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

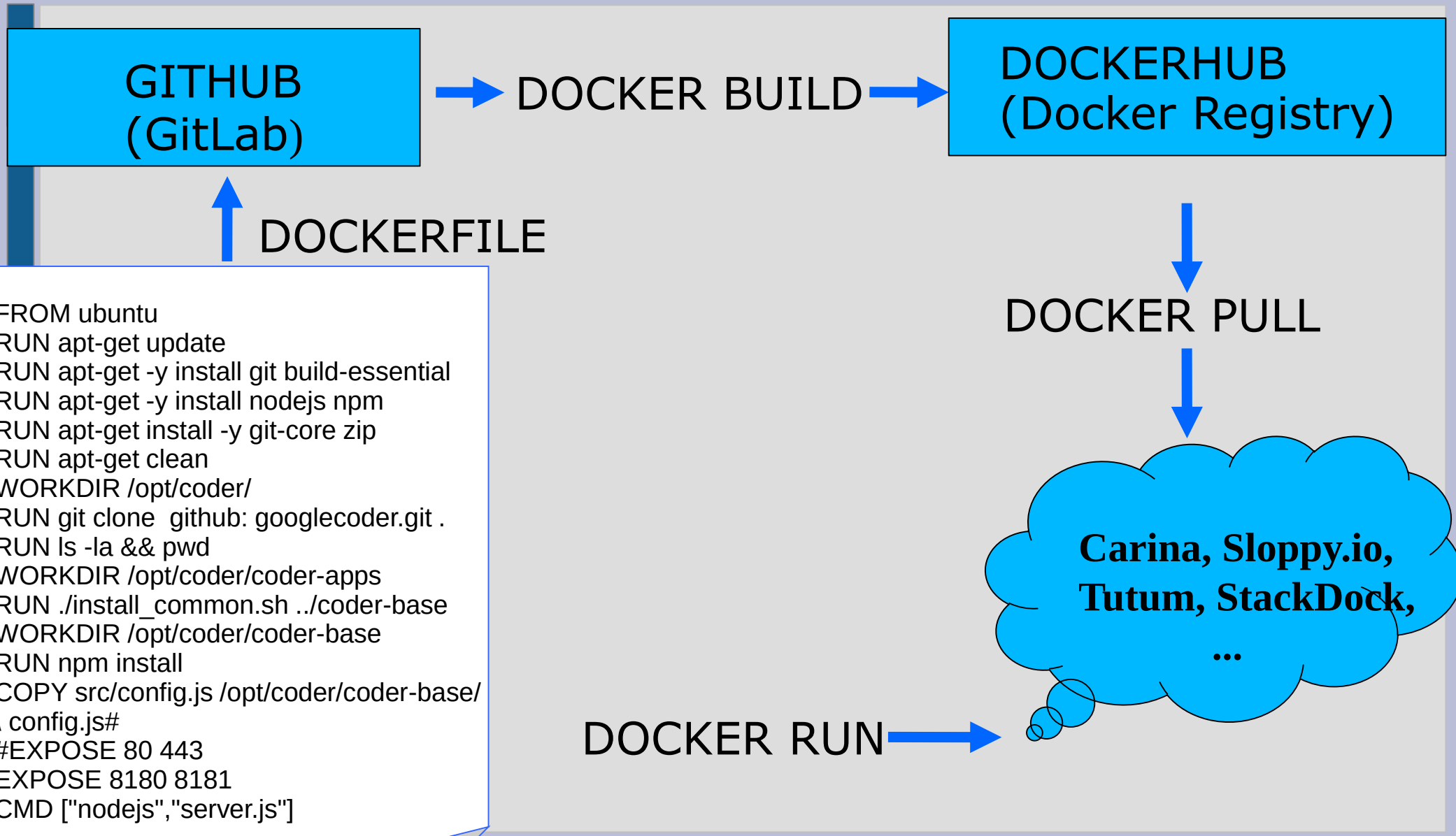
General purpose operating systems:

