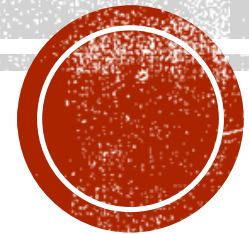
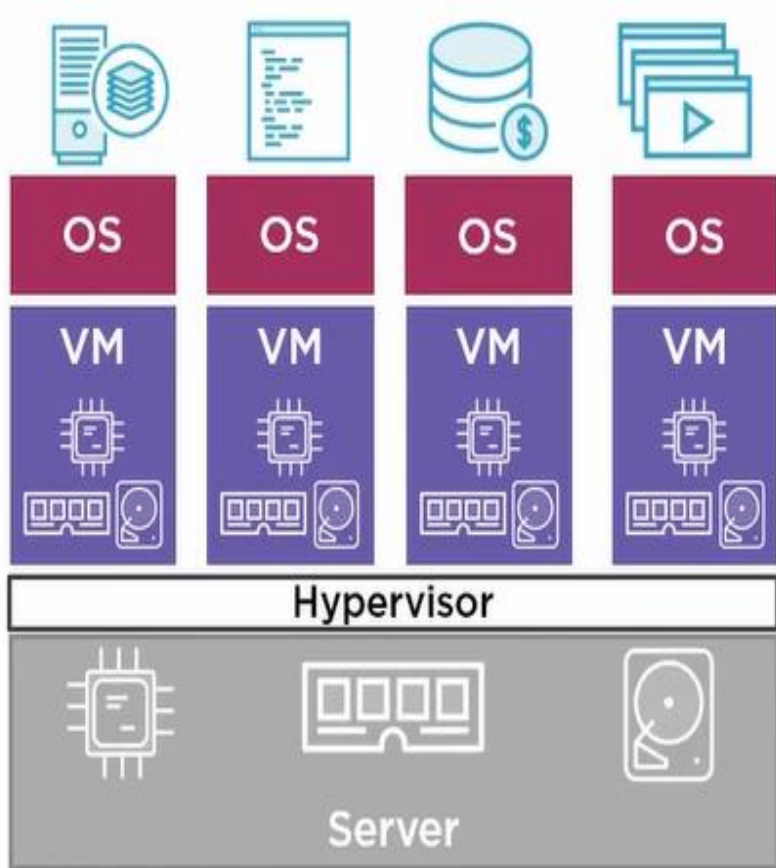


