



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 LAB09

«Περισσότερα 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

Περιεχόμενα

- ▶ Περισσότερα παραδείγματα εφαρμογών με Docker
 - ▶ 3rd Docker Tip – PORT Mapping in Docker
 - ▶ 4th Docker Tip – Environmental Variables in Docker
 - ▶ 5th Docker Tip – Command Line Arguments in Docker
- ▶ Δημιουργία του δεύτερου Docker

docker

Περισσότερα παραδείγματα εφαρμογών με Docker

docker

3rd Docker Tip

A simple flask webapp demonstrates the port mapping in docker.

3rd Docker Tip

Port mapping in docker host (1/4)

5

University of West Attica - Dr. Grigorios Koulouras

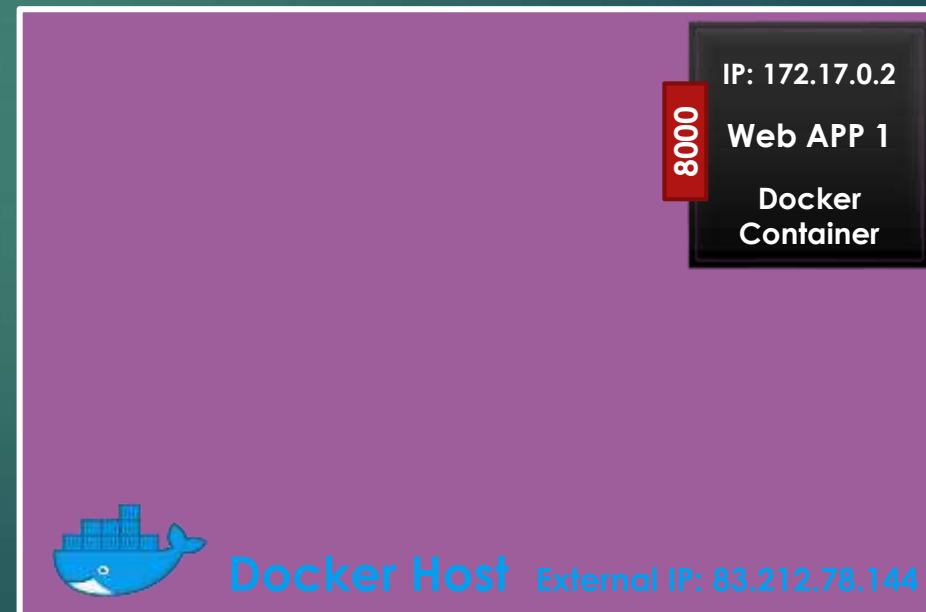
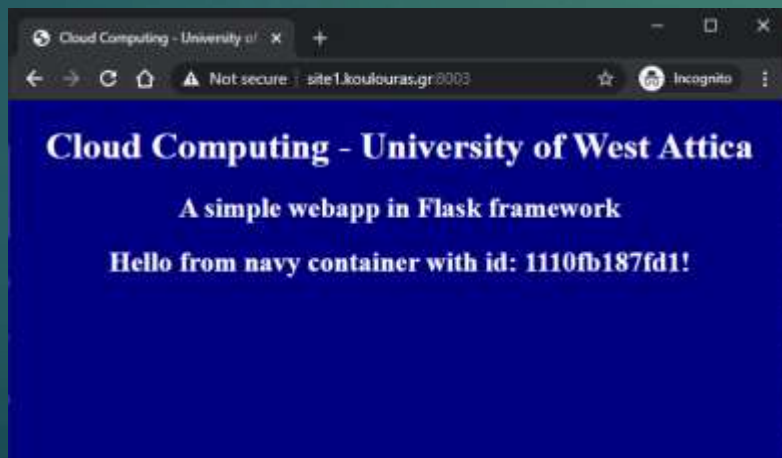
- Για τις ανάγκες του μαθήματος υλοποιήθηκε ένα πολύ απλό Web Application σε Flask, το οποίο σερβίρει μια απλή σελίδα Web.

```
user@snf-15678:~$ docker run --name webapp1 gregkoul/webapp-flask-bgcolor:2.0
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
user@snf-15678:~$
```



http://172.17.0.2:8000

Internal IP



3rd Docker Tip

Port mapping in docker host (2/4)

