



ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ &  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ

1

University of West Attica - Dr. Grigorios Koulouras

Προπτυχιακό Πρόγραμμα Σπουδών  
«Ηλεκτρολόγων και Ηλεκτρονικών Μηχανικών»

# Μάθημα «Νεφροϋπολογιστική»

Hands-on LAB08  
«Απλά παραδείγματα εφαρμογών με Docker»

Κωδικός Μαθήματος: EEE-9.2.4, EEE-9.3.7

Ακαδημαϊκό Έτος: 2023-2024

Διδάσκων:

Δρ. ΚΟΥΛΟΥΡΑΣ ΓΡΗΓΟΡΙΟΣ  
Αναπληρωτής Καθηγητής

email: [gregkoul@uniwa.gr](mailto:gregkoul@uniwa.gr)

# Περιεχόμενα

- ▶ Απλά παραδείγματα εφαρμογών με Docker
  - ▶ 1<sup>st</sup> Docker Tip – STDIN and STDOUT in Docker
  - ▶ 2<sup>nd</sup> Docker Tip – VOLUME Mapping in Docker
- ▶ Δημιουργία του πρώτου Docker

# docker

Απλά παραδείγματα εφαρμογών με Docker

# docker

## 1<sup>st</sup> Docker Tip - STDIN and STDOUT in Docker

A very simple shellapp demonstrates how you can use STDIN and STDOUT in docker.

# 1<sup>st</sup> Docker Tip

## Dockerized simple Shell Application (1/4)

- ▶ Για τις ανάγκες του μαθήματος υλοποιήθηκε ένα πολύ απλό Shell Application, το οποίο ζητάει με διαλογικό τρόπο από το χρήστη το όνομά του και στη συνέχεια τυπώνει ένα δυναμικό μήνυμα καλωσορίσματος.
- ▶ Εδώ φαίνεται το περιεχόμενο του αρχείου `app.sh`, για να καταλάβουμε την λειτουργία του.

```
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$ cat app.sh
echo -n "Welcome to UNIWA! Please enter your name:"
read name
echo " "
echo "Hello $name! You are going to learn docker!"
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$
```

- ▶ Στη συνέχεια εκτελούμε το shell application τοπικά στο Host.

```
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$ sh app.sh
Welcome to UNIWA! Please enter your name:Gregory

Hello Gregory! You are going to learn docker!
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$
```

# 1<sup>st</sup> Docker Tip

## Dockerized simple Shell Application (2/4)

- ▶ Σε επόμενη ενότητα θα δείξουμε αναλυτικά πως μπορεί να δημιουργηθεί ένα docker image αυτού του πολύ απλού Shell Application.
- ▶ Με την εντολή **docker run**, δημιουργούμε ένα docker container από ένα docker image που έχει ανέβει για τις ανάγκες του μαθήματος στην Public Registry του DockerHub.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker run gregkoul/shellapp-stdin:1.1
Unable to find image 'gregkoul/shellapp-stdin:1.1' locally
1.1: Pulling from gregkoul/shellapp-stdin
188c0c94c7c5: Pull complete
d956180d7f4a: Pull complete
Digest: sha256:7b2b5168ed52cc5687f2975b40120f3a4d9a1d1e0f1e1e76c7edff3412744ffd
Status: Downloaded newer image for gregkoul/shellapp-stdin:1.1
Welcome to UNIWA! Please enter your name:

Hello ! You are going to learn docker!
user@snf-15678:~/Dockerized-shellapp-stdin$
```

# 1<sup>st</sup> Docker Tip

## Dockerized simple Shell Application (3/4)

- ▶ Με την εντολή **docker image ls**, βλέπουμε όλα τα docker images.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
gregkoul/shellapp-stdin	1.1	f442dbb41154	8 minutes ago	5.57MB
alpine	latest	d6e46aa2470d	5 weeks ago	5.57MB
docker/whalesay	latest	6b362a9f73eb	5 years ago	247MB

```
user@snf-15678:~/Dockerized-shellapp-stdin$
```

- ▶ Με την εντολή **docker container ls**, βλέπουμε όλα τα docker containers.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9a90005542f4	gregkoul/shellapp-stdin:1.1	"sh app.sh"	5 minutes ago	Exited (0)	5 minutes ago	shell

