IBM Cloud-IKS(public)
HITAM

Arashad Ahamad

Software Designer
IBM-ISL Hyderabad
arahamad@in.ibm.com
arashad.ahamad@gmail.com

Agenda:

1- Containers, Docker and Kubernetes

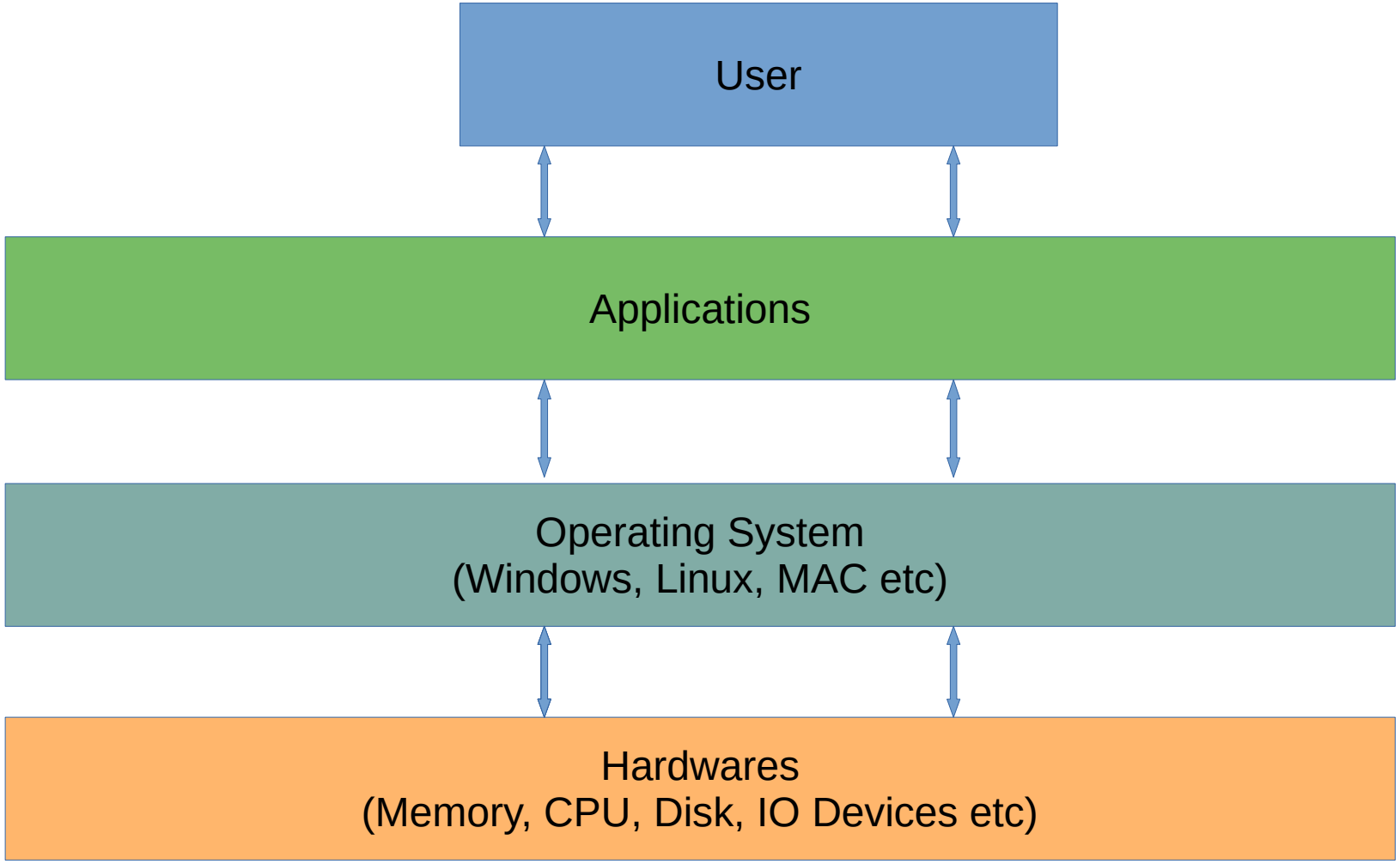
2- IBM Cloud and IKS

3- Demo on IKS

Expected outcome: After this session students will be aware about basic understanding about cloud technologies and IBM IKS

Back Ground:

Operating System:



Features:

- Resource Management
- File Management
- Process Management
- Memory Management
- Security Management

Process:

Some very basic facts:

- Running entity on OS.
- Process layout
- PCB
- Sources limits

Key linux features:

- Cgroup
  Limit/manage resources

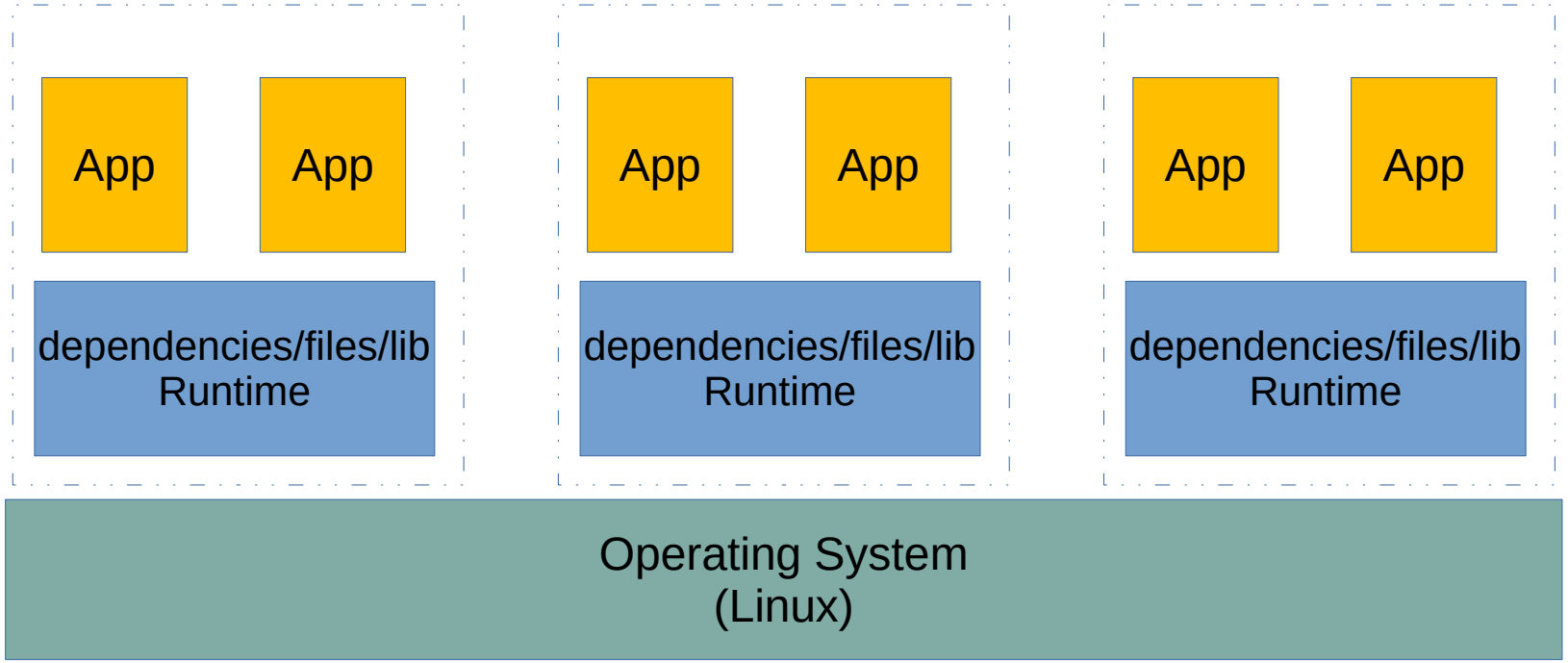
- Namespaces
  limits resources visibility

  pid, net, mnt, ipc etc

  NOTE: kernel >= 2.6.24 after 2008

Linux Containers:

Set of one or more processes that are isolated from the rest of the system.



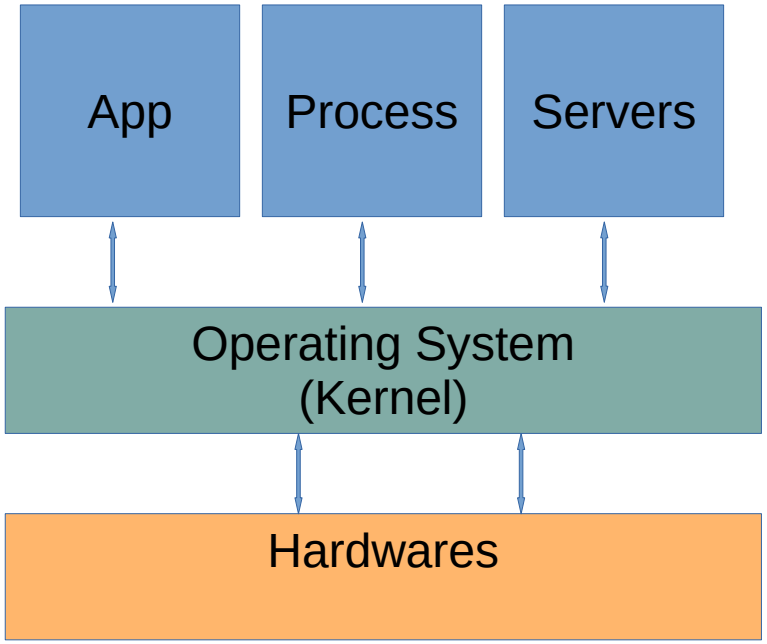
- Abstraction at the app layer
- Very less in size
- Portable and consistent
- Same operating system kernel
- Arch dependent
- Image:
  - runtime, libs, apps

Steps:

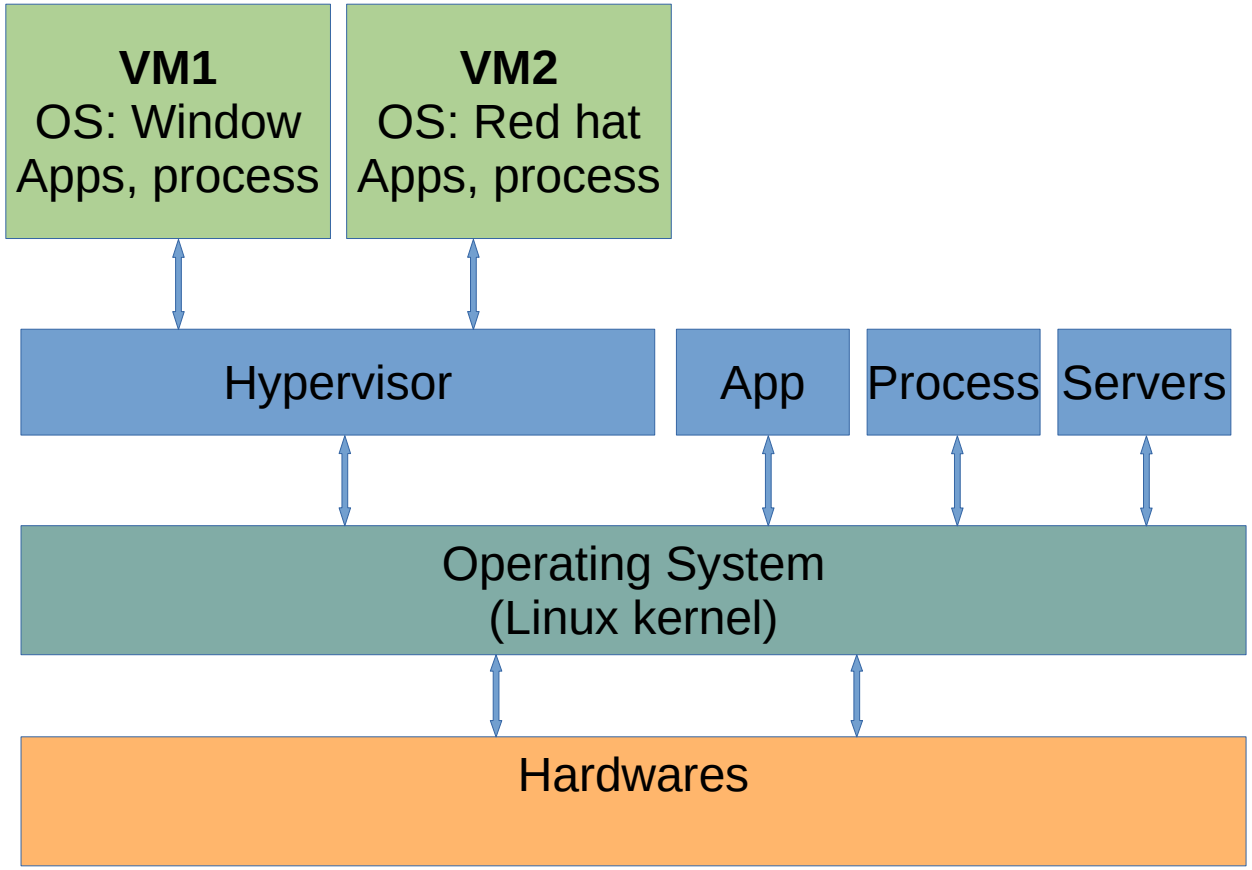
Template-> ls -1 /usr/local/share/lxc/templates
Create-> lxc-create -n myContainer -t /usr/local/share/lxc/templates/lxc-centos
Start-> lxc-start -n myContainer
Info-> lxc-info -n myContainer
Delete-> lxc-destroy -n myContainer
Connect-> lxc-console -n myContainer

Virtual Machine:

Virtual machine (VM) is an emulation of a computer system.



