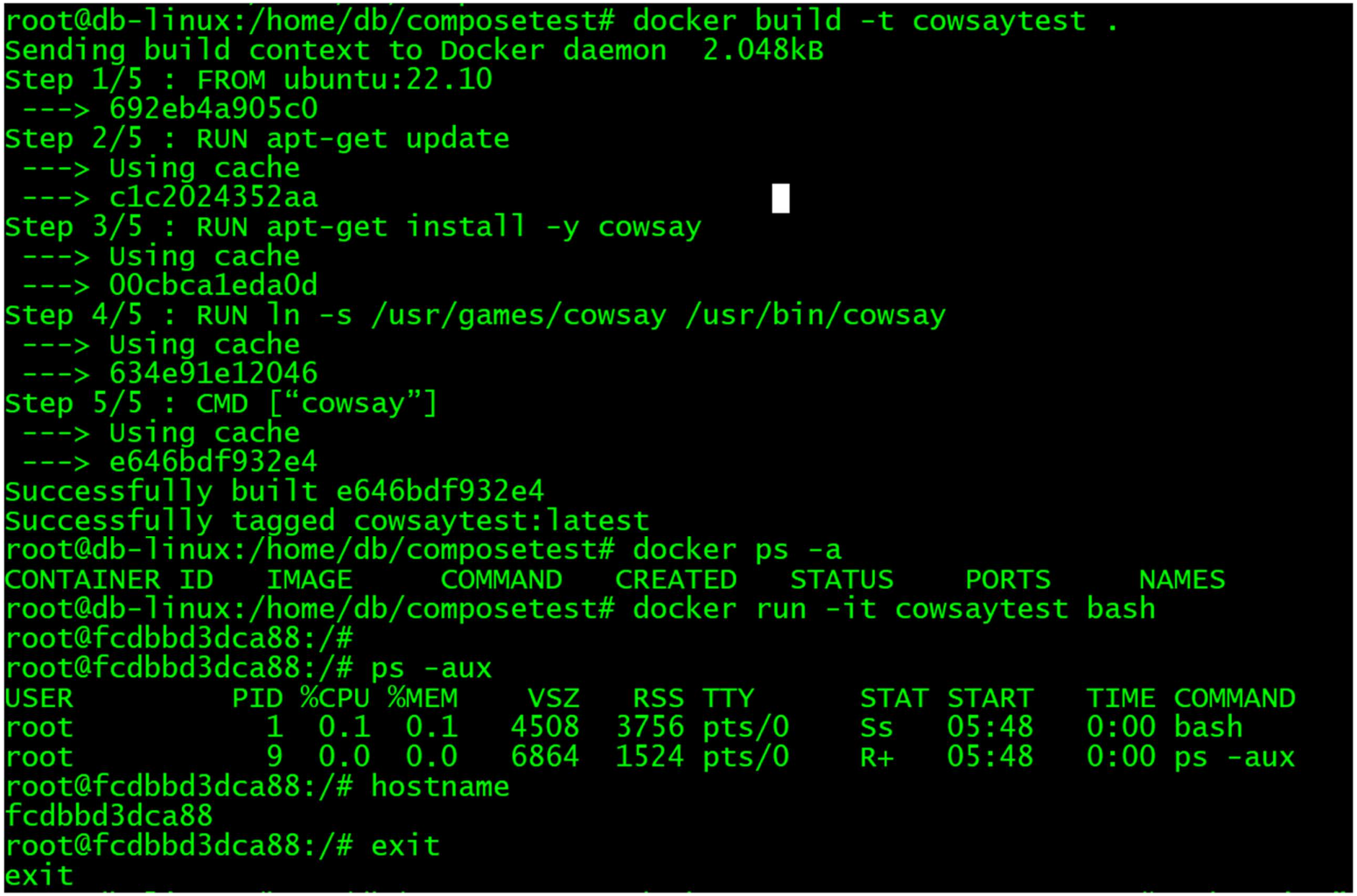
1 Вариант, как показано на лекции и на семинаре, (пример с cowsay)

