**Basics**

Virtual Machine

Container Evolution

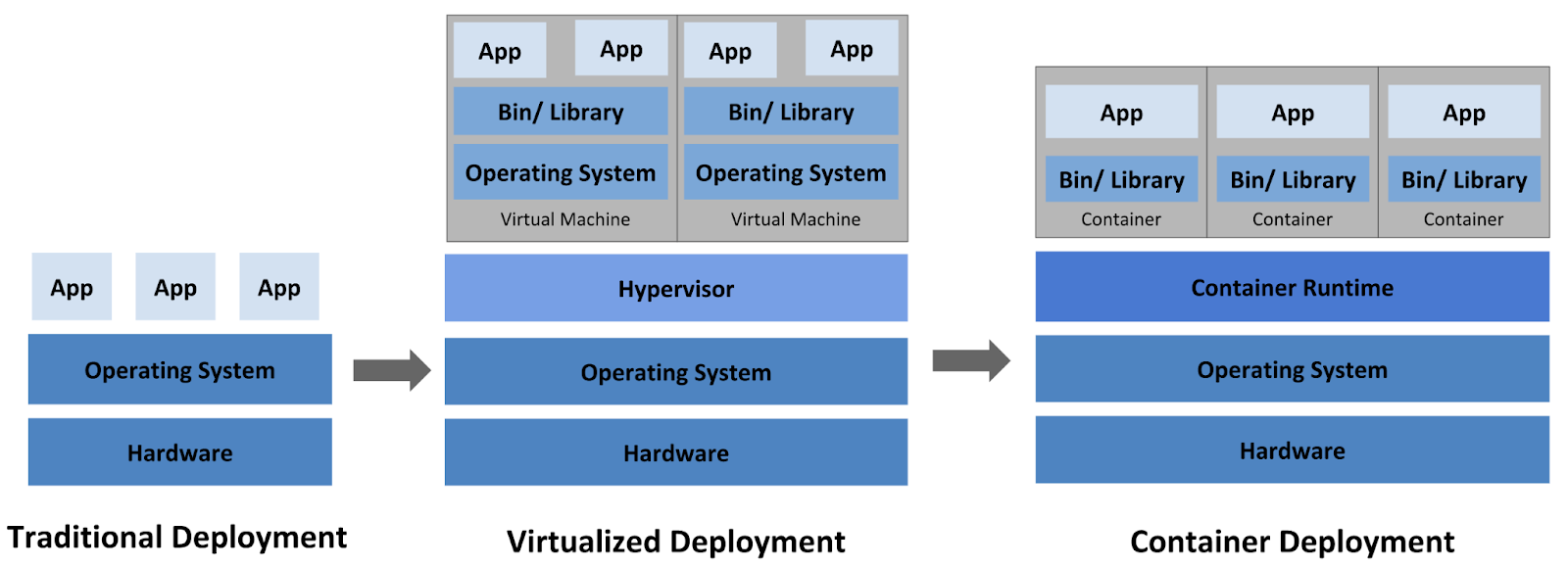
Container Runtime Engine

Installation and Configuration

Docker basics - Create, View, and Delete Containers

Architecture

# Container Evolution



Cloud

Region01(us-east)

AZ01(datacenter)

BareMetal Server(128GB mem)

VM01(4GB mem+2vCPU+192.168.1.2)

Storage

Compute(mem+cpu)

Network

VM02(4GB mem)

Container01(512mb)

Container02(512mb)

Container03(512mb)

VPC(network)

Storage

AZ02

Region02(us-west)

AZ01

AZ02

Region03(us-central)

