Docker Containers

Marko Ambrož,
Žiga Hudolin
Ministrstvo za javno upravo
DIREKTORAT ZA INFORMATIKO

“Vendor lock in” with DOCKER ???

DOCKER CLIENT	DOCKER HUB
DOCKER ENGINE	DOCKER REGISTRY
DOCKER SWARM	DOCKER CLOUD
DOCKER COMPOSE	DOCKER UCP Universal Control Plane
DOCKER MACHINE	DOCKER TOOLBOX

http://blog.dennybritz.com/2015/10/01/a-brief-guide-to-the-docker-ecosystem/

Supported operating systems

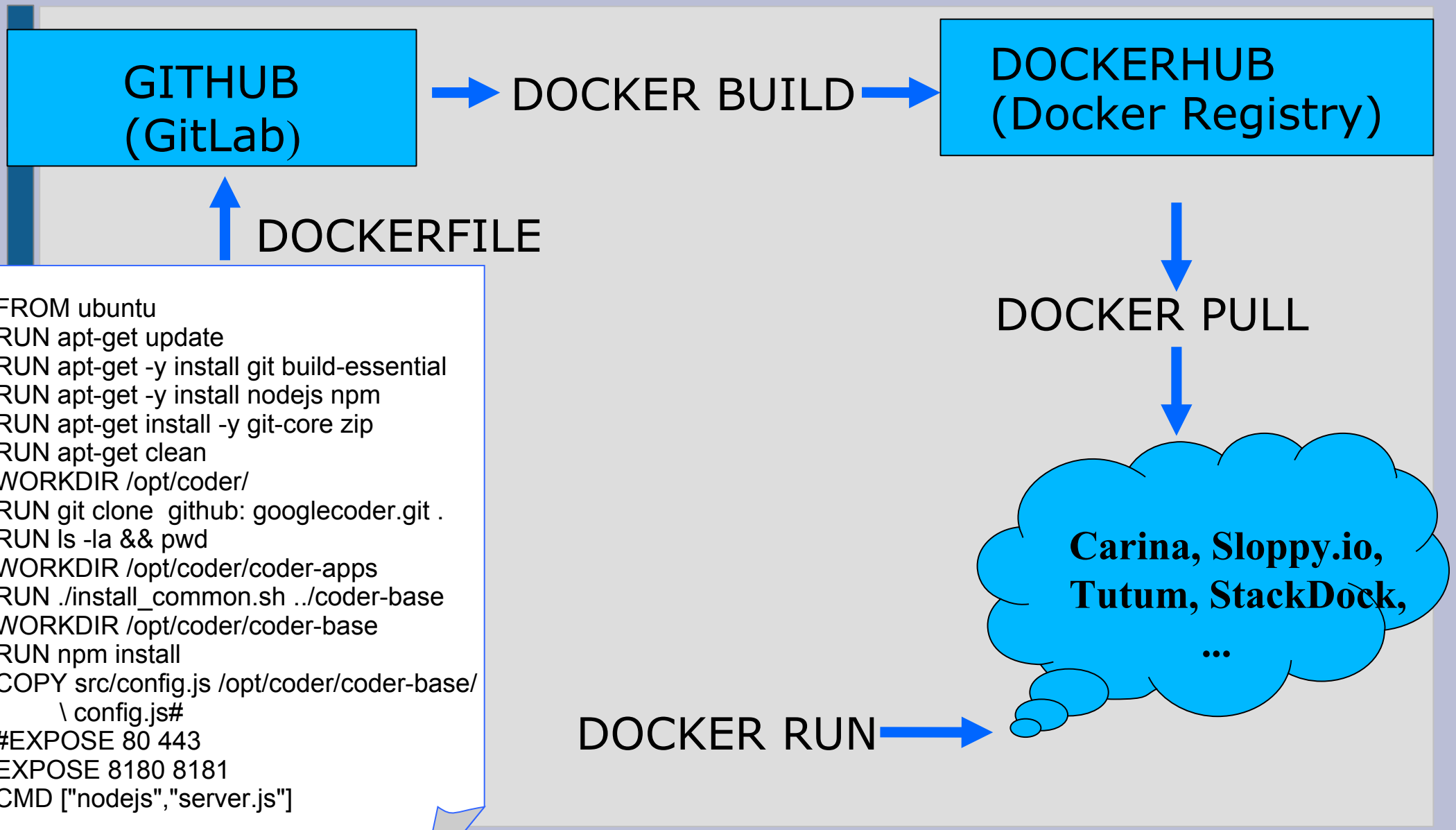
General purpose operating systems:

Amazon EC2
Arch Linux
Azure Microsoft
CentOS
CRUX Linux
Debian
Fedora
FrugalWare
Gentoo
Google Cloud
Joyent
Mac OS X
Oracle Linux
Rackspace Cloud
Red Hat Enterprise Linux
SoftLayer IBM
openSUSE and SUSE Linux
Ubuntu
Windows 20+

Container optimized operating systems:

RedHat Project Atomic
Boot2Docker
CoreOS
OSv
VMWare PhotonOS
RancherOS
Ubuntu Core 7+

Dockerization of applications



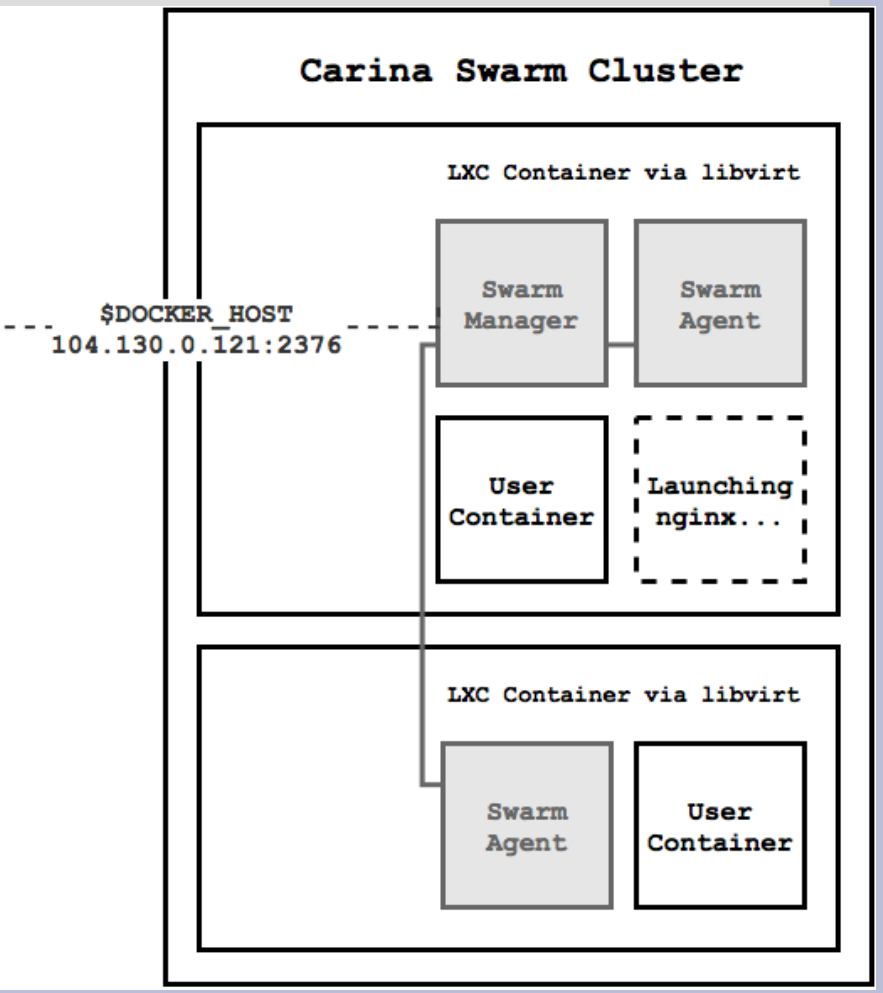

```
FROM ubuntu
RUN apt-get update
RUN apt-get -y install git build-essential
RUN apt-get -y install nodejs npm
RUN apt-get install -y git-core zip
RUN apt-get clean
WORKDIR /opt/coder/
RUN git clone  github: googlecoder.git .
RUN ls -la && pwd
WORKDIR /opt/coder/coder-apps
RUN ./install_common.sh ../coder-base
WORKDIR /opt/coder/coder-base
RUN npm install
COPY src/config.js /opt/coder/coder-base/
        \ config.js#
#EXPOSE 80 443
EXPOSE 8180 8181
CMD ["nodejs","server.js"]
```

Are Docker Containers cloud neutral?

