**Docker for Developers – LinkedIn Learning**

A person in a suit

Description automatically generated with medium confidence

Docker Overview:

A person holding his head in his hand

Description automatically generated with low confidence

How can you guarantee all developers will run the same environment, resources, let alone the same versions? Docker **sets up, install and run your resources automatically,** and more often than not, **much faster too.**

A picture containing text, screenshot, font, electric blue

Description automatically generatedA diagram of a docker components

Description automatically generated with medium confidence

A diagram of a network

Description automatically generated with low confidence

Each container is independent (no idea of dependency). Here networking comes into play by connecting them together.

DockerFile: Creates an Image based on the instruction of your DockerFile.

1. docker build -t algofields/simple-backend .

2. docker images

3. docker run -p 4000:4000 algofields/simple-backend

4. docker ps

5. docker stop [first few container ID]

6. docker kill [first few container ID] #if a container wont stop

7. docker push/pull [image tag]

A picture containing text, font, screenshot

Description automatically generatedA picture containing text, screenshot, font, number

Description automatically generated

Docker Compose: a tool to manage multiple containers and set the entire backend to a single file.

1. Docker-compose build

2. Docker-compose up -d mongo #ensure mongo is built first as DB has to exist before the app

3. Docker ps #check if mongo is running

4. Docker-compose up -d app

5. Docker ps #check if both mongo and app are running

6. Docker logs [first few container ID] #open logs for containers

7. #By this time you can launch localhost:4000

8. Docker-compose stop #stopping all containers

9. Docker ps #check nothing is no longer running

10.

Docker Swarm

A screenshot of a video

Description automatically generated with medium confidence

Adding node to swarm

1. Docker swarm init

2. #run the token on other machine/computer

3. Docker info

4. Docker node ls

5. Docker stack deploy -c docker-compose.yml

6. Docker service create --replicas 1 --name nodeserver2 node ping docker.com

7. Docker service ls

8.

Overview to Kubernetes

A picture containing text, screenshot, font

Description automatically generated

#Make sure Kubernetes, minikube is installed and Docker is running

1. Minikube start

2. Kubectl cluster-info

3. Kubectl create deployment nodeapplication2 --image=[node/URL]

4.

Continuous Integration (CI) and Deployment Workflow

A screen shot of a diagram

Description automatically generated with low confidence