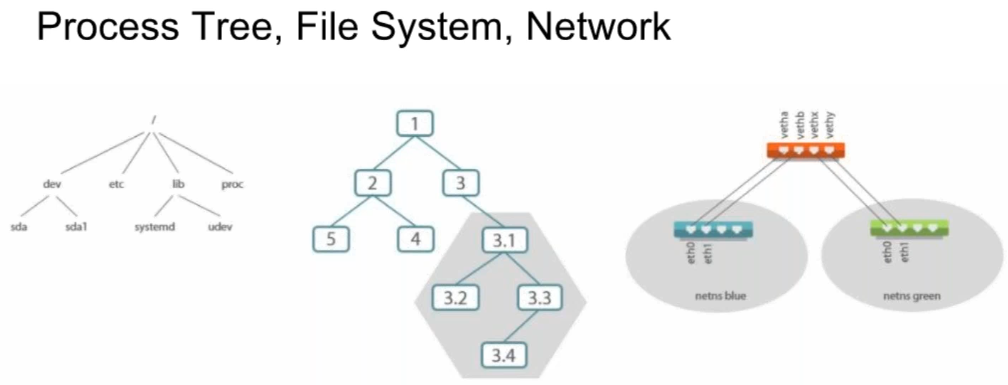
* In containers, it will have multiple user space.



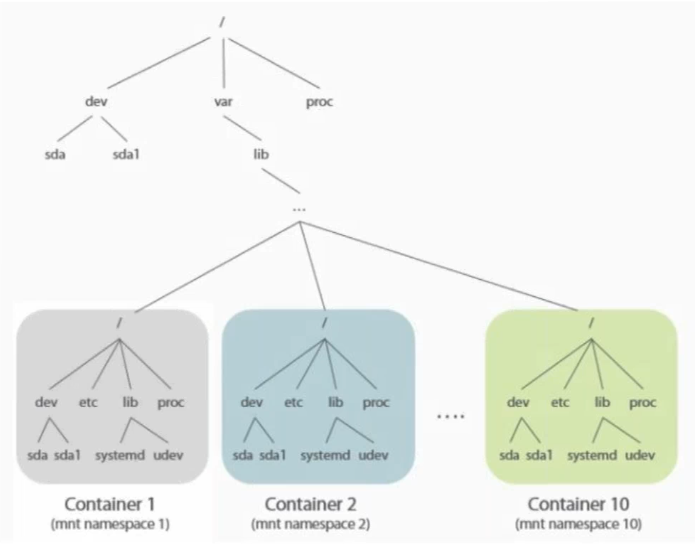
* In that we will be launching applications



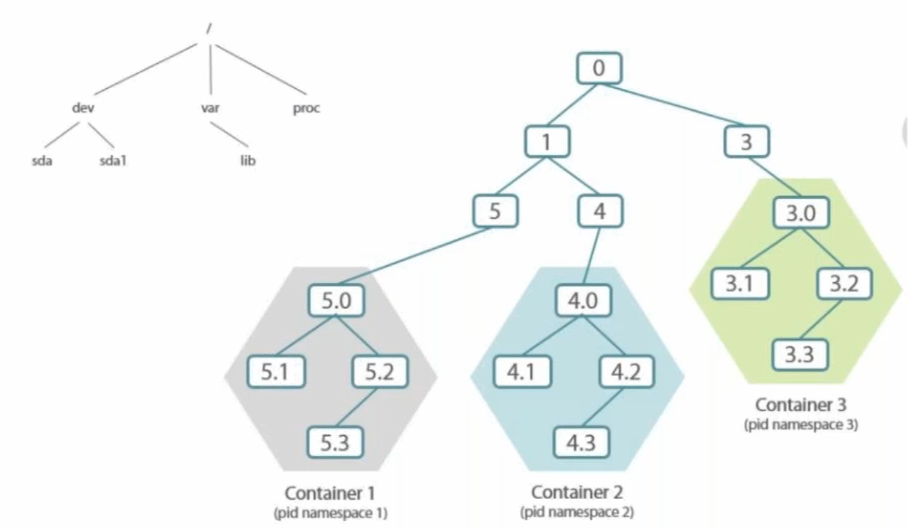
* One isolated space cannot access other
* Each isolated space is called container
* to our OS, it is just a process and for the application it looks like a full-blown OS