```
user@snf-15678:~/Dockerized-shellapp-stdin$
```

# 1<sup>st</sup> Docker Tip

## Dockerized simple Shell Application (4/4)

- ▶ Παρατηρούμε ότι δεν μπορούμε να χρησιμοποιήσουμε το πληκτρολόγιο για να βάλουμε είσοδο στο shellapp όπως όταν το τρέχαμε τοπικά. Γιατί συμβαίνει αυτό;
  - ▶ Αυτό συμβαίνει γιατί το docker container από default δεν ακούει σε κάποιο standard input (stdin). Ακόμα και να έχουμε κάνει attach στην κονσόλα του, δεν ακούει καμία είσοδο από το πληκτρολόγιο. Ο λόγος είναι ότι τρέχει σε non-interactive mode.
- ▶ Αν θέλουμε να αντιστοιχίσουμε το stdin και το stdout του host στο docker container, θα πρέπει να τρέξουμε την ίδια εντολή με τις παραμέτρους:
  - ▶ -i : Keep STDIN open even if not attached
  - ▶ -t : Allocate a pseudo-tty

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker run -it gregkoul/shellapp-stdin:1.1
Welcome to UNIWA! Please enter your name:Gregory
Hello Gregory! You are going to learn docker!
user@snf-15678:~/Dockerized-shellapp-stdin$
```



# docker

## 2<sup>nd</sup> Docker Tip – VOLUME Mapping in Docker

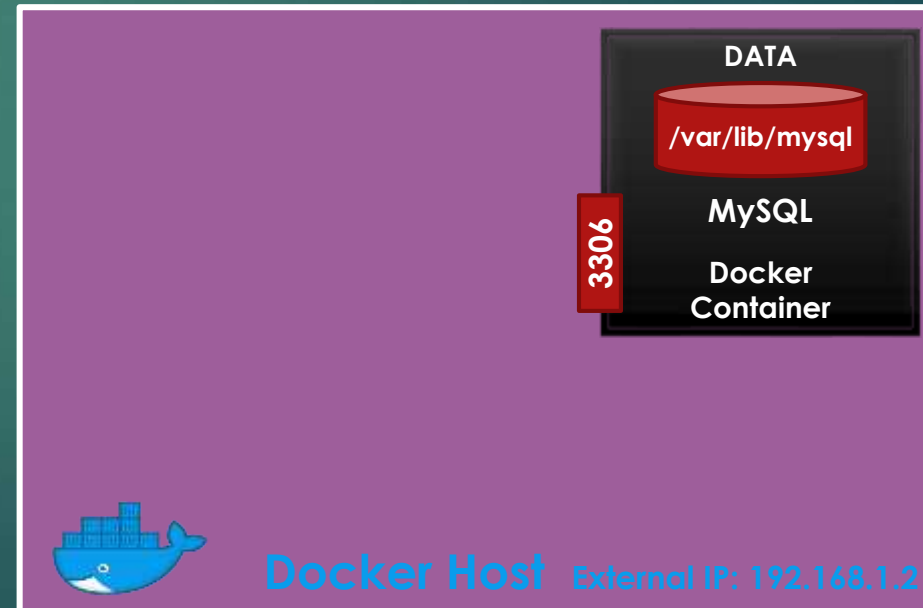
A simple example demonstrates the volume mapping in docker.

# 2<sup>nd</sup> Docker Tip

## Volume mapping in docker host (1/6)

- Υποθετικά μια εφαρμογή χρειάζεται μια Database MySQL. Η παρακάτω εντολή ξεκινάει ένα instance MySQL Server.

```
user@snf-15678:~$ docker run -d mysql
045bc2f8a7316c5d73d1332c24fa7a82f677900a52397e343da2378231121480
user@snf-15678:~$
```