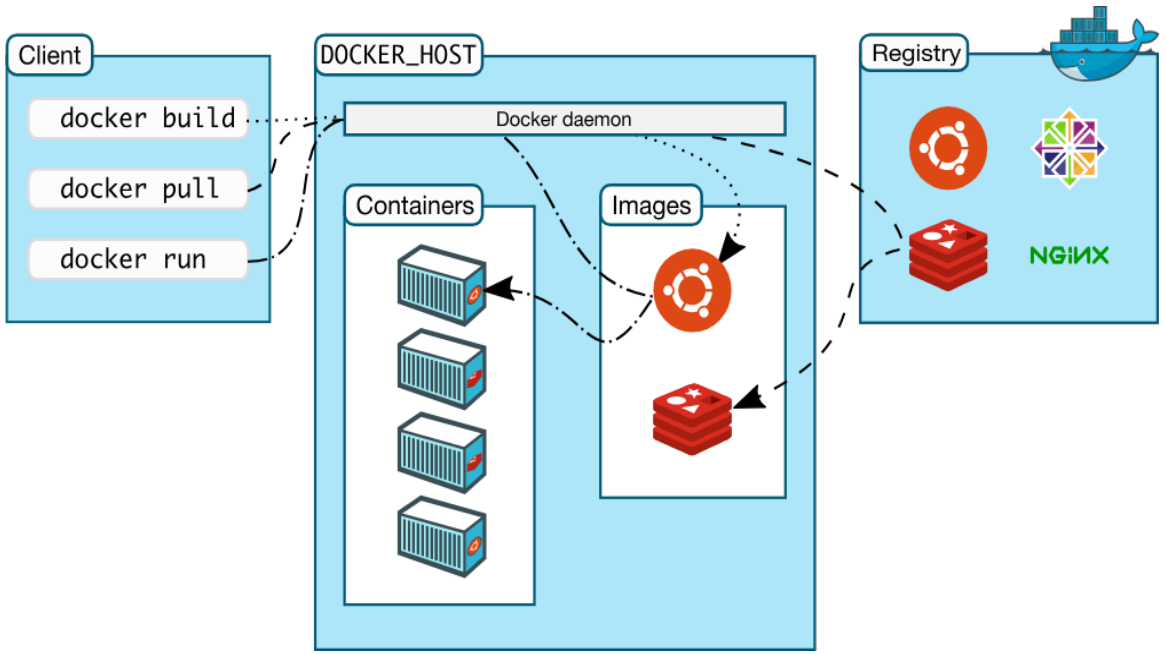
No Virtualization

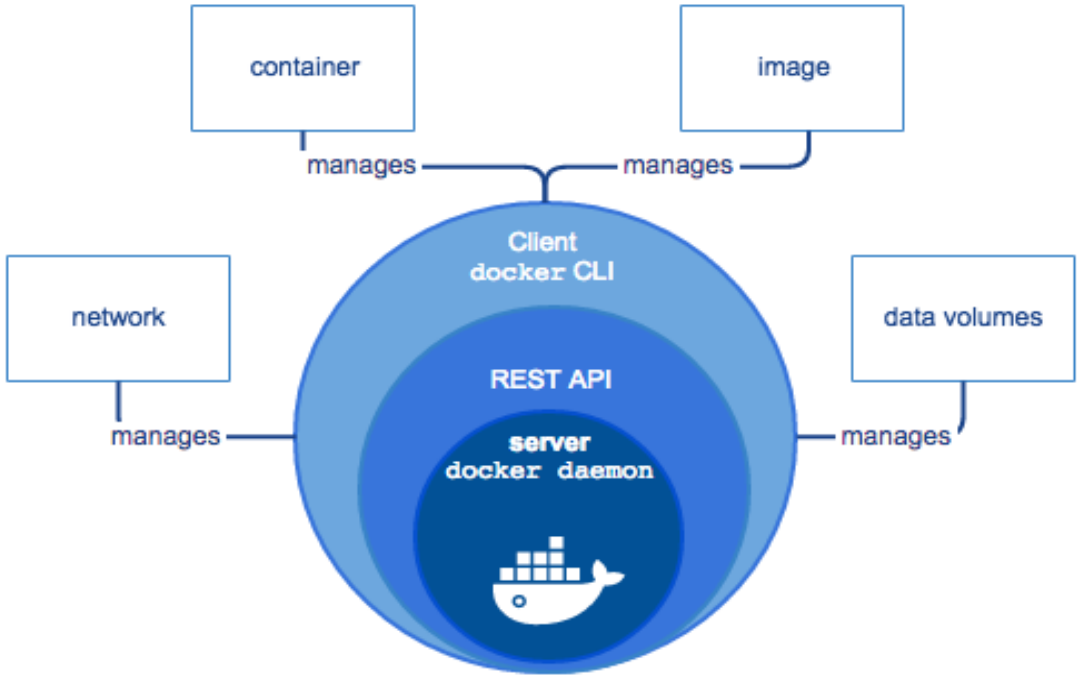


Virtualization

Docker:

- Open platform for developing, shipping, and running applications
- Create and manage containers
- Docker containers can be run on VM



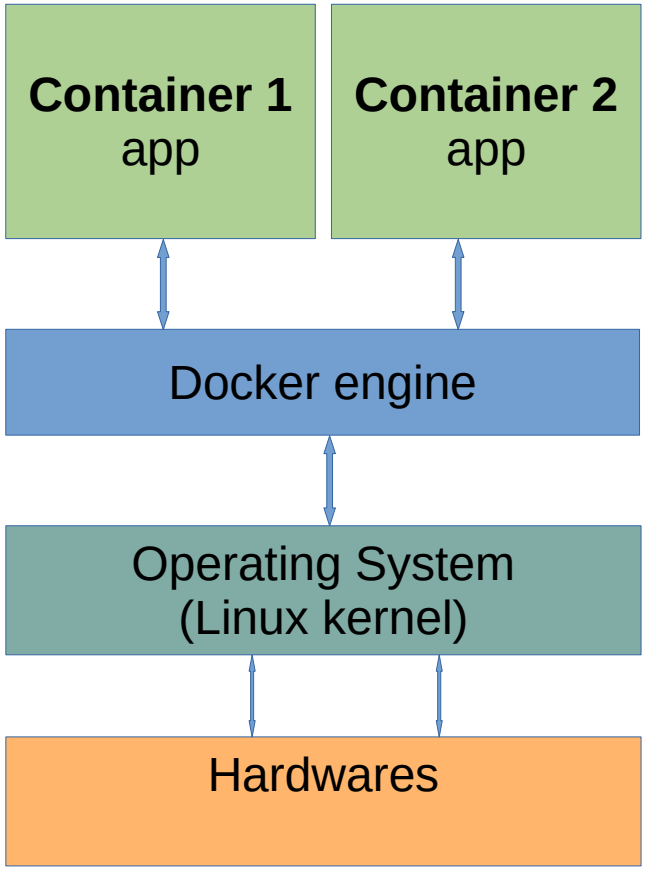


Docker Commands:

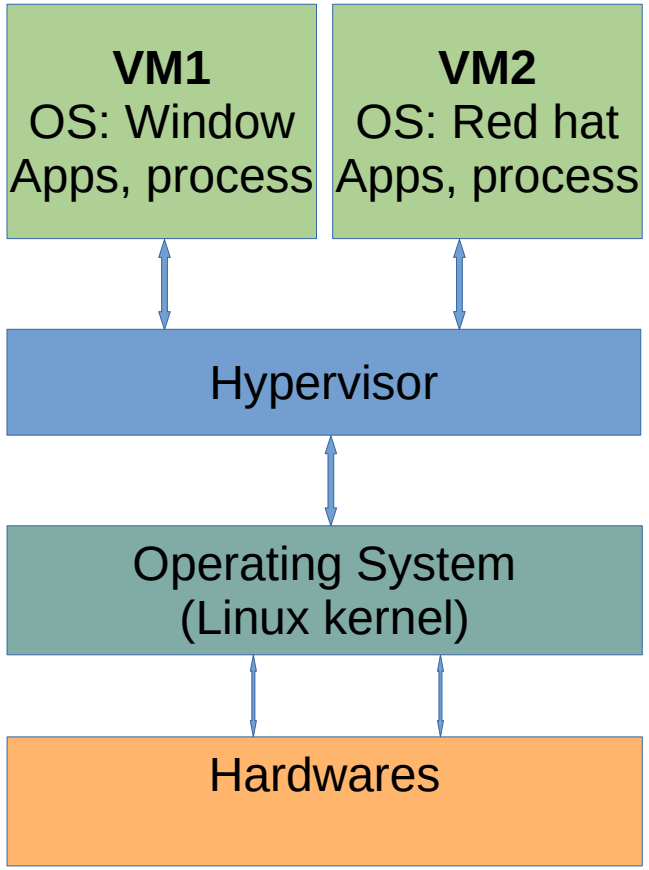
- **Create container**
  $docker create --name mycontainer -t -i ubuntu bash
- **Rename container**
  $docker rename CONTAINER NEW_NAME
- **Run container**
  $docker run --name myruncon -it ubuntu
- **Remove container**
  $docker rm --force myruncon
- **Start container**
  $docker start mycontainer
- Stop **container**
  $docker stop mycontainer
- **Pause container**
  $docker pause mycontainer
- **Attach container**
  $docker attach mycontainer
- **Logs**
  $docker logs mycontainer
- **Running Process in container**
  $ docker top mycontainer
- **Copy files to/from container**
  $ docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|
  $ docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH
- **Execute a command on container**
  $docker exec -it mycontainer bash