6

- Για να είναι ορατό το Web Application έξω από το ιδιωτικό περιβάλλον του Host, θα πρέπει να γίνει ένα Port mapping.

```
user@snf-15678:~$ docker run -p 8000:8000 --name webapp2 gregkoul/webapp-flask-bgcolor:2.0
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
user@snf-15678:~$
```



http://172.17.0.2:8000

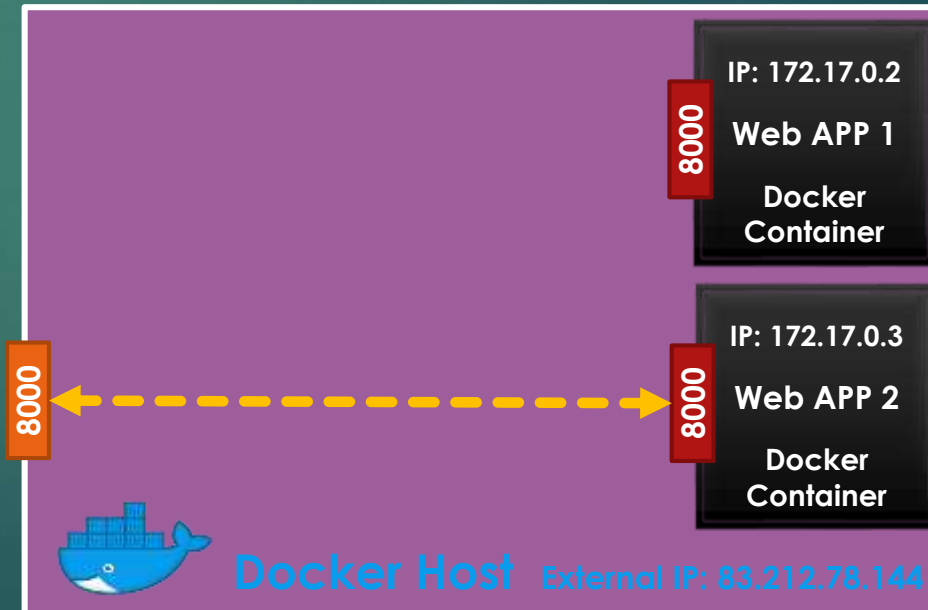
Internal IP

http://172.17.0.3:8000

Internal IP

http://83.212.78.144:8000

External IP



3rd Docker Tip

Port mapping in docker host (3/4)