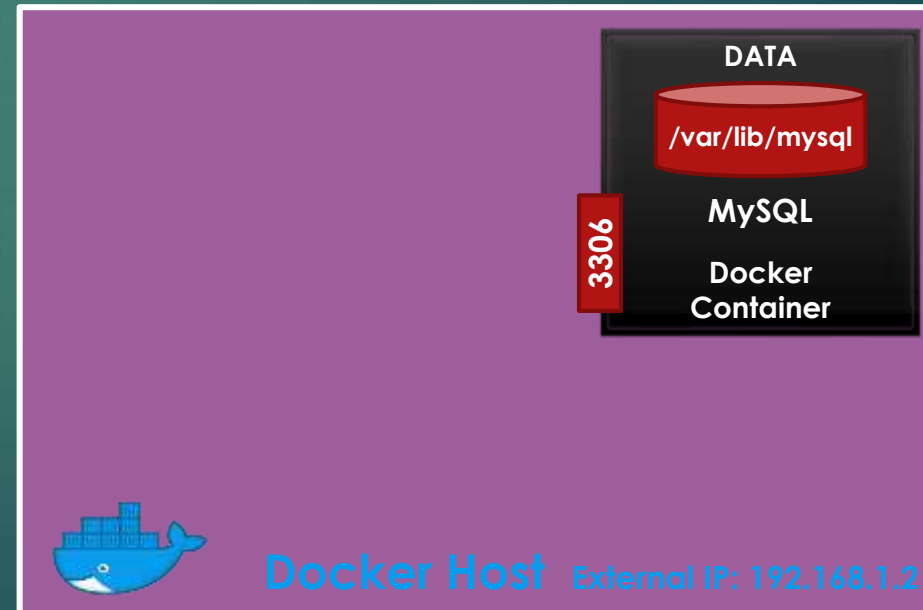


# 2<sup>nd</sup> Docker Tip

## Volume mapping in docker host (2/6)

- Σταματώντας το instance του MySQL Server, τα DATA είναι ακόμα μέσα στο Docker Container.

```
user@snf-15678:~$ docker stop 045bc2f8a731
045bc2f8a731
user@snf-15678:~$
```

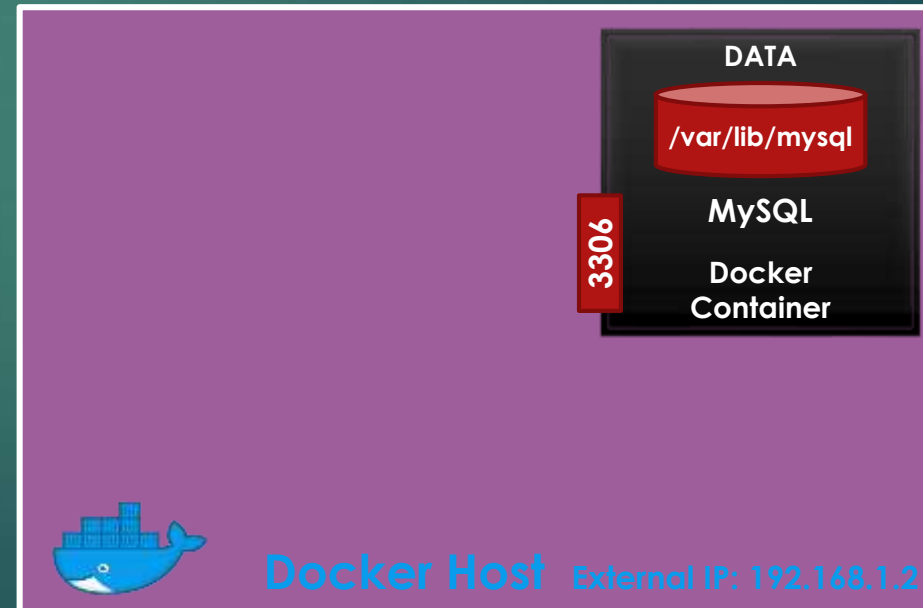


# 2<sup>nd</sup> Docker Tip

## Volume mapping in docker host (3/6)

- Τι γίνεται αν σβήσουμε το instance του MySQL Server; Προφανώς, όλα τα δεδομένα θα σβηστούν καθώς είναι μέσα στο Docker Container.

```
user@snf-15678:~$ docker rm 045bc2f8a731
045bc2f8a731
user@snf-15678:~$
```