- **Docker images**
  $docker images
- **Image build**
  $docker build .
  $docker build -t my/myimage:2.0 .
- **Remove image**
  $docker rmi <image name>
- **Save image to tar file**
  $docker save my_image:my_tag | gzip > my_image.tar.gz
- **Image push**
- $docker push my_image:my_tag
- **Image pull**
  $docker pull

Docker VS Virtual Machine:



Docker

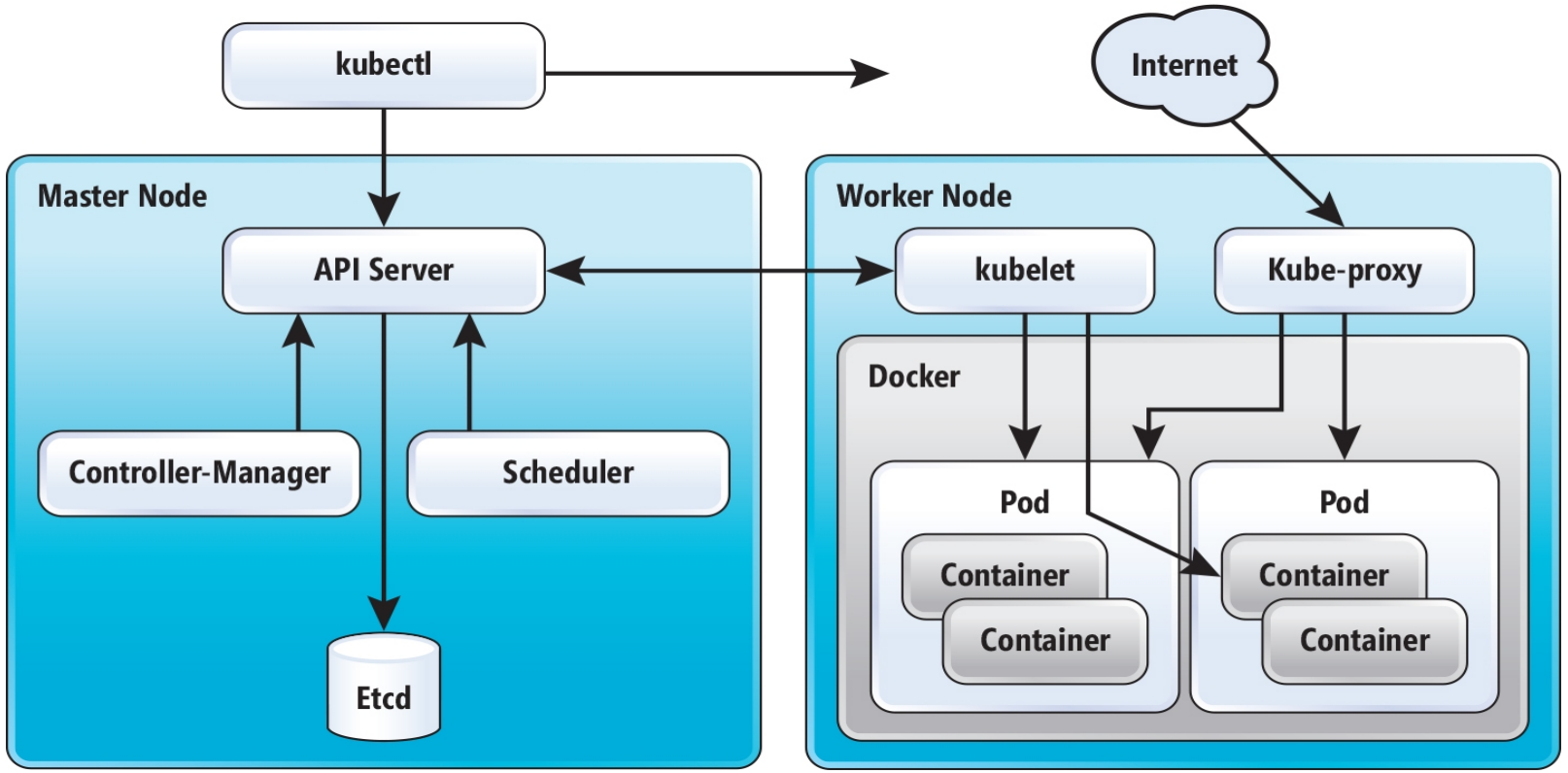


Virtual Machine

Kubernetes:

Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services.