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+

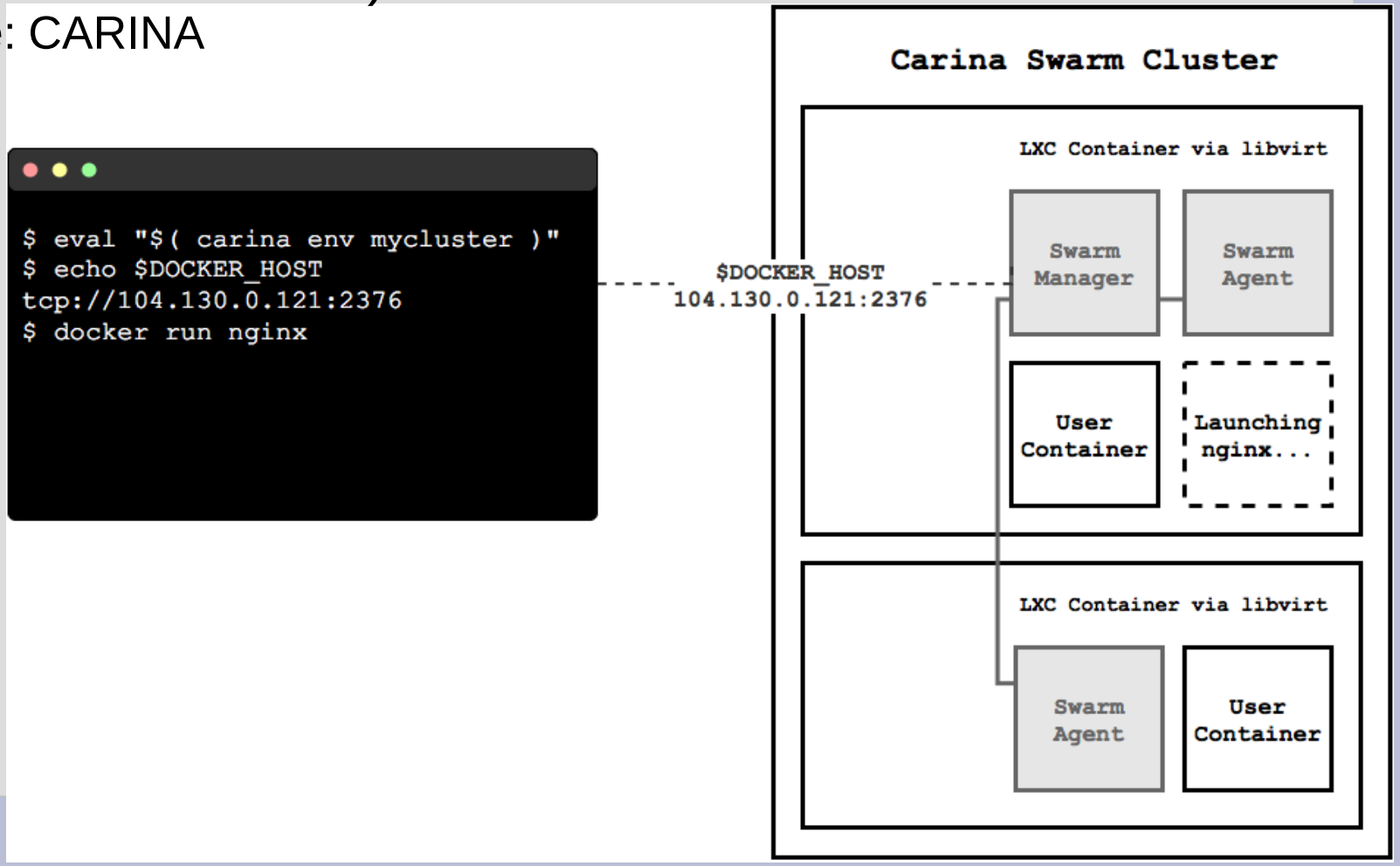
Dockerization of applications



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



If the demo fails you can swim with:

GoogleCoder in a Container at DigitalOcean:

https://188.166.34.228:7443

Password: SInog32016

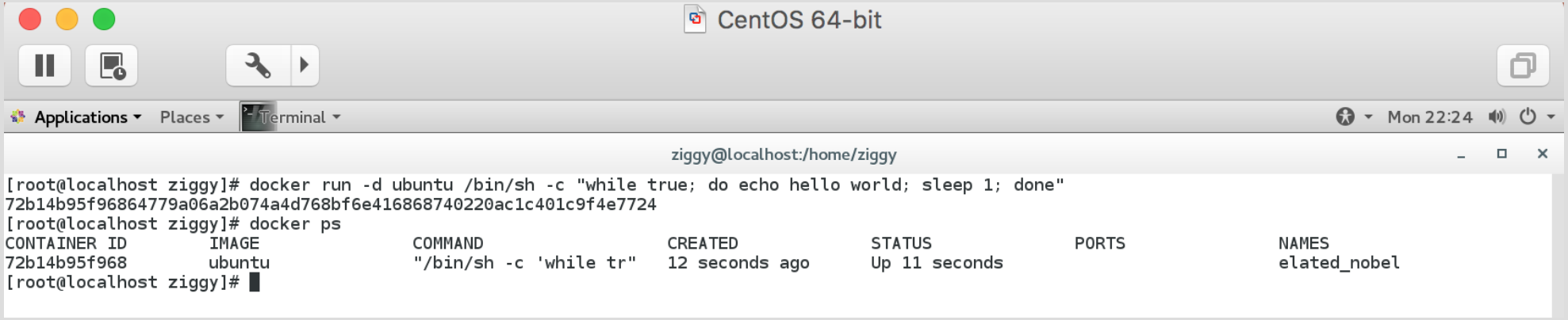
Demonstration

Example 1

- Goal: Create a container running 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

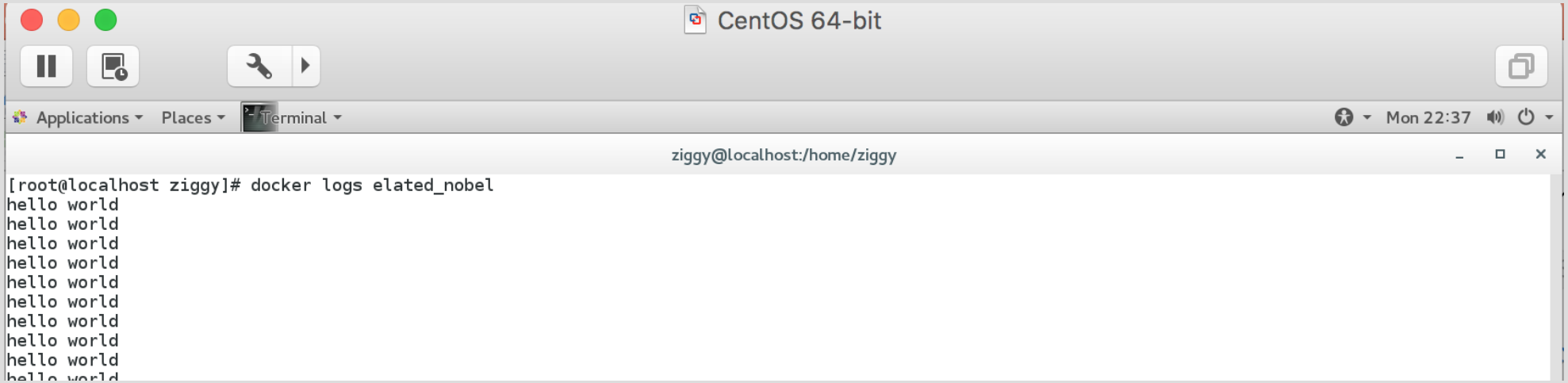
- docker run -d ubuntu /bin/sh -c "while true; do echo hello world; sleep 1; done”



- 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 use the docker logs command



