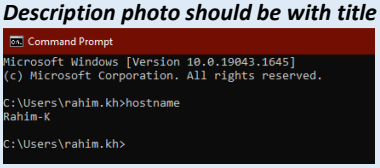
	Document name / number / date:	Lecture	2	20.05.2022

DOCUMENT RULES:	
Lecture are consisting of modules and sub modules	**1. Lectures** **a. Sub lectures** **b. Sub lectures**
Task name & column name should be written:	**Bold (CTRL+B)**
Commands should be written in the after # sign:	*Italic (CTRL+I) #hostname*
Output photo should be cropped or compressed: **Photo could be more than one:** **If you need extra lines, add the line next after it:**	***Description photo should be with title bar (CTRL + I + B)*** 
All other text should be written:	Standard
Font name and text size:	Calibri and 9
Introduction to Kubernetes	4. Kubernetes introduction 5. Containers Introduction 6. Kubernetes setup 7. Local setup with minikube 8. Demo: minikube 9. Installing Kubernetes using the Docker Client 10. Minikube vs Docker Client vs Kops vs Kubeadm 11. Introduction to Kops 12. Demo: Preparing kops install 13. Demo: Preparing AWS for kops install 14. Demo: DNS Troubleshooting (Optional) 15. Demo: Cluster setup on AWS using kops 16. Building docker images 17. Demo: Building docker images 18. Docker Image Registry 19. Demo: Pushing Docker Image 20. Running first app on Kubernetes 21. Demo: Running first app on Kubernetes 22. Demo: Useful commands 23. Service with Load Balancer 24. Demo: Service with AWS ELB Load Balancer 25. Recap: introduction to Kubernetes Practical test?

4. Kubernetes introduction

What is Kubernetes?

- Kubernetes is an open-source **orchestration** system for Docker containers
    - It lets you schedule **containers** on a cluster of machines
    - You can run **multiple containers** on one machine
    - You can run long running **services** (like web applications)
    - Kubernetes will **manage** the state of these containers
        - Can start the container on specific nodes
        - Will restart a container when it gets killed
        - Can move containers from one node to another node

- Instead of just running a few docker containers on one host manually, Kubernetes is a platform that will manage the containers for you
- Kubernetes clusters can start with one node until thousands of nodes
- Some other popular docker orchestrators are:
    - Docker Swarm
    - Mesos Learn

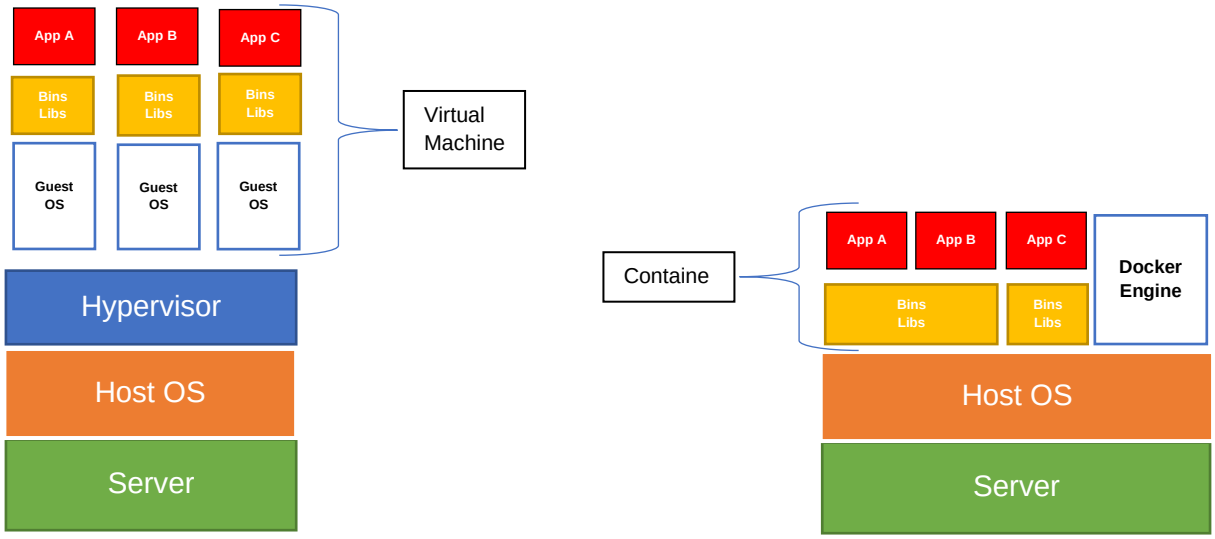
Kubernetes Advantages!