IaaS (Infrastructure as a Service)
e.g.: DigitalOcean droplets: Ubuntu, Fedora, Debian, CoreOS, Centos

CaaS (Container as a Service)
e.g.: RackSpace: CARINA

```
$ eval "$( carina env mycluster )"
$ echo $DOCKER_HOST
tcp://104.130.0.121:2376
$ docker run nginx
```



If the demo fails you can swim with:

GoogleCoder in a Container at DigitalOcean:

https://188.166.34.228:7443/

Password: SInog32016

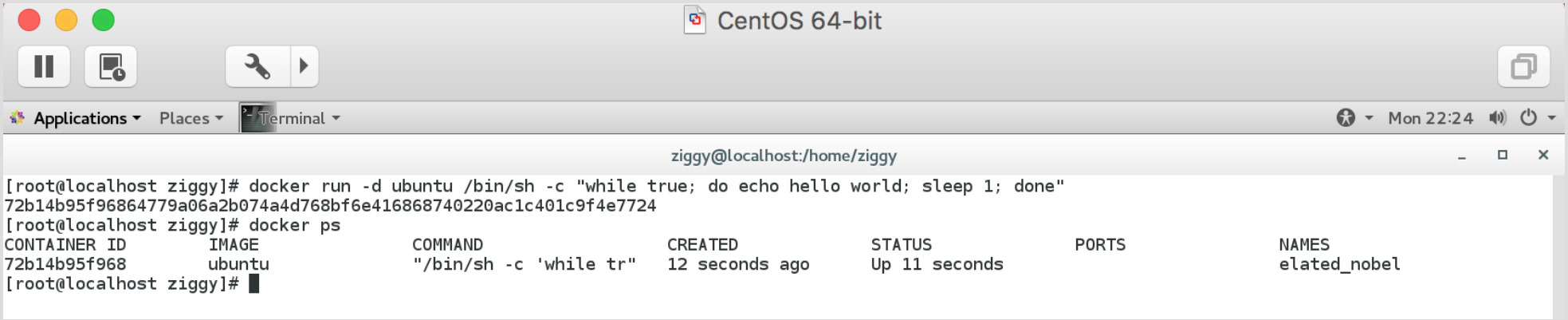
Demonstration

Example 1

- Target: Create a container that runs as a deamon and a command that print Hello World on screen every second
- In this example we will use
  - docker run; runs the container
  - -d; flag runs the container in the background
  - ubuntu; is the image you would like to run
  - docker logs; looks inside the container
- Finally we specify a command to run:
  - /bin/sh -c "while true; do echo hello world; sleep 1; done”

Example 1: docker run

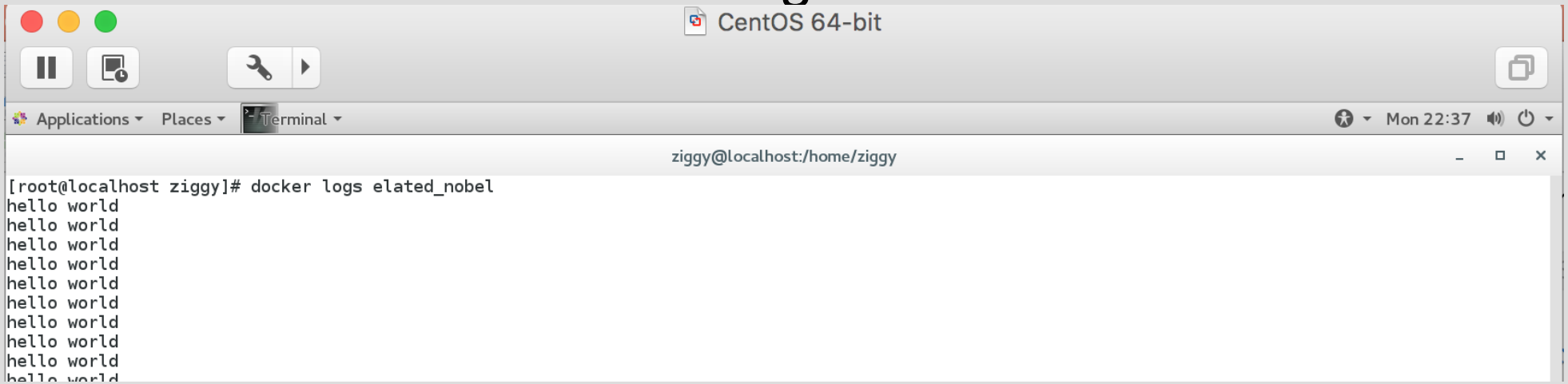
- docker run -d ubuntu /bin/sh -c "while true; do echo hello world; sleep 1; done”



- This long string “72b14b95f96864779a06a2b...” is called a container ID
- docker ps command queries the Docker daemon for information about all the containers.