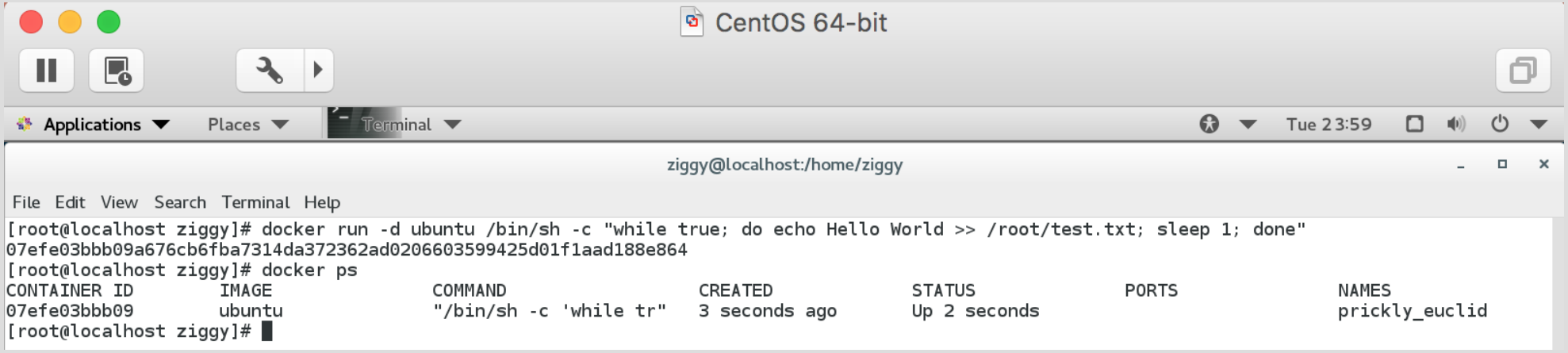
- docker stop command tells Docker to politely stop the running container and returns the name of the container it stopped in our case “elated_nobel”

Example 2

- Goal: Create a container running 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
  - 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 bash session in a running container
- Finally we specify a command to run an write result in a file

Example 2: docker run

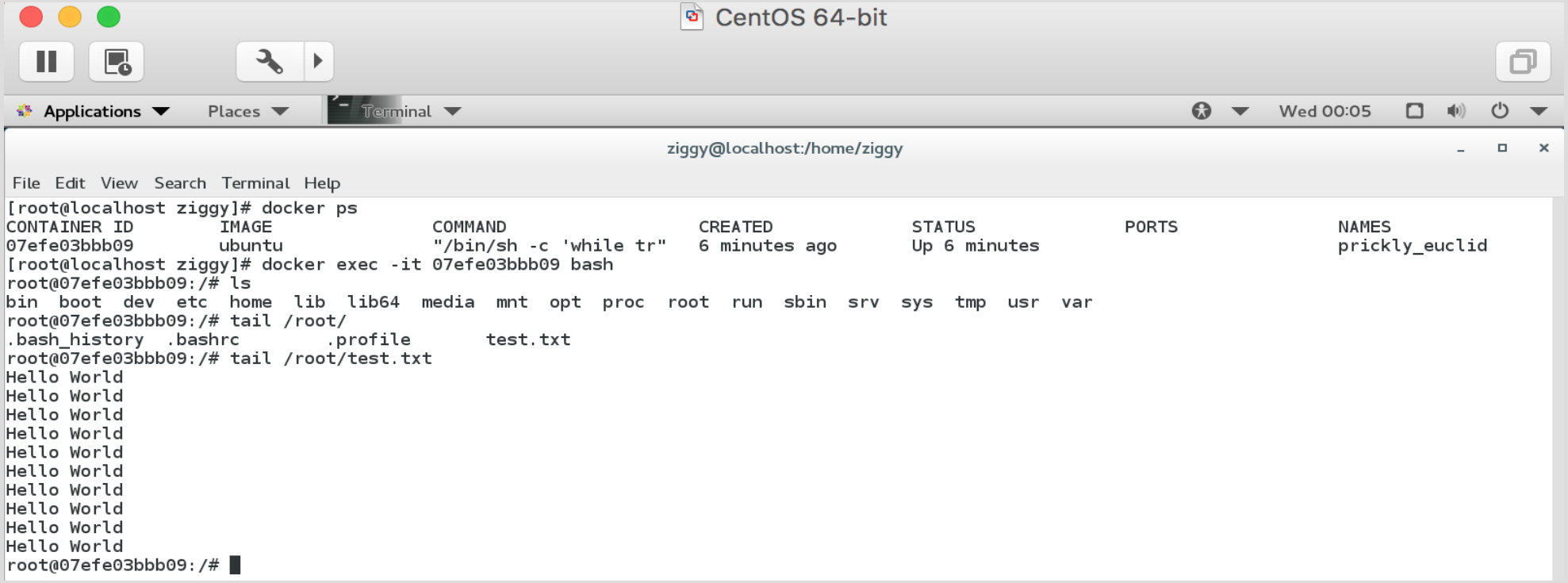
- docker run -d ubuntu /bin/sh -c "while true; do echo Hello World >> /root/test.txt; sleep 1; done”



- This long string “07efe03bbb09a676cb6fba73...” is called a container ID
- docker ps command queries the Docker daemon for information about all the containers.