- You can run **Kubernetes** anywhere:
    - On-premises (own datacenter)
    - Public (Google cloud, AWS)
    - Hybrid: public & private (high powerful)
- Highly modular (make changes if necessary)
- Open source (project on git hub changes make easily and every one can connect to proj)
- Great community
- Backed by Google (Google released and uses containerized own system over 10 years)

5. Containers Introduction

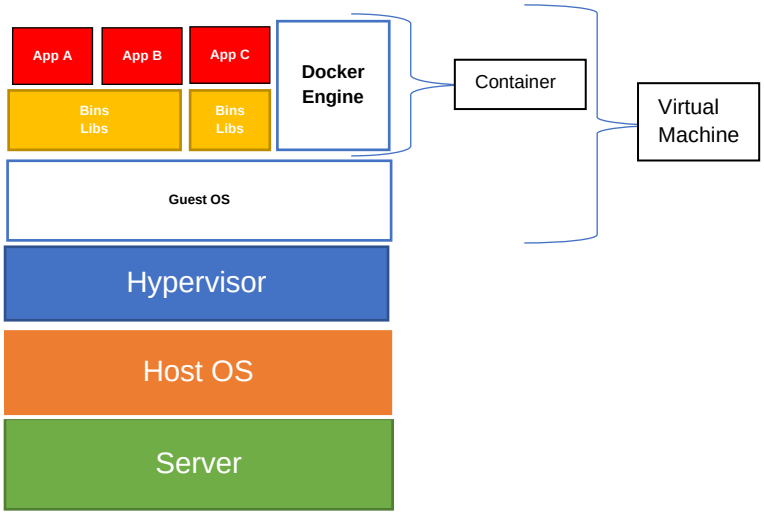
Virtual Machines vs Containers

Left side you see physical machine, Host OS, Hypervisor (it manages multiple Guest Operating System), each Guest VM are isolated from each other and there are isolated own binaries (bins), own libraries (libs), and own applications. Their whole bootup takes over one minutes.
Right side you see Physical Server, Host OS, Multiple containers on host operating system using Docker Engine. Docker is just one example of software and there is other software are available for containers as well. Here we have three containers contain binaries, libraries with applications in them. They don't need Boot up the full guest operating system so we lunch containers instead of waiting minutes it was instantly starts Containers. They are so lightweight. Containers size measuring with megabytes.



Containers on Cloud Providers!

Containers on the Clouds providers are a bit different because Providers still using hypervisors to isolate customer from each other. There is still server, host operating system, hypervisor then Guest Operating system then containers. You will use all benefits of containerization even on cloud. There is extra abstraction layer with your containers and operating systems. There are multiple guest operating systems and multiple container types.



Docker

- **Docker** is the most popular container software
  - An alternative to Docker is rkt (rocket) - which also works with Kubernetes
- Docker **Engine**
  - The Docker runtime
  - Software to make run docker images
- Docker **Hub**
  - Online service to **store** and **fetch** docker images *(so once you built image and you can store it on the internet using docker Hub. Public Private repositories)*
  - Also allows you to **build docker** images online *(in your PC and in Online) (Clouds have alternatives to the Docker Hub. For instance, AWS have own repository where you can save your docker images. You can also run your own docker hub, you can run your own repository if you would like to:)*

Docker Benefits

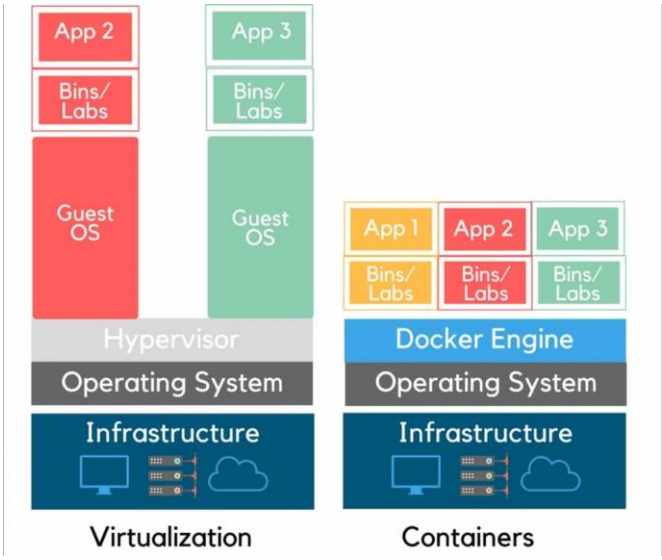
- **Isolation**: you ship a binary with all the dependencies

