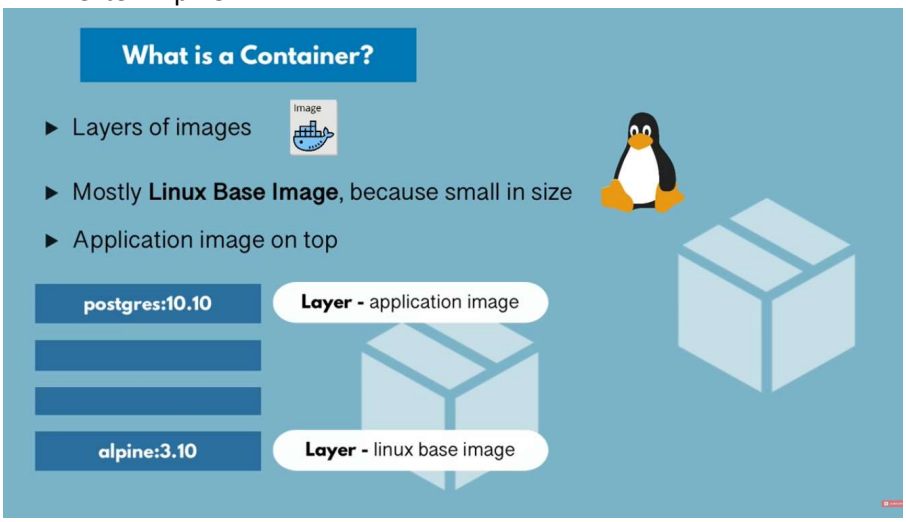
Docker

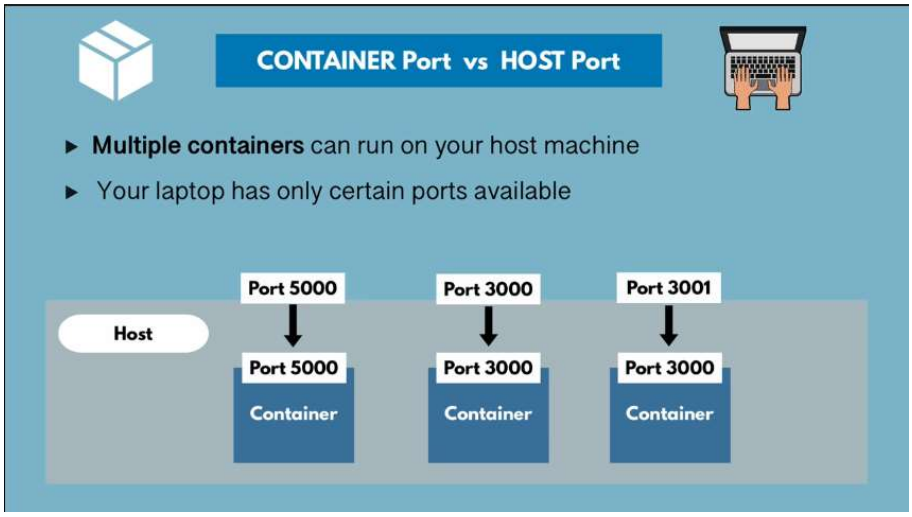
22. januar 2022 14:31

- **Container**
  - Way to package application with all dependencies and configurations
  - Ensure applications always run in the same environment
    - Makes portable artifacts that can be spun up easily on different machines
      - Makes deployment easier since versioning and configuration is already done in container.
    - Installation process is same for all machines
  - Can be uploaded to container repositories
    - Companies often have their own private repositories
    - Dockerhub is a public container repository
  - Layers of images
  - Usually Linux base image, because small in size
    - Often Alpine
  - 

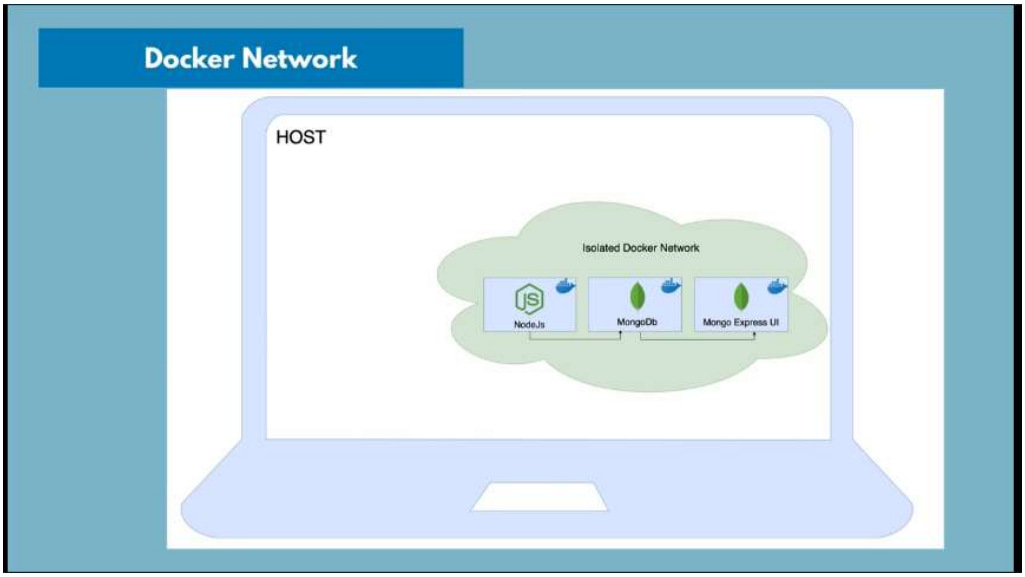
  - When containers are stopped, they lose all application data.
    - Volumes can be used to persists data