Kubernetes is an orchestration tool for containers.

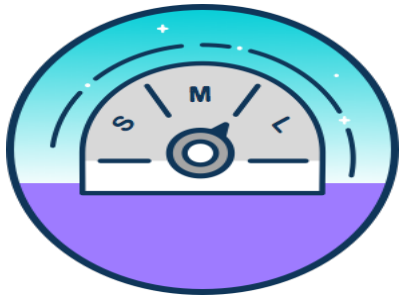


- Cluster management
- Container security and isolation policies
- Design your own cluster

Kubernetes Capabilities:

Intelligent Scheduling

Self-healing



Horizontal scaling

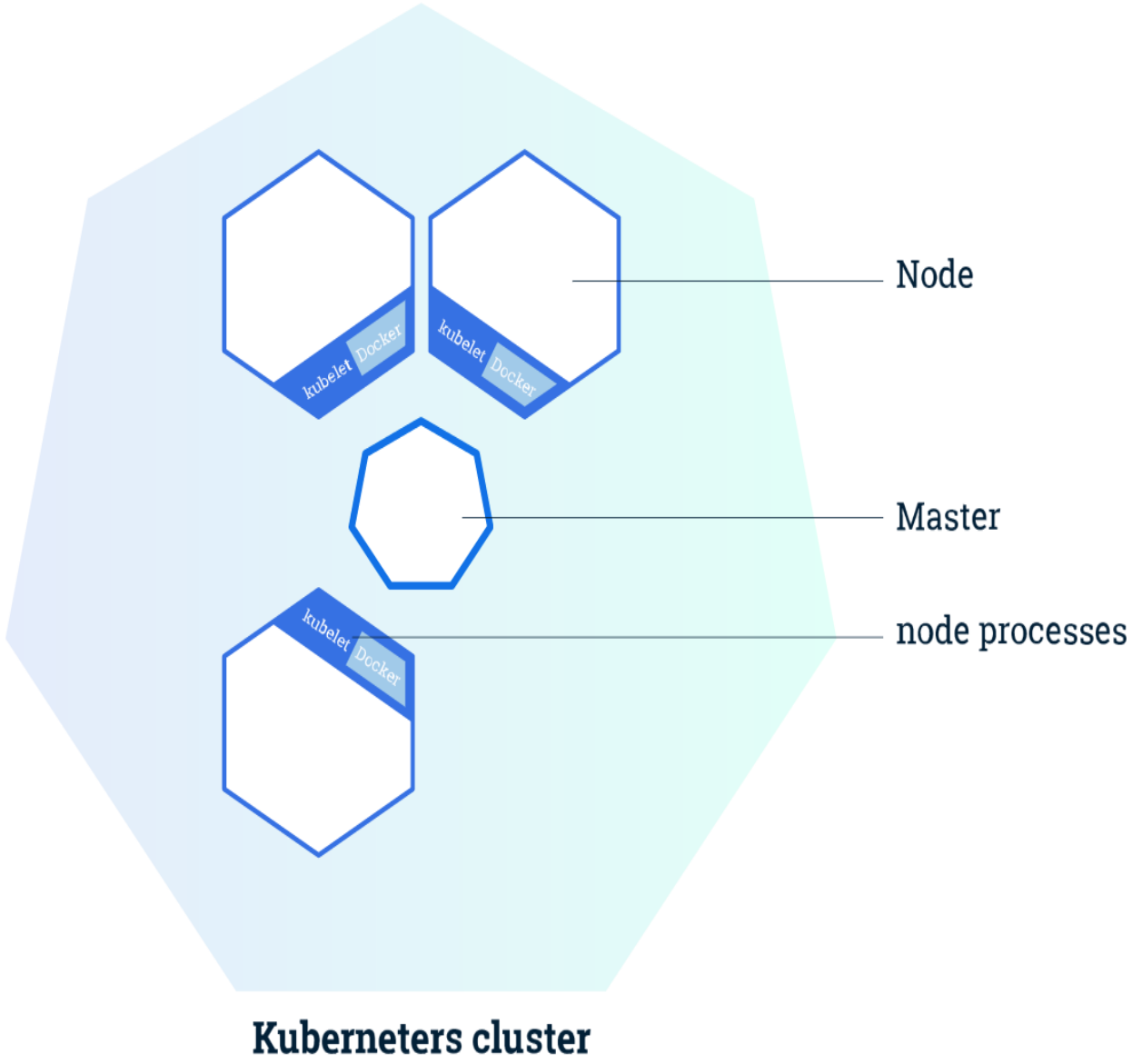
Service discovery & load balancing

Automated rollouts and rollbacks

Secret and configuration management

Kubernetes Clusters:

- highly available cluster of interconnected computers
- **Master-** co-ordinates the cluster
- **Node/Worker -** application run on this worker