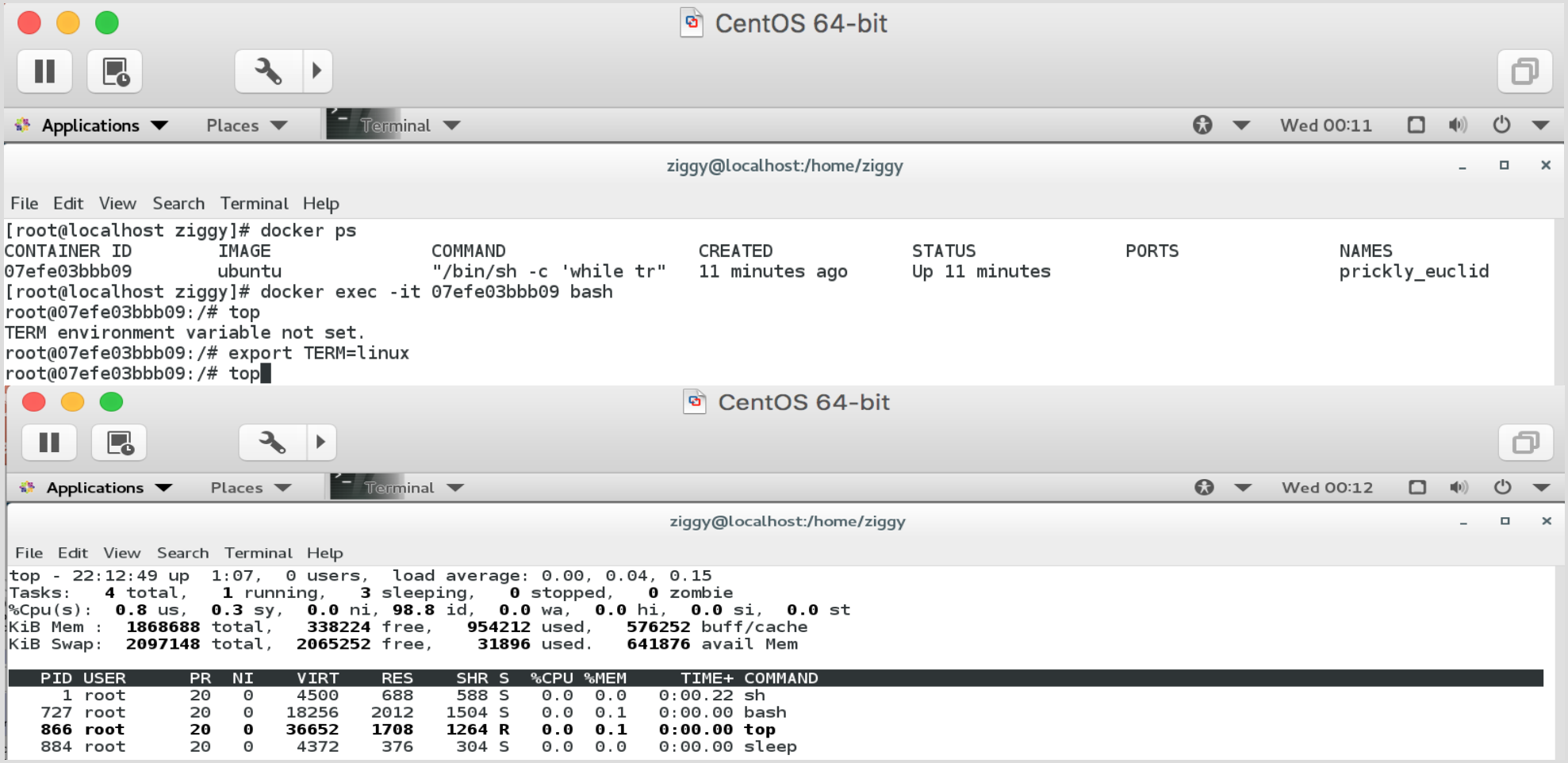
Example 2: docker exec

- With command docker exec we run a new bash session in a running container and we can check the file
  - -i -- interactive; keep STDIN open even if not attached
    -- privileged; Give extended Linux capabilities to the command
  - -t -- tty; Allocate a pseudo-TTY