Example 1: docker logs

- To see what is going on inside the container we will use the docker logs command



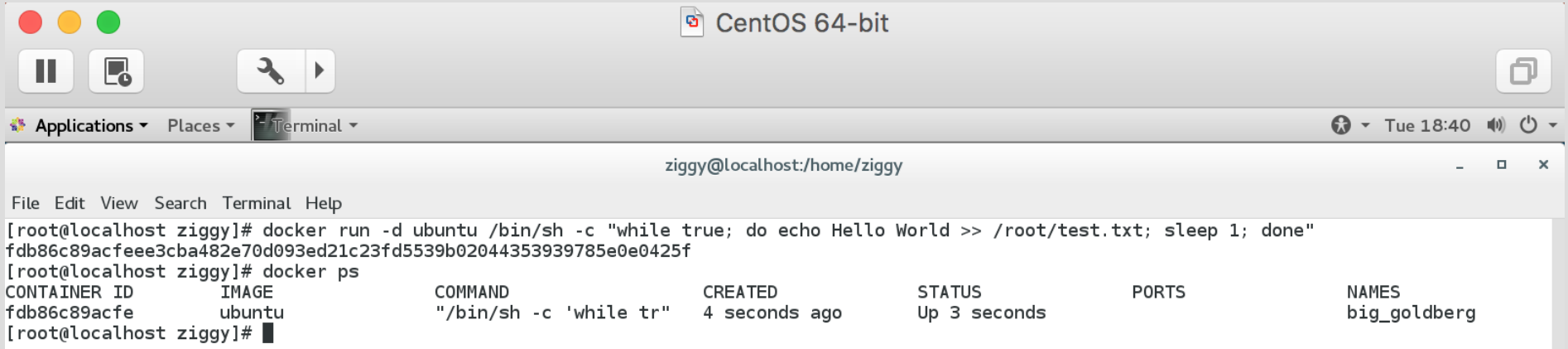
- docker stop command tells Docker to politely stop the running container and returns the name of the container it stopped “elated_nobel”

Example 2

- Target: Create a container that runs as a deamon and a command that print Hello World every second and write it in a specific file
- In this example we will use next commands
  - docker run; runs the container
  - -d; flag runs the container in the background
  - ubuntu; is the image you would like to run
  - docker exec; runs a new command in a running container
- Finally we specify a command to run an write result in a file

Example 2: docker run

- docker run -d ubuntu /bin/sh -c "while true; do echo Hello World >> /root/test.txt; sleep 1; done"



- This long string “fdb86c89acfeee3cba482e70…” is called a containerID
- docker ps command queries the Docker daemon for information about all the containers.

Example 2: docker exec

- With command docker exec we run a new command in a running container

CentOS 64-bit

Applications Places Terminal Tue 18:56

ziggy@localhost:/home/ziggy

File Edit View Search Terminal Help

```
[root@localhost ziggy]# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS               NAMES
fdb86c89acfe        ubuntu              "/bin/sh -c 'while tr"   15 minutes ago      Up 15 minutes                           big_goldberg
[root@localhost ziggy]# docker exec -it fdb86c89acfe bash
root@fdb86c89acfe:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@fdb86c89acfe:/# top
TERM environment variable not set.
root@fdb86c89acfe:/#
```

- We set TERM environment variable in running container

CentOS 64-bit

Applications Places Terminal Tue 18:58

ziggy@localhost:/home/ziggy

File Edit View Search Terminal Help

```
[root@localhost ziggy]# docker exec -it fdb86c89acfe bash
root@fdb86c89acfe:/# export TERM=linux
root@fdb86c89acfe:/# top
```

Example 3: docker commit

- Target: update a container created from an image with specific file and commit the results to an image
- In this example we will use next command
  - docker commit; create a new image from a container's changes