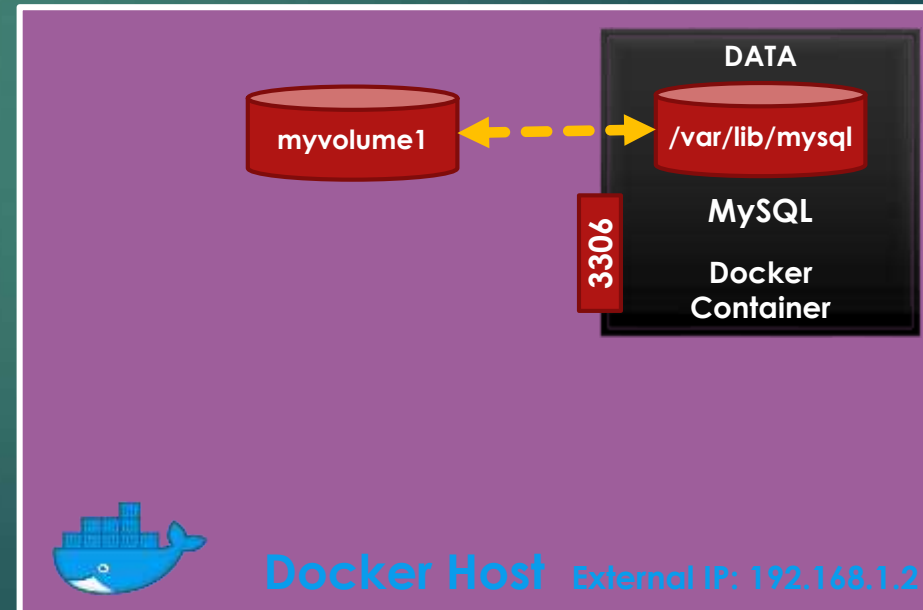
# 2<sup>nd</sup> Docker Tip

## Volume mapping in docker host (4/6)

13

- ▶ Η παρακάτω εντολή ξεκινάει ένα instance MySQL Server, με Volume mapping στο docker Host.

```
user@snf-15678:~$ docker run -d -v myvolume1:/var/lib/mysql mysql  
52c7272889ede27784d3289d99adc042aaf3e1e6cb8fec8f1ae63c399f57b676  
user@snf-15678:~$
```



# 2<sup>nd</sup> Docker Tip

## Volume mapping in docker host (5/6)

14

- Σε αυτή την περίπτωση δεν υπήρχε το persistent volume με το όνομα **myvolume1** στο Host. Έτσι δημιουργήθηκε αυτόματα στο path **/var/lib/docker/volumes/myvolume1/\_data**, όπως φαίνεται παρακάτω.

```
user@snf-15678:~$ docker inspect 52c7272889ed
.....
  "Mounts": [
    {
      "Type": "volume",
      "Name": "myvolume1",
      "Source": "/var/lib/docker/volumes/myvolume1/_data",
      "Destination": "/var/lib/mysql",
      "Driver": "local",
      "Mode": "z",
      "RW": true,
      "Propagation": ""
    }
  ],
  .....
```

```
user@snf-15678:~$
```

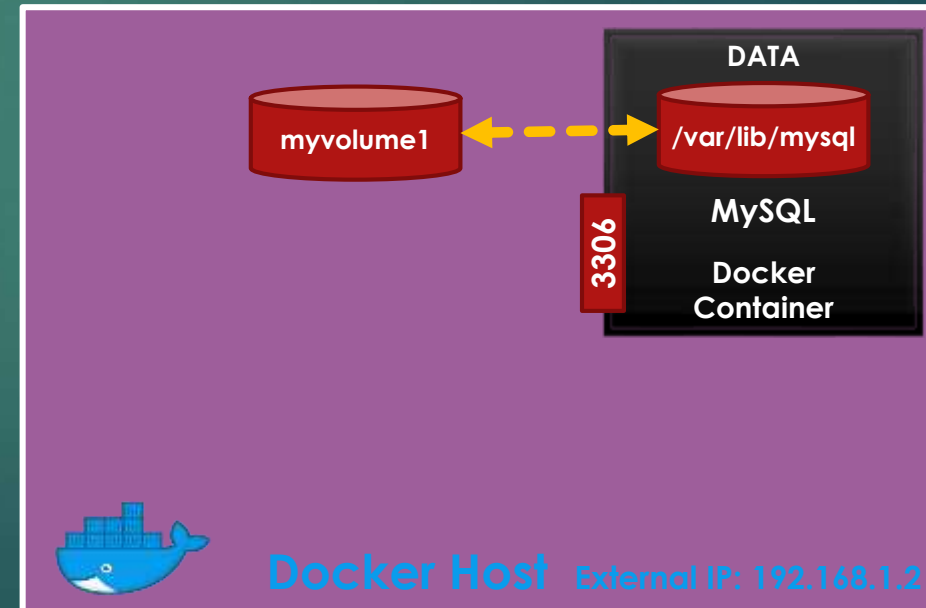
# 2<sup>nd</sup> Docker Tip

## Volume mapping in docker host (6/6)

15

- Τι γίνεται αν σβήσουμε τώρα το instance του MySQL Server; Προφανώς, όλα τα δεδομένα δεν θα σβηστούν από το myvolume1.

```
user@snf-15678:~$ docker container rm 52c7272889ed
52c7272889ed
user@snf-15678:~$
```