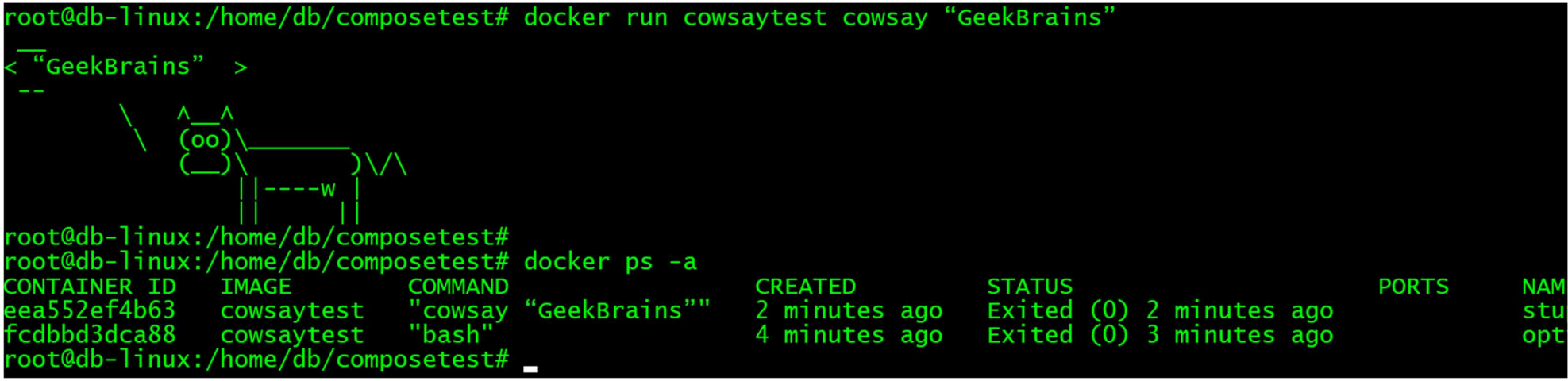
```
db@db-linux:~$ sudo su
[sudo] password for db:
root@db-linux:/home/db#
root@db-linux:/home/db#
root@db-linux:/home/db#
root@db-linux:/home/db#
root@db-linux:/home/db# cd composetest/
root@db-linux:/home/db/composetest# ls
Dockerfile
root@db-linux:/home/db/composetest# cat Dockerfile
FROM ubuntu:22.10
RUN apt-get update
RUN apt-get install -y cowsay
RUN ln -s /usr/games/cowsay /usr/bin/cowsay
CMD [“cowsay”]

root@db-linux:/home/db/composetest# docker rm $(docker ps -a -q) --force
8ea27062544c
51f817762982
8c6bbe22c0bf
7636434d53d7
a7c892c91339
root@db-linux:/home/db/composetest# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@db-linux:/home/db/composetest#
```

Создание образа


root@db-linux:/home/db/composetest# docker build -t cowsaytest .
Sending build context to Docker daemon 2.048kB
Step 1/5 : FROM ubuntu:22.10
---> 692eb4a905c0
Step 2/5 : RUN apt-get update
---> Using cache
---> c1c2024352aa
Step 3/5 : RUN apt-get install -y cowsay
---> Using cache
---> 00cbca1eda0d
Step 4/5 : RUN ln -s /usr/games/cowsay /usr/bin/cowsay
---> Using cache
---> 634e91e12046
Step 5/5 : CMD ["cowsay"]
---> Using cache
---> e646bdf932e4
Successfully built e646bdf932e4
Successfully tagged cowsaytest:latest
root@db-linux:/home/db/composetest# docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
root@db-linux:/home/db/composetest# docker run -it cowsaytest bash
root@fcdbbd3dca88:/#
root@fcdbbd3dca88:/# ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.1 0.1 4508 3756 pts/0 Ss 05:48 0:00 bash
root 9 0.0 0.0 6864 1524 pts/0 R+ 05:48 0:00 ps -aux
root@fcdbbd3dca88:/# hostname
fcdbbd3dca88
root@fcdbbd3dca88:/# exit
exit

Запуск



2 Вариант:

Формирование образа при помощи исходника на C++

В папке DockerHelloWorldProject0 находится Dockerfile и рабочая директория Helloworld,

в ней находится исходник helloword.cpp с программой, выводящей в консоль “Helloworld 0 !”