CentOS 64-bit

Applications Places Terminal Mon 23:59

ziggy@localhost:/home/ziggy

```
[root@localhost ziggy]# docker run -d ubuntu /bin/touch /root/krneki.txt
f5de9e15f2a7602c6fc054a95842ba7cbda612eb9b4b65454d8b26433bef7804
[root@localhost ziggy]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS                      PORTS               NAMES
f5de9e15f2a7        ubuntu              "/bin/touch /root/krn"   9 seconds ago       Exited (0) 8 seconds ago                        jolly_curie
[root@localhost ziggy]# docker commit f5de9e15f2a7 ubuntu
sha256:e1f0e7ea00b8e5576c7ec443953eff2a0985be0dff016e30952ecde8cf14a830
[root@localhost ziggy]# docker run ubuntu /bin/ls /root/
krneki.txt
[root@localhost ziggy]# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED              STATUS                          PORTS               NAMES
467e061190ec        ubuntu              "/bin/ls /root/"         16 seconds ago       Exited (0) 15 seconds ago                           amazing_dubin
sky
f5de9e15f2a7        2fa927b5cdd3        "/bin/touch /root/krn"   About a minute ago   Exited (0) About a minute ago                       jolly_curie
[root@localhost ziggy]#
```

Example 4

- Target: Install aplication Google Coder
- In this example we will use next commands
  - docker pull; pull an image or a repository from the registry
  - docker images; list images
- Finally we check our application in browser

Example 4: docker pull

- We will use image from ambrom/ambro-gc repository on dockerhub

CentOS 64-bit

Applications Places Terminal Tue 00:43

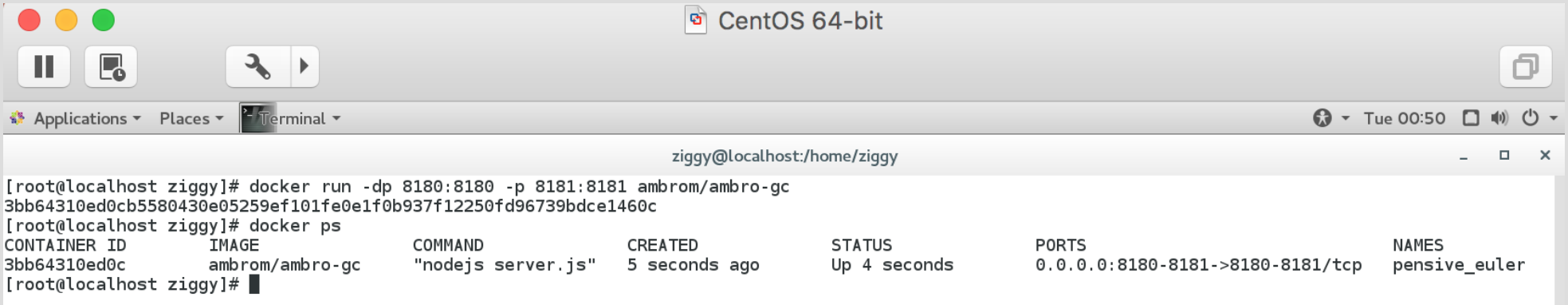
ziggy@localhost:/home/ziggy

```
[root@localhost ziggy]# docker pull ambrom/ambro-gc
Using default tag: latest
latest: Pulling from ambrom/ambro-gc

72b39c1d4615: Pull complete
46a2d5ede4a6: Pull complete
d7caf6e91ad4: Pull complete
c7ac9f284354: Pull complete
a3ed95caeb02: Pull complete
34c616d65681: Pull complete
b5817a0be70e: Pull complete
e86cc6a6aba4: Pull complete
96df172b271e: Pull complete
a3c9f24cb71f: Pull complete
9df399d952c1: Pull complete
e00e3772ef94: Pull complete
8cf1244f1777: Pull complete
385994367178: Pull complete
Digest: sha256:ca4d6c530ee5cdc9feb0f43b5cf2dab917ed8cd9f0bc73d767909f9a358322b0
Status: Downloaded newer image for ambrom/ambro-gc:latest
[root@localhost ziggy]#
```

Example 4: docker run

- With command docker run we will run our application, but we have to specify ports “8180 and 8181”



- We will open the browser and check if application is alive and responsible

Primer 4: test aplikacije

- https://localhost:8181