# docker

Δημιουργία του πρώτου Docker



# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (1/10)

17

- ▶ Αρχικά θα δούμε την λειτουργία μιας απλής εφαρμογή shell-app που χρησιμοποιεί το stdin και το stdout.
- ▶ Για το σκοπό αυτό έχει δημιουργηθεί ένα repository στο github. Με την παρακάτω εντολή κλωνοποιούμε το Project.

```
user@snf-15678:~$ git clone https://github.com/gregkoul/Dockerized-shellapp-stdin.git
Cloning into 'Dockerized-shellapp-stdin'...
remote: Enumerating objects: 18, done.
remote: Counting objects: 100% (18/18), done.
remote: Compressing objects: 100% (16/16), done.
remote: Total 18 (delta 6), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (18/18), done.
user@snf-15678:~$
```

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (2/10)

- ▶ Μπαίνουμε στο φάκελο `Dockerized-shellapp-stdin`.

```
user@snf-15678:~$ cd ~/Dockerized-shellapp-stdin
user@snf-15678:~/Dockerized-shellapp-stdin$
```

- ▶ Βλέπουμε τα περιεχόμενα του φακέλου.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ ls -la
total 60
drwxr-xr-x 4 gregory gregory 4096 Nov 29 14:28 .
drwxr-xr-x 3 gregory gregory 4096 Nov 29 14:28 ..
drwxr-xr-x 8 gregory gregory 4096 Nov 29 14:28 .git
-rw-r--r-- 1 gregory gregory  104 Nov 29 14:28 Dockerfile
-rw-r--r-- 1 gregory gregory 35149 Nov 29 14:28 LICENSE
-rw-r--r-- 1 gregory gregory   412 Nov 29 14:28 README.md
drwxr-xr-x 2 gregory gregory  4096 Nov 29 14:28 shellapp-stdin
user@snf-15678:~/Dockerized-shellapp-stdin$
```

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (3/10)

- ▶ Στη συνέχεια μπαίνουμε στο φάκελο του Project.

```
user@snf-15678:~/ Dockerized-shellapp-stdin $ cd shellapp-stdin
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin $
```

- ▶ Βλέπουμε τα περιεχόμενα του φακέλου.

```
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin $ ls -la
total 12
drwxr-xr-x 2 gregory gregory 4096 Nov 29 14:28 ./
drwxr-xr-x 4 gregory gregory 4096 Nov 29 14:28 ../
-rw-r--r-- 1 gregory gregory  122 Nov 29 14:28 app.sh
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin $
```

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (4/10)

- ▶ Στη συνέχεια τυπώνουμε το περιεχόμενο του αρχείου `app.sh`, για να καταλάβουμε την λειτουργία του.

```
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$ cat app.sh
echo -n "Welcome to UNIWA! Please enter your name:"
read name
echo " "
echo "Hello $name! You are going to learn docker!"
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$
```

- ▶ Εκτελούμε το shell application.

```
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$ sh app.sh
Welcome to UNIWA! Please enter your name:Gregory

Hello Gregory! You are going to learn docker!
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$
```

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (5/10)

- ▶ Αφού είδαμε την λειτουργία της εφαρμογής shellapp-stdin, θα δούμε πως δημιουργούμε ένα docker image με αυτή την εφαρμογή.
  - ▶ Επιστρέφουμε στο φάκελο **Dockerized-shellapp-stdin**.

```
user@snf-15678:~/Dockerized-shellapp-stdin/shellapp-stdin$ cd ~/Dockerized-shellapp-stdin
user@snf-15678:~/Dockerized-shellapp-stdin$
```

- ▶ Βλέπουμε ξανά τα περιεχόμενα του φακέλου, όπου παρατηρούμε την ύπαρξη του αρχείου **Dockerfile**.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ ls -la
total 60
drwxr-xr-x 4 gregory gregory 4096 Nov 29 14:28 .
drwxr-xr-x 3 gregory gregory 4096 Nov 29 14:28 ..
drwxr-xr-x 8 gregory gregory 4096 Nov 29 14:28 .git
-rw-r--r-- 1 gregory gregory 104 Nov 29 14:28 Dockerfile
-rw-r--r-- 1 gregory gregory 35149 Nov 29 14:28 LICENSE
-rw-r--r-- 1 gregory gregory 412 Nov 29 14:28 README.md
drwxr-xr-x 2 gregory gregory 4096 Nov 29 14:28 shellapp-stdin
user@snf-15678:~/Dockerized-shellapp-stdin$
```