В Dockerfile рабочая директория указана в команде WORKDIR /Helloworld/

```
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$ ls -l
total 12
-rw-rw-r-- 1 db db  147 июл 14 09:36 Dockerfile
-rw-rw-r-- 1 db db  130 июл 14 09:36 Dockerfile~
drwxrwxr-x 2 db db 4096 июл 14 09:36 HelloWorld
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$ cat Dockerfile
FROM amytabb/docker_ubuntu16_essentials
COPY HelloWorld /HelloWorld
WORKDIR /HelloWorld/
RUN g++ -o HelloWorld helloworld.cpp
CMD ["./HelloWorld"]
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$ ls -l HelloWorld/
total 8
-rw-rw-r-- 1 db db 110 июл 14 09:36 helloworld.cpp
-rw-rw-r-- 1 db db 140 июл 14 09:36 helloworld.cpp~
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$ cat HelloWorld/helloworld.cpp
#include <iostream>
using namespace std;

int main()
{
  cout << "Hello world 0!" << endl;
  return 0;
}
```

Создаем образ:

```
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$ sudo docker build -t hello0 .
[sudo] password for db:
Sending build context to Docker daemon  5.632kB
Step 1/5 : FROM amytabb/docker_ubuntu16_essentials
latest: Pulling from amytabb/docker_ubuntu16_essentials
8ee29e426c26: Pull complete
6e83b260b73b: Pull complete
e26b65fd1143: Pull complete
40dca07f8222: Pull complete
b420ae9e10b3: Pull complete
dfaf13193fe3: Pull complete
Digest: sha256:aa1aee49d49b6d641a81afeedb252befe55fbbce6a6badc50e61cf649dc9e8e6
Status: Downloaded newer image for amytabb/docker_ubuntu16_essentials:latest
 ---> 13f6d277d91b
Step 2/5 : COPY HelloWorld /HelloWorld
 ---> e496df5f8a72
Step 3/5 : WORKDIR /HelloWorld/
 ---> Running in 0d164fb10521
Removing intermediate container 0d164fb10521
 ---> 71205e292548
Step 4/5 : RUN g++ -o HelloWorld helloworld.cpp
 ---> Running in bafdb6eb2f7e
Removing intermediate container bafdb6eb2f7e
 ---> 42056a141dce
Step 5/5 : CMD ["./HelloWorld"]
 ---> Running in b71cb339431a
Removing intermediate container b71cb339431a
 ---> 83de42b2bcdc
Successfully built 83de42b2bcdc
Successfully tagged hello0:latest
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$
```

Запуск:

```
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$ sudo docker run -it hello0
Hello world 0!
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$
```

```
db@db-linux:~/docker-tutorial-files/DockerHelloWorldProject0$ sudo docker ps -a
[sudo] password for db:
CONTAINER ID   IMAGE              COMMAND                  CREATED              STATUS
171ceb03f8c6   hello0             "./HelloWorld"           27 minutes ago       Exited (0) 27 minutes ago
```

3 Вариант, на чистом C

```
root@db-linux:/DockerWorld# ls -l
total 8
-rw-r--r-- 1 root root 121 июл 14 11:18 Dockerfile
-rw-r--r-- 1 root root 110 июл 14 09:20 dockerworld.c
root@db-linux:/DockerWorld# cat Dockerfile
FROM gcc:latest

COPY . /DockerWorld

WORKDIR /DockerWorld/

RUN gcc -o DockerWorld dockerworld.c

CMD ["./DockerWorld"]
root@db-linux:/DockerWorld#
root@db-linux:/DockerWorld# cat dockerworld.c
#include <stdio.h>

// Driver code
int main()
{
    printf("Welcome to Docker World!!!\n");
    return 0;
}
root@db-linux:/DockerWorld# _
```

Создаем образ и запускаем

```
root@db-linux:/DockerWorld# docker build -t docker-world-gcc /DockerWorld/
Sending build context to Docker daemon  3.072kB
Step 1/5 : FROM gcc:latest
 ---> ce4bf977c328
Step 2/5 : COPY . /DockerWorld
 ---> 96d311f68e89
Step 3/5 : WORKDIR /DockerWorld/
 ---> Running in 21b545077dab
Removing intermediate container 21b545077dab
 ---> 2706241f957b
Step 4/5 : RUN gcc -o DockerWorld dockerworld.c
 ---> Running in 547bcda663df
Removing intermediate container 547bcda663df
 ---> 6704d4a6dcc9
Step 5/5 : CMD ["./DockerWorld"]
 ---> Running in 07fb0a839d54
Removing intermediate container 07fb0a839d54
 ---> c14b379e7b61
Successfully built c14b379e7b61
Successfully tagged docker-world-gcc:latest
root@db-linux:/DockerWorld# docker run -it docker-world-gcc
Welcome to Docker World!!!
root@db-linux:/DockerWorld#
```