# DOCKER CONTAINERS

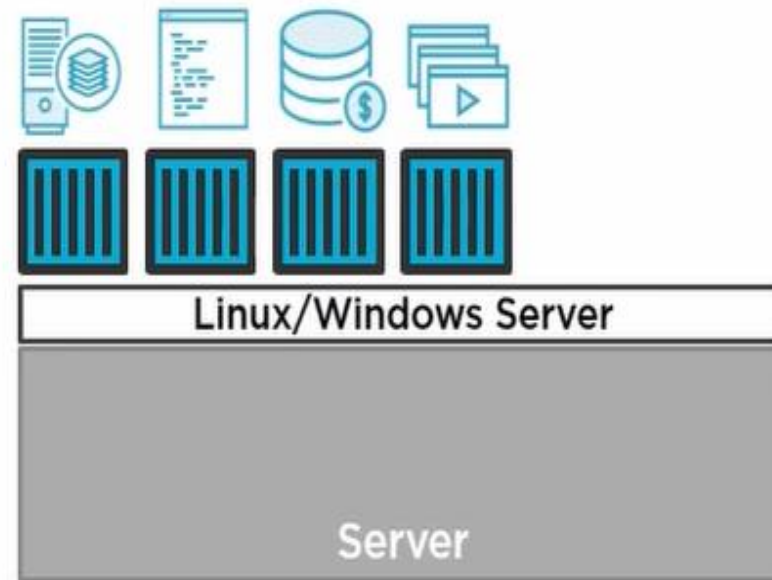
An Introduction



# COMPARING DOCKER TO VMS

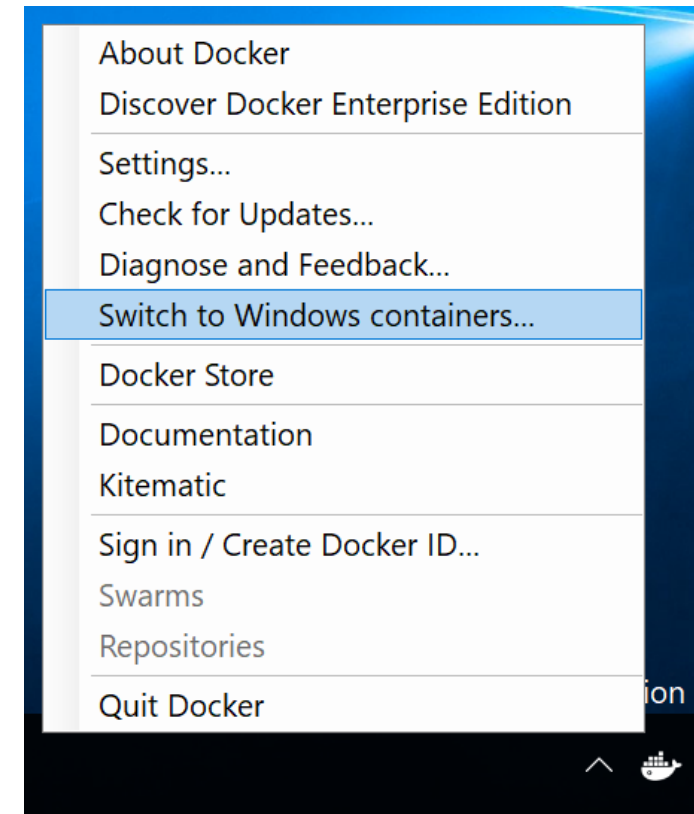


- Each VM on a server gets a slice of the server's hardware resources, creating virtual CPUs, disk drives etc. On top of that, each VM needs an OS – this is a waste of server resources
- Docker, on the other hand, runs one OS for many containers. They are like lightweight, really fast VMs running containerized apps