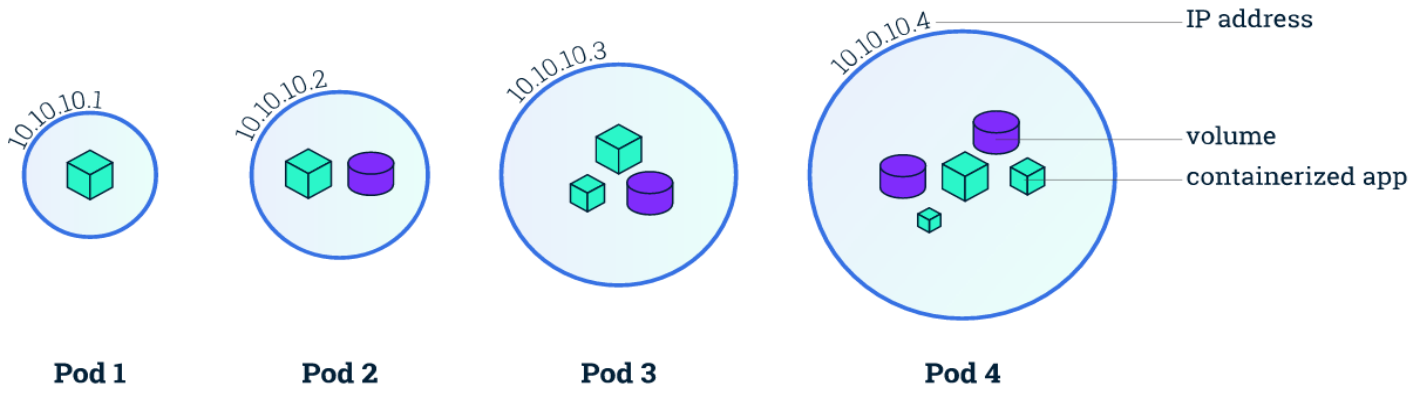
Kubernetes POD and Node:

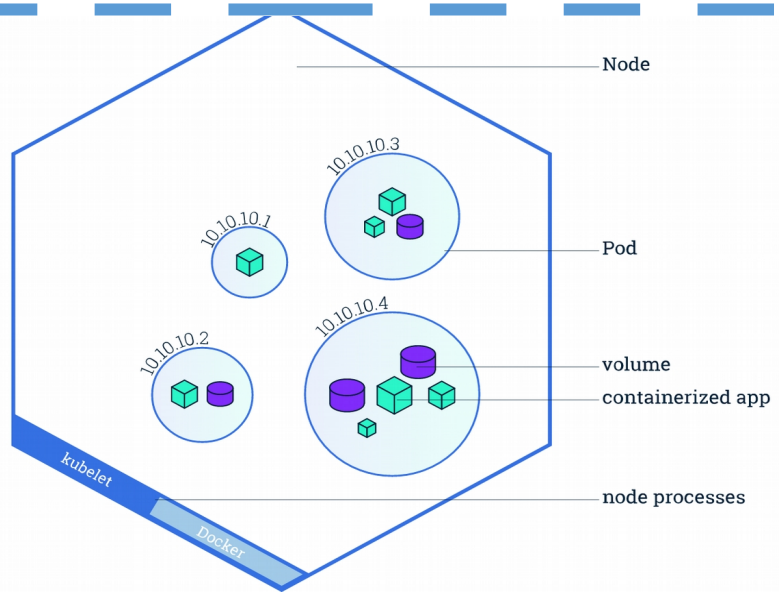
POD

- Group of one or more tightly coupled containers and shared resources
  - Shared storage, as Volumes
  - Networking, as a unique cluster IP address
  - Container image,ports
- Atomic and smallest logical unit on k8s platform
- POD deployed on single node
- Unique Ip across the cluster

Node

- Worker Machine in k8s network
- VM or Physical Machine
- Node Components
  - Kublet – resposible for communication between master and node
  - Container runtime -docker





Docker-cli	Kubectl-cli
docker run -d --restart=always -e DOMAIN=cluster --name nginx-app -p 80:80 nginx	kubectl run --image=nginx nginx-app --port=80 --env="DOMAIN=cluster kubectl expose deployment nginx-app --port=80 --name=nginx-http
docker ps	kubectl get pod|deployment|service
docker exec <cont-id> cat /etc/hostname	kubectl exec <pod-name> -- cat /etc/hostname
docker logs –f <cont-id>	kubectl logs –f <pod-name>
docker stop|rm <cont-id>	kubectl delete deployment-name|service-name nginx-app Notice that we don't delete the pod directly. With kubectl we want to delete the Deployment that owns the pod. If we delete the pod directly, the Deployment will recreate the pod.
docker info	kubectl cluster-info

5- About IBM Cloud and IKS(PaaS) (with demo)

1- IBM Cloud

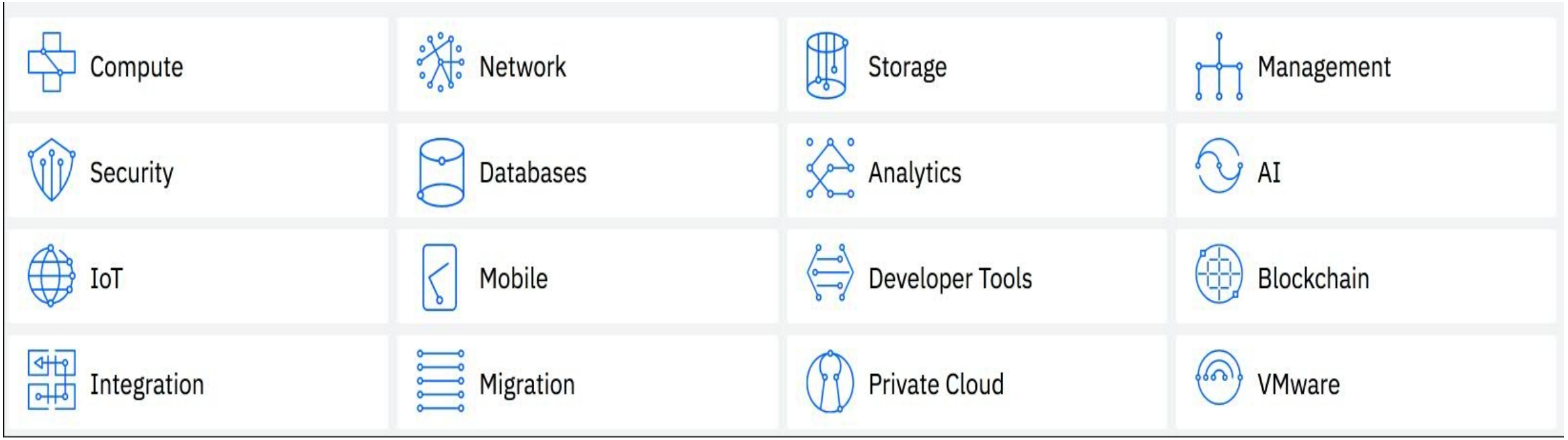
2- IKS

3- CLI installation and cluster creation

IBM Cloud:

A full-stack cloud platform that spans public, private and hybrid environments. Build with a robust suite of advanced data and AI tools, and draw on deep industry expertise to help you on your journey to the cloud.

A full-stack cloud platform with over 170 products and services covering data, server-less, containers, AI, IoT and blockchain.



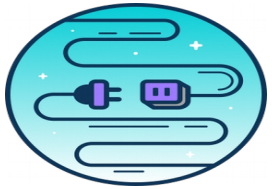
IKS:

IBM Cloud™ Kubernetes Service is a managed container service for the rapid delivery of applications that can bind to advanced services like Watson™ and blockchain.

Simplified Cluster Management

Container Security & Isolation

Design Your Own Cluster



Leverages IBM Cloud & Watson

Native Kubernetes Experience

Integrated Operational Tools

CLI installation and cluster creation :

https://cloud.ibm.com/docs/containers?topic=containers-container_index#container_index?cm_sp=Cloud-CloudDevSvc-_-OnPageNavCTA2-IBMCloud Platform_IBMContainers-_-PDP

https://cloud.ibm.com/registration?target=%2Fcontainers-kubernetes%2Fcatalog%2Fcluster%2Fcreate&cm_sp=Cloud-CloudDevSvc-_-OnPageNavCTA1-IBMCloud Platform_IBMContainers-_-PDP

OR
Step 1- Search “IBM Cloud and IKS” on google
Step 2- Click on first searched item
Step 3- Click on `Get started with clusters` button

IBMCloud cli installation:

https://cloud.ibm.com/docs/cli?topic=cloud-cli-ibmcloud-cli#ibmcloud-cli

Setting up cluster:

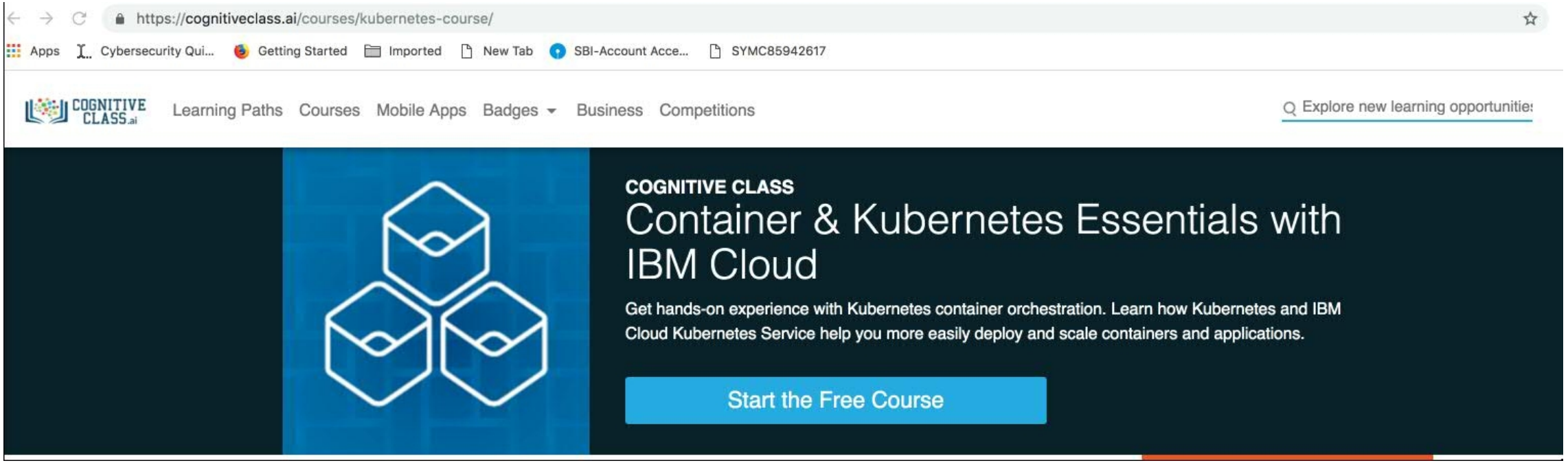
https://cloud.ibm.com/docs/containers?topic=containers-clusters#clusters_cli

Characteristics	Lite clusters	Standard clusters
Available in Bluemix Public	✓	✓
In-cluster networking	✓	✓
Public network app access by a NodePort service	✓	✓
User access management	✓	✓
Bluemix service access from the cluster and apps	✓	✓
Disk space on worker node for storage	✓	✓
Persistent NFS file-based storage with volumes		✓
Public or private network app access by a load balancer service		✓
Public network app access by an Ingress service		✓
Portable public IP addresses		✓
Available in Bluemix Dedicated (Closed Beta)		✓

Table 1. Differences between lite and standard clusters

Free Course, you can also get Badge for this course:

https://cognitiveclass.ai/courses/kubernetes-course/



Thank you!!!

Questions/Answers