- **Image**
  - The actual package / artifact, that can be moved around.
  - Container is a running instance of an image

- **Docker vs VM**
  - Docker virtualizes the application layer of the OS and uses kernel of host
  - VM virtualizes both the OS Kernel AND the applications layer of the OS.
  - Docker image size is much smaller since it only virtualizes the app layer.
  - Docker containers start and run faster
  - Virtual machines can run on any hosts since it virtualizes the OS kernel.
    - Docker might not be able to run a linux base image on Windows kernel, depending on version.
    - Docker uses "Docker toolbox" to abstract the kernel to enable docker to run on any platform

- 

- 

- **Docker Network**
  - When docker runs, it creates it own network
  - Containers in the network can communicate with each other using only their container names
  - 

Commands

23. januar 2022 15:17

- **Docker ps**
  - Shows all processes (containers)
  - -a
    - Shows all stopped and running containers

- **Docker run IMAGE_NAME**
  - Runs an image
    - e.g. Docker run postgres
      - Spins up a container with postgres
      - Can also do Docker run postgres:SOME_VERSION, which does Docker pull and then docker start for you
    - -d
      - Run in (d)etatched mode (Indepedendent from the Terminal process)
    - -p3001:3000
      - Specifies the (p)ort binding
      - Binds port 3001 on host machine to port 3000 in container
      - We can bind to same container ports as long as we use different host ports
    - --name SOME_NAME
      - Specifies the container name
    - --network NETWORK_NAME
      - Specifies which docker network container should be part of
    - -e ENVIRONMENT_VARIABLE_NAME
      - Specifies (e)nvironment variables

- **Docker start CONTAINER_ID**
  - Starts a container
  - Container has retained all the options provided during the intial 'docker run' command

- **Docker stop CONTAINER_ID**
  - Stops a container

- **Docker logs CONTAINER_ID or Docker logs CONTAINER_NAME**
  - Provides the logs for a container, which can be used for debugging.

- **Docker exec -it CONTAINER_ID /bin/bash or docker exec -it CONTAINER_NAME /bin/bash**
  - Fetches an interactive terminal with root user perms for the container
  - e.g. Docker exec -it 873ds822 /bin/bash
    - Then env to check if environment variables are set correctly
  - Use exit to exit the terminal

- **Docker network ls**
  - Shows all created docker networks

- **Docker network create NETWORK_NAME**
  - Creates a new docker network

- **Docker-compose -f docker-compose.yml up**
  - Starts 'up' containers from the (-f)ile docker-compose.yml
  - -d
    - Runs docker-compose in (d)etached mode

- **Docker-compose -f COMPOSE_FILE.yml down**
  - Stops all containers specified in the docker-compose file and removes the created docker network

- **Docker build PATH_TO_DOCKERFILE**
  - Builds a new Image from 'Dockerfile'
  - -t
    - Specifies a (t)ag that identifies the new image, e.g. 'node-alpine:3.0', 'my-app' etc.

Docker-compose

23. januar 2022 17:55

- Docker-compose can be used to spin up multiple containers at once with one file
- Docker compose automatically creates a common docker network for all the containers registered under 'services'

mongo-docker-compose.yml

```yaml
version: "3"
services:
    mongodb:
        image: mongo
        ports: -27017:27017
        environment:
            - MONGO_INITDB_ROOT_USERNAME=admin
            - MONGO_INITDB_ROOT_PASSWORD=password

    mongo-express:
        image: mongo-express
        ports: -8081:8081
        environment:
            - ME_CONFIG_MONGODB_ADMINUSERNAME=admin
            - ME_CONFIG_MONGODB_ADMINPASSOWRD=password
            - ME_CONFIG_MONGODB_SERVER=mongodb
```