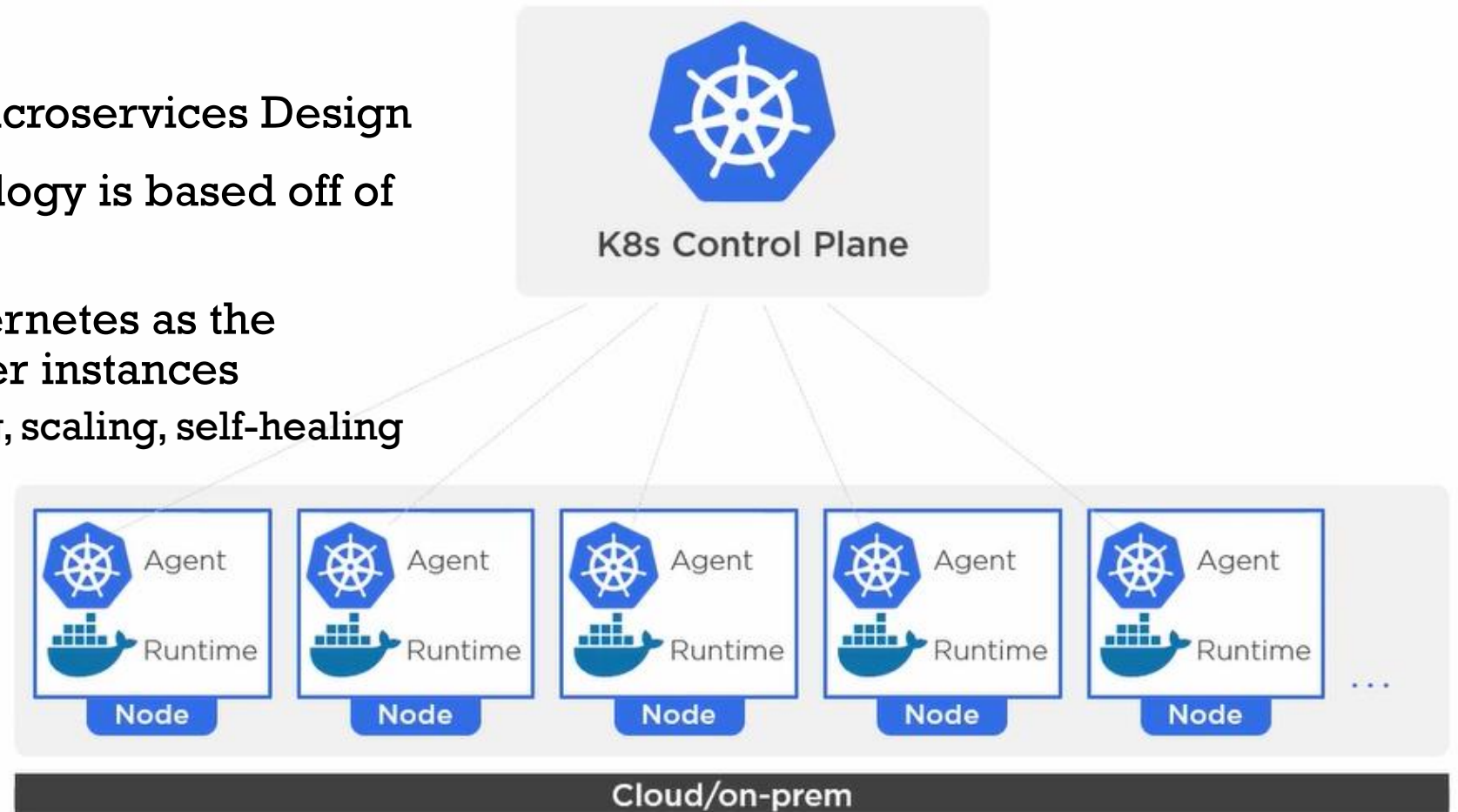
# DOCKER DESKTOP FOR WINDOWS

- On a Windows machine, you have the choice of running Linux apps on a Linux OS or Windows apps on a Windows OS
- Using Docker Desktop you can toggle your default/active container to Windows or Linux
- We will be using the CE, or Community Edition, of Docker Desktop



# DOCKER TECHNOLOGY

- Uses Cloud Native Microservices Design
- Its foundation technology is based off of **Kubernetes**
- You can think of Kubernetes as the conductor of container instances
  - Scheduling, updating, scaling, self-healing



# DOCKER IMAGES

- Images can be conceptualized as prepackaged application – everything that it requires to run is bundled in the image
- Images can be pulled from the [Docker Hub](#)
  - You will require a sign in!
- Downloading an image
  - Use the docker [pull command](#)
- You can view which images you have on your machine by running this console command: **docker image ls**
- To remove an image, use:
  - **docker rm <image name or id> [-f]**
  - You may have to use the force option if an image is running in a background container



# RUNNING AN IMAGE IN A CONTAINER

- We use the run command:  
**docker run [OPTIONS] IMAGE [COMMAND] [ARG...]**
- When you execute the run command, 2 things happen:
  - A container is placed over the image
  - The image runs inside the container
- To view all running containers:
  - **docker ps -a**
- To stop a container:
  - **docker stop <container name or id>**
  - The default action is a SIGTERM, which means it is a graceful exit in 10 seconds, giving time for other processes/resources to release. You can change the time period for releasing resources.



# EXPERIENCE TIME

- In a short exercise, you will play with common Docker commands to run a MongoDB container
- First, let's all create a Docker account!





# DOCKER COMPOSE

- Docker allows us to easily work with multi-container docker applications. This is exactly what we need to play with a fully distributed Hadoop Cluster!
- Let's review some online help for the [docker-compose](#) command now
- Please download the **.yaml** and **.docker files** that are connected to these slides in BlackBoard. We will review them now.
- Finally, we will complete a little exercise to learn how to get our Hadoop Cluster running using Docker