* Every system has process, file system and network

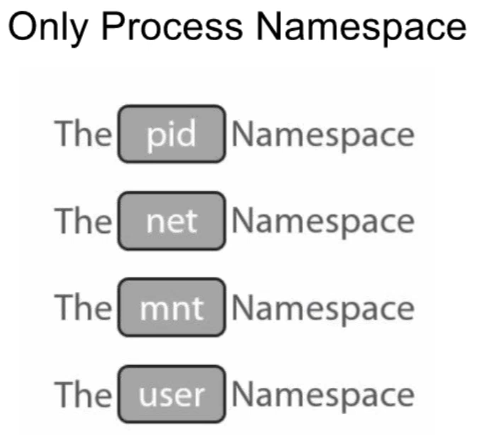


* As above, it comes from var/lib and in docker it again starts from root
* That is virtual file system inside the isolated space



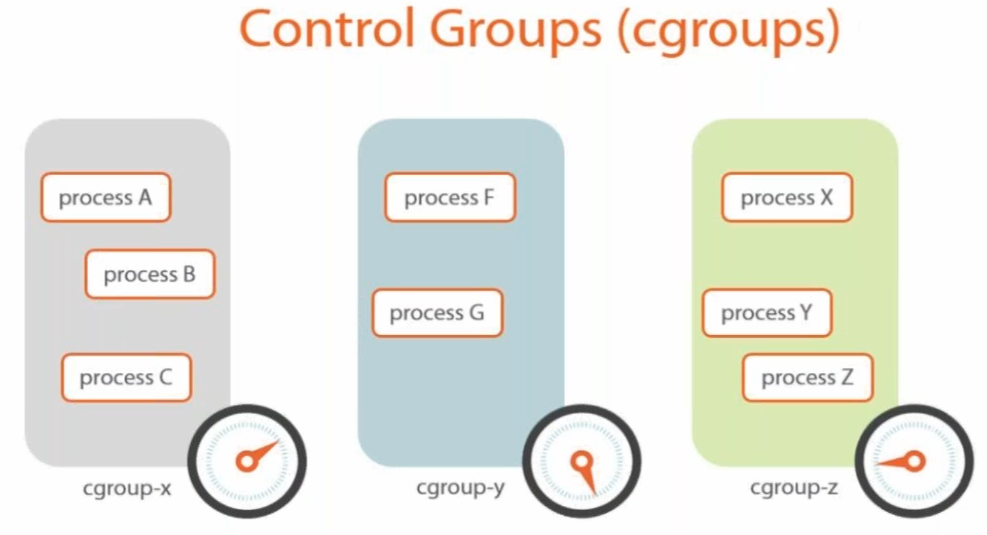
**Namespace**:

* Same thing happens to the process also



* Each container is not physical, but it looks like an OS
* namespaces are used to create isolations
* all mount points, users etc. isolations maintained by a Linux kernel called namespaces

**cgroups**:



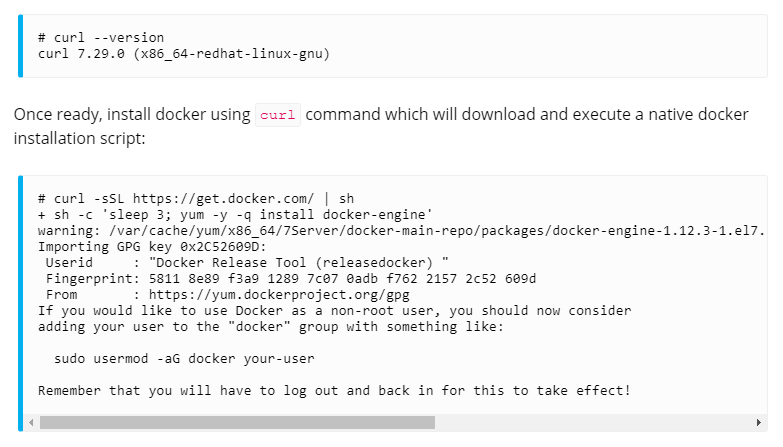
* Cgroups is used to increase the ram, memory etc.

**Capabilities**:



* We have capabilities, as in Linux if we want a user who access only network, instead of giving sudo which makes access for everything, we give capability to only network
* Docker works with capabilities not with sudo
* capabilities are used for the permissions
* We can have container inside container and we can also have multiple applications inside one container. But both are not good practices
* docker cannot control Linux kernel, if kernel got updates, then docker might not work
* lib container has the Linux kernel capabilities which docker supports, this is developed by docker
* when we install, we have lib container, this lib container speaks to Linux kernel

**Installation:**



* Use the above shell script to install docker simply. It will always install the latest version