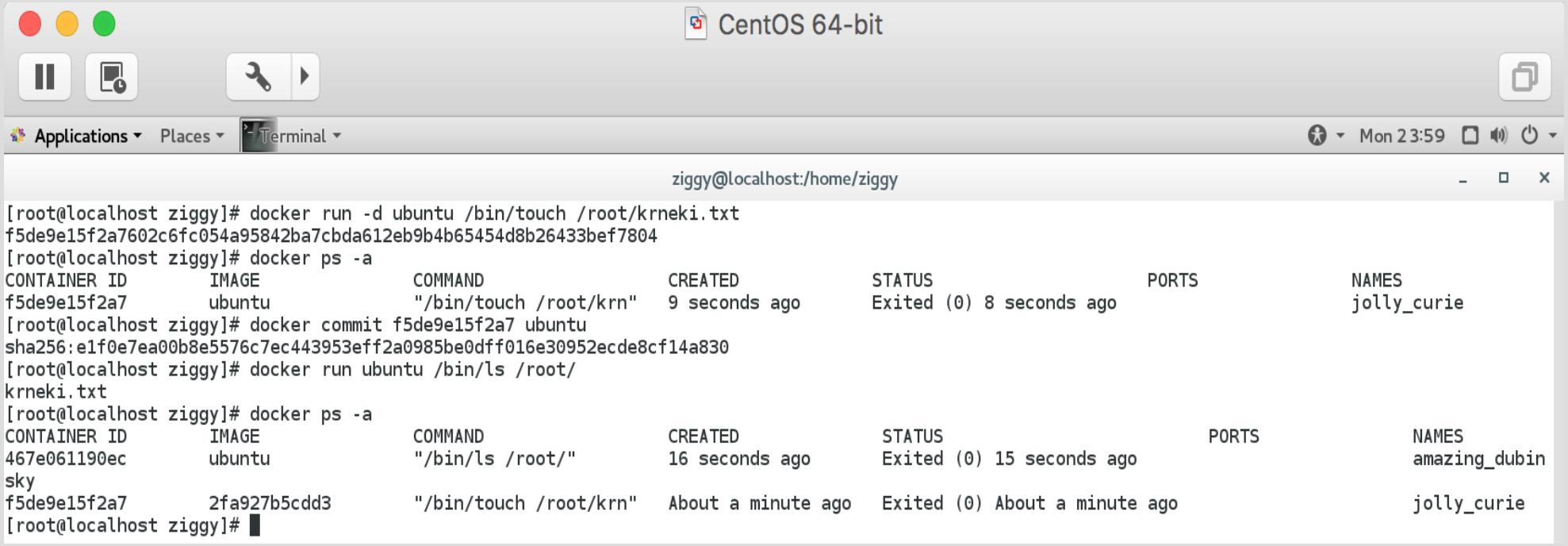
Example 2: export TERM

- We set TERM environment variable in running container



Example 3: docker commit

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

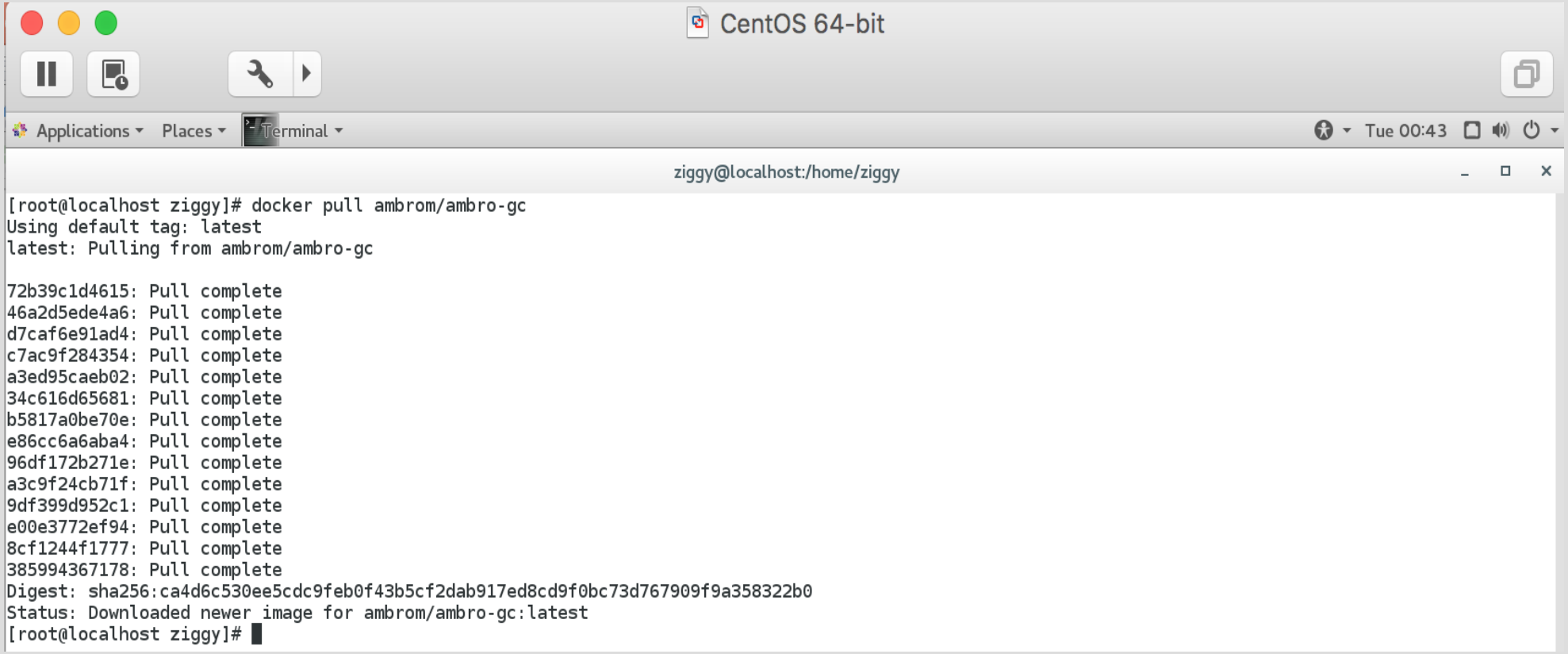