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (6/10)

- ▶ Στη συνέχεια τυπώνουμε το περιεχόμενο του αρχείου **Dockerfile**, για να καταλάβουμε πώς γίνεται το Dockerization.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ cat Dockerfile
FROM alpine:latest
ADD shellapp-stdin/app.sh /opt/app.sh
WORKDIR /opt/
ENTRYPOINT ["sh"]
CMD ["app.sh"]
user@snf-15678:~/Dockerized-shellapp-stdin$
```

- ▶ Αυτό το αρχείο δίνει τις οδηγίες για το πώς πρέπει να πακεταριστεί σε ένα docker image η εφαρμογή **shellapp-stdin**.

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (7/10)

- ▶ Με την εντολή **docker build**, δημιουργούμε ένα νέο docker image.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker build . -t youruser/shellapp-stdin:1.1
Sending build context to Docker daemon 142.3kB
Step 1/5 : FROM alpine:latest
----> d6e46aa2470d
Step 2/5 : ADD shellapp-stdin/app.sh /opt/app.sh
----> 0c927b035dcd
Step 3/5 : WORKDIR /opt/
----> Running in bb6d0a42203a
Removing intermediate container bb6d0a42203a
----> a268d734ab84
Step 4/5 : ENTRYPOINT ["sh"]
----> Running in d041ce569c9f
Removing intermediate container d041ce569c9f
----> 53ca11813d84
Step 5/5 : CMD ["app.sh"]
----> Running in b922a03f2321
Removing intermediate container b922a03f2321
----> f442dbb41154
Successfully built f442dbb41154
Successfully tagged youruser/shellapp-stdin:1.1
user@snf-15678:~/Dockerized-shellapp-stdin$
```

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (8/10)

- ▶ Με την εντολή **docker image ls**, βλέπουμε όλα τα docker images.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
youruser/shellapp-stdin	1.1	f442dbb41154	8 minutes ago	5.57MB
alpine	latest	d6e46aa2470d	5 weeks ago	5.57MB
docker/whalesay	latest	6b362a9f73eb	5 years ago	247MB

```
user@snf-15678:~/Dockerized-shellapp-stdin$
```

- ▶ Με την εντολή **docker run**, δημιουργούμε ένα docker container από ένα docker image.

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker run youruser/shellapp-stdin:1.1
```

Welcome to UNIWA! Please enter your name:  
Hello ! You are going to learn docker!

```
user@snf-15678:~/Dockerized-shellapp-stdin$
```



# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (9/10)

- ▶ Παρατηρούμε ότι δεν μπορούμε να χρησιμοποιήσουμε το πληκτρολόγιο για να βάλουμε είσοδο στο shellapp όπως όταν το τρέχαμε τοπικά. Γιατί συμβαίνει αυτό;
  - ▶ Αυτό συμβαίνει γιατί το docker container από default δεν ακούει σε κάποιο standard input (stdin). Ακόμα και να έχουμε κάνει attach στην κονσόλα του, δεν ακούει καμία είσοδο από το πληκτρολόγιο. Ο λόγος είναι ότι τρέχει σε non-interactive mode.
- ▶ Αν θέλουμε να αντιστοιχίσουμε το stdin και το stdout του host στο docker container, θα πρέπει να τρέξουμε την ίδια εντολή με τις παραμέτρους:
  - ▶ -i : Keep STDIN open even if not attached
  - ▶ -t : Allocate a pseudo-tty

```
user@snf-15678:~/Dockerized-shellapp-stdin$ docker run -it youruser/shellapp-stdin:1.1
Welcome to UNIWA! Please enter your name:Gregory
Hello Gregory! You are going to learn docker!
user@snf-15678:~/Dockerized-shellapp-stdin$
```

# Δημιουργία του πρώτου Docker

## Dockerize a simple Shell Application (10/10)

26

- ▶ Συγχαρητήρια!!
- ▶ Μόλις ολοκληρώσατε επιτυχώς το «πακετάρισμα» σε docker image, της πρώτης σας Shell εφαρμογής.

# Αναφορές

27

- ▶ Poulton, N. (2019). Docker Deep Dive. JJNP Consulting Limited.