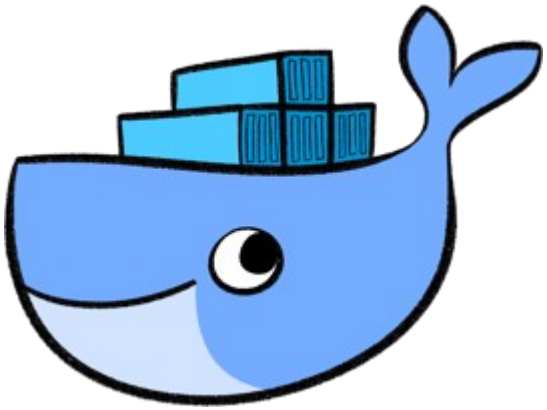
Raspberry Pi i Docker u IoT okruženjima
RaspiTajSe! 2018



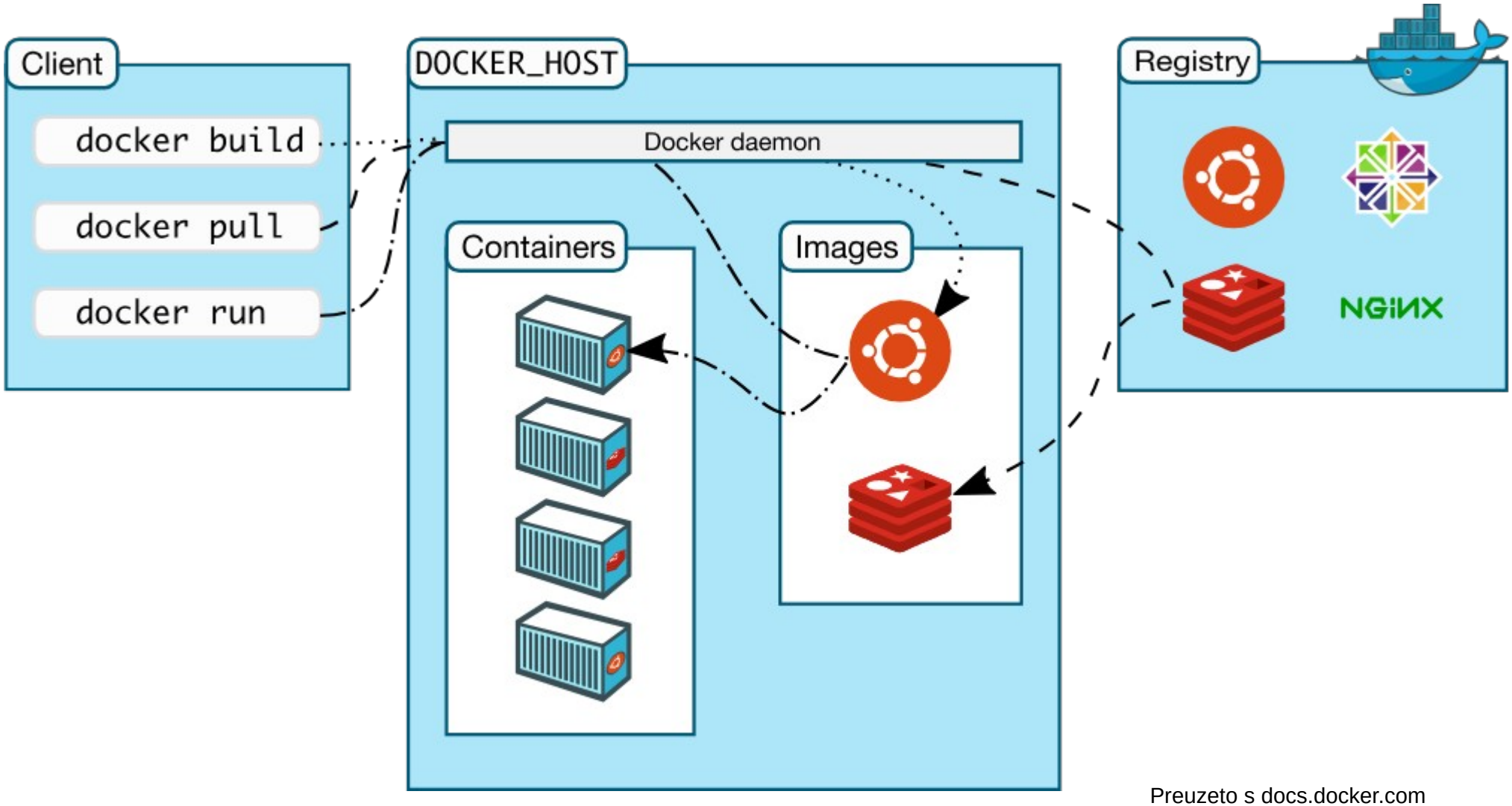
Tomislav Lokotar

git clone https://gitlab.com/tloki/raspitajse.git

Što je Docker i zašto?