  - no more it works on my machine, but not in production
- Closer **parity** between dev, QA, and production environments *(because you are using same binary)*
- Docker makes development teams able to ship **faster** *(we can build images quick, you can them distribute them easily using docker hub or another registry and then servers can start pulling those images. This would be a lot more cumbersome with virtual machine images than with container images)*
- You can run the same docker image, unchanged, on laptops, data center VMs, and Cloud providers. *(so, you can have same image that you run on premise and in the cloud, and should act exactly the same)*
- Docker uses Linux Containers (a kernel feature) for operating system-level isolation *(Docker use Linux containers, which is a kernel feature for operating system level isolation. The reason why containers cannot go to the whole system or to containers or you don't see information)*

  It means containers to system
  System to containers
  Containers to containers isolation.

Containerization

If you have read something about containers, you probably have seen this analogy already. The containers contain all of dependencies, so it should execute exactly same, whether you can run it locally, in the cloud, on the Kubernetes, or anywhere you want to run it and that is the main advantage of using containers.



6. Kubernetes setup

The difference ways of setting up Kubernetes

Kubernetes setups

- Kubernetes should really be able to run **anywhere**
- But there are more **integrations** for certain Cloud Providers, like AWS & GCE
    - Things like **Volumes** and **External Load Balancers** work only with **supported** Cloud Providers *(so, if the integration for your platform is not available then you can still use **Kubernetes,** but you will not able to use **Volumes** and **External Load Balancers. "AWS, GCE and Azure" have but for other cloud providers you must look at the documentation and the cloud provider plugins to see which ones are supported.**)*
- I will first use **minikube** to quickly spin up a local single machine with a Kubernetes cluster *(you can use **minikube** on your laptop or your desktop to spin up a Kubernetes cluster)*
- I'll then show you how to spin up a cluster on AWS using **kops**
    - This tool can be used to spin up a highly available **production cluster**
- Doing the labs, yourself is possible *(and highly recommended):*
- Using the AWS Free tier (gives you 750 hours of t2.micro's / month) *(it is not just one t2.micro, it's 750 hours per month. So, you can run multiple t2.micro's as long as you don't exceed the number of 750 hours per month.)*
    - http://aws.amazon.com *(The URL to sign up is:)*
- Using your local machine *(you can do same labs using your local machine, you will not be able to test integrations with cloud.*
    - Use minikube from https://github.com/kubernetes/minikube *(you can still use Kubernetes, you can use minikube from this URL)*
- Using DigitalOcean (if you want to use Digital Ocean)
    - Use https://cloud.digitalocean.com/

7. Local setup with minikube
Set up Kubernetes locally

Minikube Setup

- **Minikube** is a tool that makes it easy to run Kubernetes locally
- Minikube runs a **single-node** Kubernetes cluster inside a Linux VM (Node is Kubernetes Object)
- It's aimed on users who want to just test it out or use if for development
- It cannot spin up a production cluster, it's a one node machine with no high availability
- It works on Windows, Linux, and MacOS (It works all of OS)
- You will need Virtualization Software installed to run minikube:
    - VirtualBox is free and can be downloaded from www.virtualbox.org *(for instance, you can use Virtual Box, which is free)*

- You can download **minikube** from https://github.com/kubernetes/minikube *(just copy paste it in your web browser)*
- To launch your cluster, you just need to enter (in a shell / terminal / powershell or CMD) *(Linux, MAC, Windows and minikube starts, if you are using Windows, you first to go in the directory where you download minikube and then you start a minikube, with this **minikube start** comman)*

$ minikube start

8. Minikub

Bu sinifdə olacaq

Let's start demonstration on Lab
Step by step: Workshop 1
Name: (Local Kubernetes setup using minikube)