Example 4

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

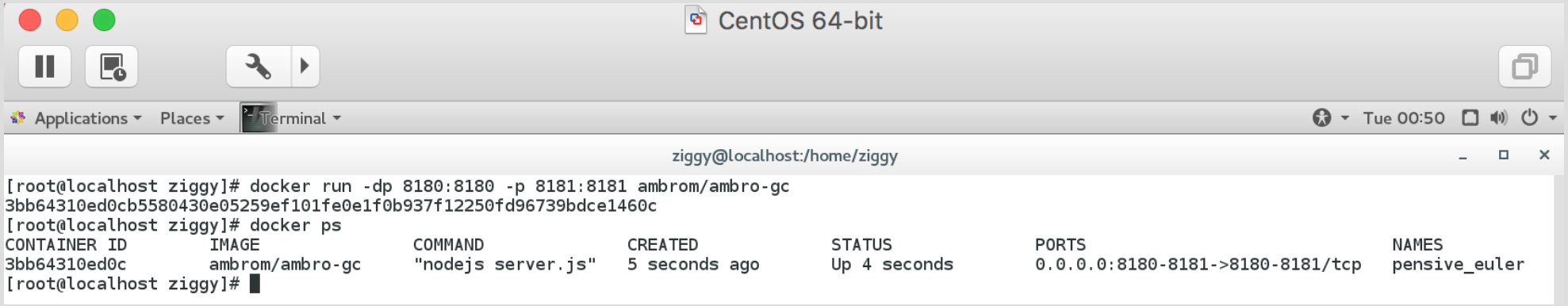
Example 4: docker pull

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



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