- Usporediv s “virtualnim strojevima”
- Hardverski “lakši”
- Potpuno druga namjena – brzi razvoj i deployment aplikacija neovisnih o sustavu
- Podržava mnoštvo ugradbenih platformi

Što je Docker i zašto?



Preuzeto s docs.docker.com

Instalacija Docker hosta na RPi

- [1] (appendix)

  curl -sSL https://get.docker.com | sh

  docker --version

  docker run arm32v7/hello-world

  sudo usermod -aG docker pi
- reboot

Pokretanje primjera

- docker run arm32v7/hello-world
- Što, kako i zašto, uskoro...

```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
d1725b59e92d: Pull complete
Digest: sha256:0add3ace90ecb4adbf7777e9aacf18357296e799f81cabc9fde470971e499788
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Docker naredbe s containerima

- docker run <owner>/<name>:<tag>
- docker container ls [-a]
- docker container kill
- docker container rename [ID or name] [name]
- docker container start
- docker container rm
- docker container prune

Još neke korisne naredbe

- docker run -it
- docker run -it -d

Još neke, ali o njima kasnije:

- docker image ls
- docker image rm

Vježba1: pokrenimo ubuntu na RPi

- googlamo "docker ubuntu arm32v7"

  Postoji i naredba:
- docker search ubuntu

  Pokrećemo s:
- docker run -it arm32v7/ubuntu (interaktivno)

  Pokušajmo instalirati vim:
- apt update && apt install vim

Zadatak 1

- Izađite iz “kontenjera” s exit
- Uvjerite se da je ugašen naredbom:
- docker image ls

- Ponovo pokrenite kontenjer u interaktivnom modu (docker run -it)
- Pokrenut (ponovno) vim – ne radi – svojstvo zaboravljanja – it’s a feature, not a bug!

Docker repozitorij

- Mnoštvo specijaliziranih image-ova
- Lako pretraživanje, besplatno objavljivanje vlastitih image-ova
- Velika zajednica
- docs.docker.com

Vježba 2: Case study – python3.7

- Googlamo “docker python arm32v7”
- docker run ?
- docker pull vs docker run?
- uvjerimo se da je kontejner pokrenut

Vježba 2: Case study – python3.7

- Zašto bi to radili?
- Rpi ima koju verziju pythona?
- Instaliravanje novog pythona – apsolutan horor!
- Odgovara li nam uopće novi python?

- Pokrenimo ga u interaktivnom modu
- Dodatak [3] (ne preporuča se u praksi)

Case study: OpenCV on RPi

- Dockerom je moguće i emulirati druge platforme (npr arm32v7) na teoretski bilo kojem računalu (često amd64 arhitekture)
- Ključne riječi qemu, cross-build
- Velika primjena u industriji

Dockerfile: majka svakog image-a

- Dokument koji služi kao uputa za generiranje vlastitog Docker image-a
- Do sad smo koristili tuđe image-ove
- Docker image-ovi imaju takozvanu slojevitu arhitekturu
- Svega nekoliko naredbi: FROM, RUN, ADD, WORKDIR, CMD…
- Neke su gotovo istoznačne – CMD/ENTRYPOINT, ADD/COPY
- Neke I nemaju ulogu (ne rade ništa) EXPOSE, MAINTAINER

Vježba: moja docker .py aplikacija

- [5]

```
#!/usr/bin/env python3
import time

i = 1
while True:
    print(i)
    i = i + 1
    time.sleep(0.5)
```

- Pokrenimo je za probu s python3 brojilo.py
- Napravimo Docker image:
- U novom folderu: aplikacija, Dockerfile
- [4]

Zadatak 2: dockerizirati skriptu

- Skinuti aplikaciju za paljenje i gašenje LED-ica
- [6]
- git clone https://gitlab.com/tloki/rpi3-ledice.git

- Nakon pokretanja, dockerizirati vlastitu aplikaciju koja naizmjence gasi i pali crvenu i zelenu LEDicu.
- Podesite da se pokreće i pri pokretanju sustava (pogledati link pod [7])
- Docker run -v i -p

Zadatak 3: dockerizirati flask app

- Na linku [8] skinut projekt
- Pokrenut final_exercise
- Implementirati TODO dijelove u app.py

- Nakon što stvar proradi dockerizirati app

- IoT strana priče: mnoštvo uređaja, jedan image, automatsko pullanje promjena