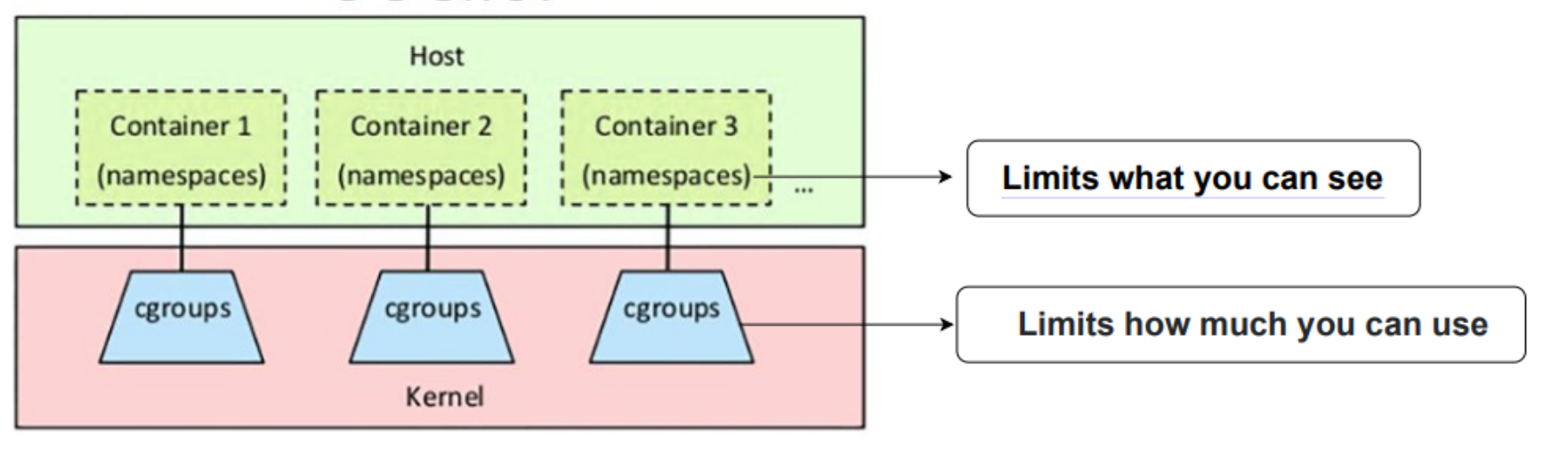
AZ01

AZ02

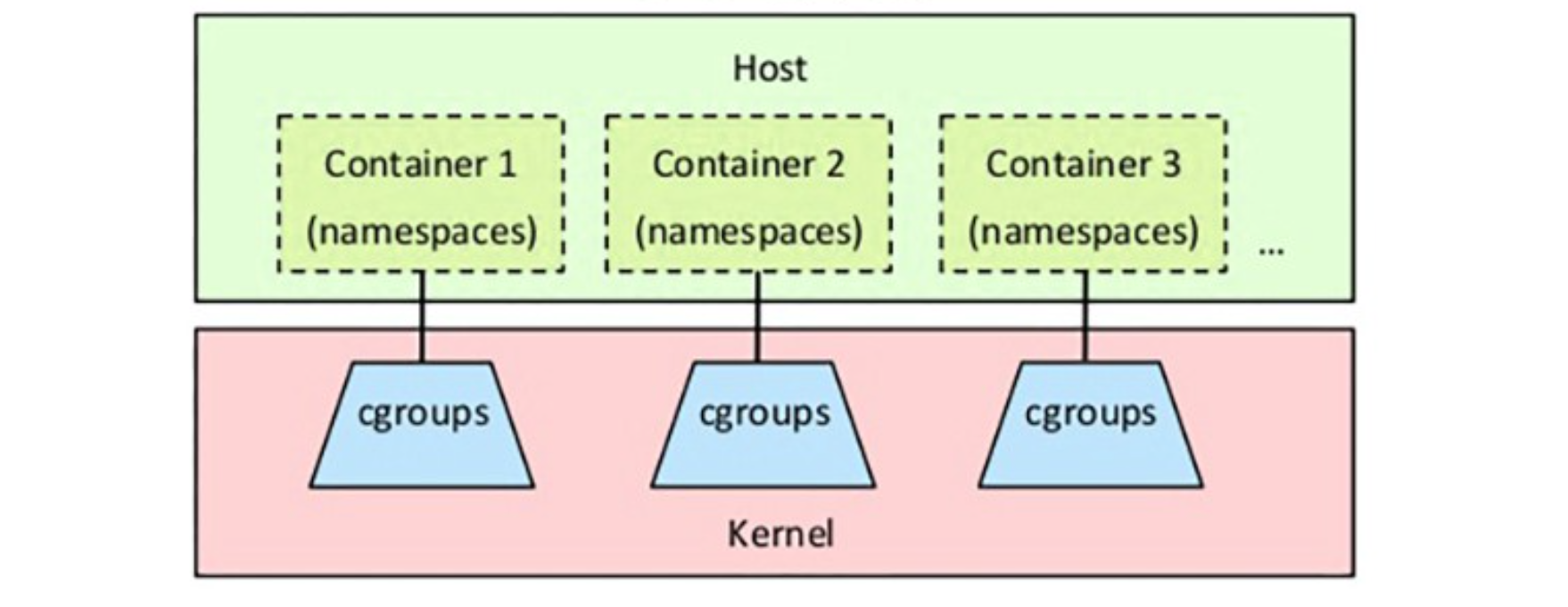
AZ03

AZ04

# Namespaces & CGroups

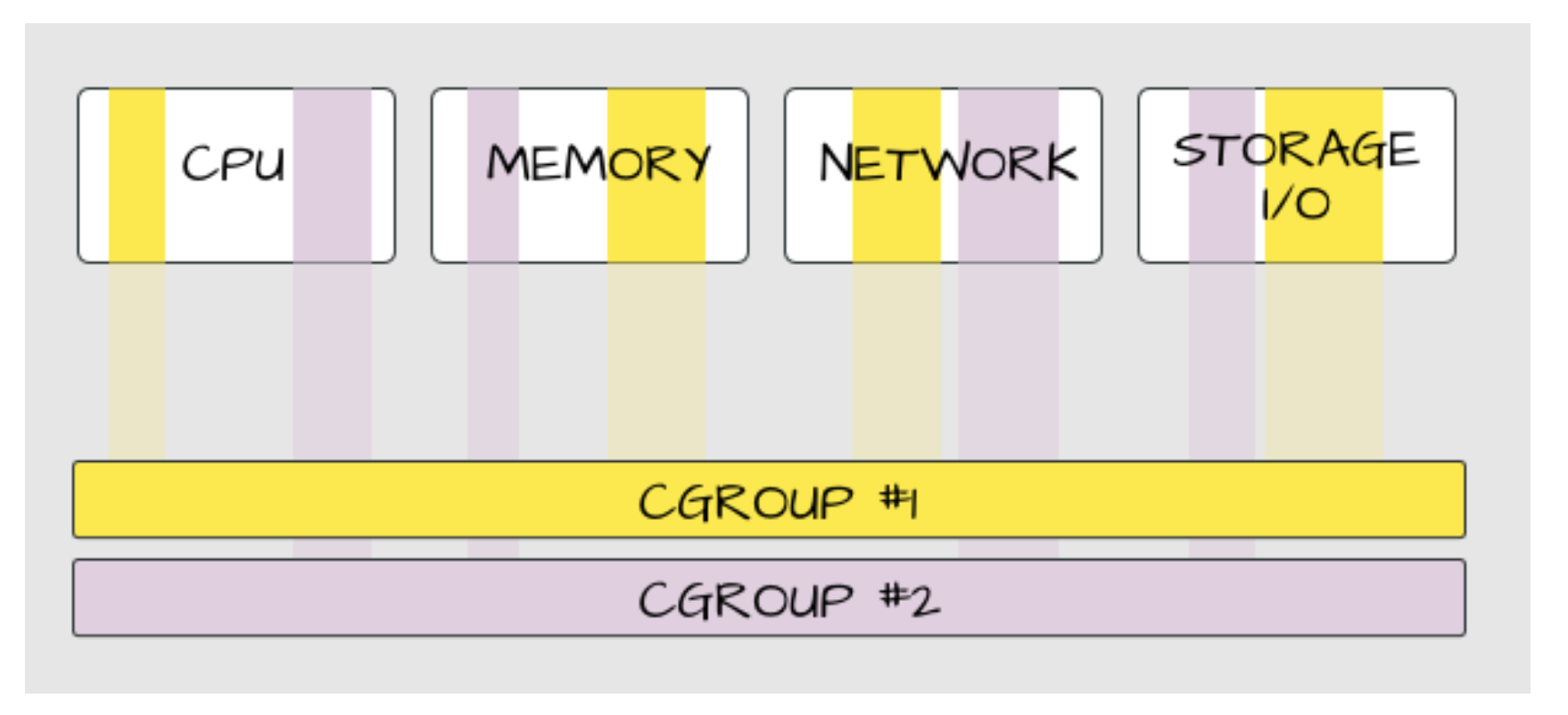


**Namespaces**



* Namespaces provide a layer of isolation for containers.
* Each aspect of a container runs in a separate namespace and its access is limited to that namespace.
* When you run a container, Docker creates a set of namespaces for that container.
* Namespace makes processes running inside that namespace believe they have their own instance of that resource.
* A namespace can limit visibility to certain process trees, network interfaces, user IDs, or filesystem mounts.

**CGroups**



* A control group (cgroup) is a Linux kernel feature that limits an application to a specific set of resource usage (CPU, memory, disk I/O, network, and so on)
* Control groups allow Docker Engine to share hardware resources to containers and optionally enforce limits and constraints.
* For example, you can limit the memory available to a specific container.

Cgroups involve resource metering and limiting:

Memory, CPU, Storage I/O, Network

# Docker Commands:

docker version

docker --help

Images: Pull & View  
docker pull ubuntu  
docker images  
docker pull ubuntu:20.04  
docker images

Images: Remove & Verify

docker rmi ubuntu

docker rmi ubuntu:20.04  
docker images

Containers: Start & Verify

docker run nginx

docker ps

docker ps -a

Containers: Remove & Verify

docker rm <container-id>

docker ps -a

Containers: Multiple containers using same image

docker run -dt --name nginx -p 80:80 nginx

docker run -dt --name **nginx81** -p **81**:80 nginx

Containers: Output through host port – 80 and 81

curl <http://localhost:80>

curl <http://localhost:81>

Containers: Stop, Start and Remove

docker stop nginx

docker start nginx

docker stop nginx

docker rm nginx

docker ps -a

Getting inside a container in Detached mode,  
docker exec -it nginx81 /bin/sh  
  
View logs of a container in Detached mode,  
docker logs nginx81  
  
Note: To use docker in $(user) prompt,  
sudo usermod -a -G docker $USER  
stop and start the Lab.