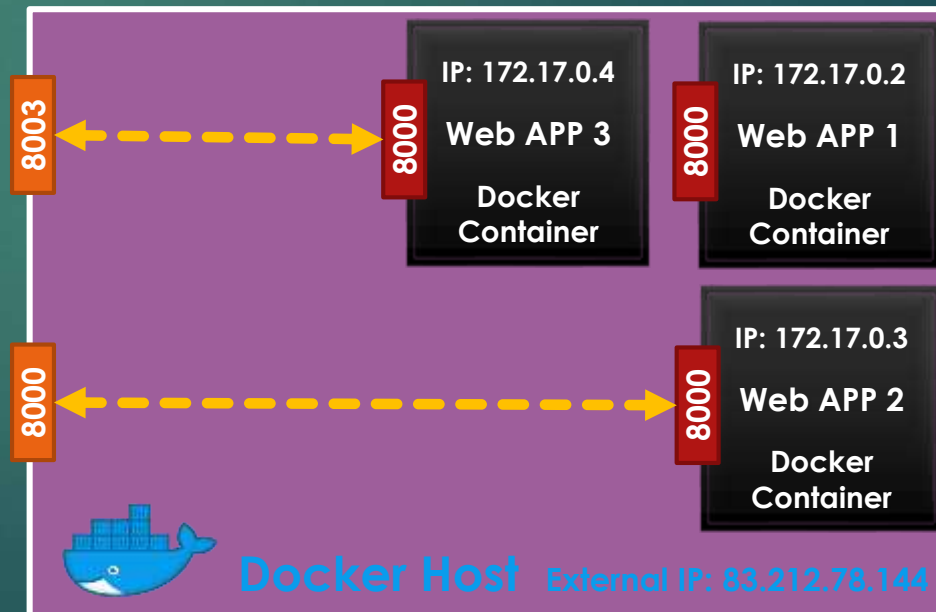
7

- Ένα ακόμα instance του Web Application ξεκινάει και είναι ορατό έξω από το ιδιωτικό περιβάλλον του Host, σε μια διαφορετική πόρτα του.

```
user@snf-15678:~$ docker run -p 8003:8000 --name webapp3 gregkoul/webapp-flask-bgcolor:2.0
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
user@snf-15678:~$
```



http://172.17.0.2:8000	Internal IP
http://172.17.0.3:8000	Internal IP
http://83.212.78.144:8000	External IP
http://172.17.0.4:8000	Internal IP
http://83.212.78.144:8003	External IP



3rd Docker Tip

Port mapping in docker host (4/4)

- ▶ Με αυτό τον τρόπο μπορούν να τρέξουν εκατοντάδες, ακόμα και χιλιάδες instances του Web Application σε ένα μόνο Host.
- ▶ Προσοχή! Το Port Mapping θα πρέπει να γίνεται σε διαφορετική πόρτα του Host!!
- ▶ Το μόνο που χρειάζεται τώρα για να γίνει scale-up το Web Application μας, είναι ένας Load Balancer για να μοιράσει την κίνηση στα instances.

docker

4th Docker Tip

A simple flask webapp demonstrates the Environmental Variables in docker.

4th Docker Tip

Environmental Variables in docker (1/6)

10

- ▶ Πολλές φορές, είναι απαραίτητο για να μπορέσει να γίνει scale-up ένα Web Application, να μπορεί να ρυθμιστεί διαφορετικά το κάθε instance της εφαρμογής.
- ▶ Ένας από τους τρόπους που μπορεί να λυθεί το συγκεκριμένο πρόβλημα είναι η χρήση των Environmental Variables.

4th Docker Tip

Environmental Variables in docker (2/6)

11

- ▶ Για να καταλάβουμε πως δουλεύουν τα Environmental Variables στο Docker, θα πρέπει να γνωρίζουμε πως δουλεύουν γενικά τα Environmental Variables σε ένα Λειτουργικό Σύστημα.
 - ▶ Αν γράψουμε στην γραμμή εντολών (cli) την εντολή `export COLOR=red`, στην πραγματικότητα δημιουργούμε αν δεν υπάρχει ήδη, μια προσωρινή μεταβλητή στο λειτουργικό μας σύστημα και της αναθέτουμε μια τιμή.
 - ▶ Αυτή την τιμή θα μπορούσαμε να την αναζητήσουμε και να την χρησιμοποιήσουμε όποτε επιθυμούμε.

```
user@snf-15678:~$ export COLOR=red
user@snf-15678:~$ echo $COLOR
red
```

- ▶ Με τον ίδιο τρόπο θα μπορούσε να την αναζητήσει και οποιοδήποτε πρόγραμμα γραμμένο σε Python, NodeJS κτλ.

4th Docker Tip

Environmental Variables in docker (3/6)

12

- ▶ Στην Python για παράδειγμα ο παρακάτω κώδικας θα τύπωνε την τιμή της μεταβλητής **COLOR** στο τερματικό μας εφόσον είχε αποκτήσει τιμή νωρίτερα.

testenv.py

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

color = os.environ.get('COLOR')
print(color)
```

```
user@snf-15678:~$ export COLOR=red
user@snf-15678:~$ echo $COLOR
red
user@snf-15678:~$ python3 testenv.py
red
```

4th Docker Tip

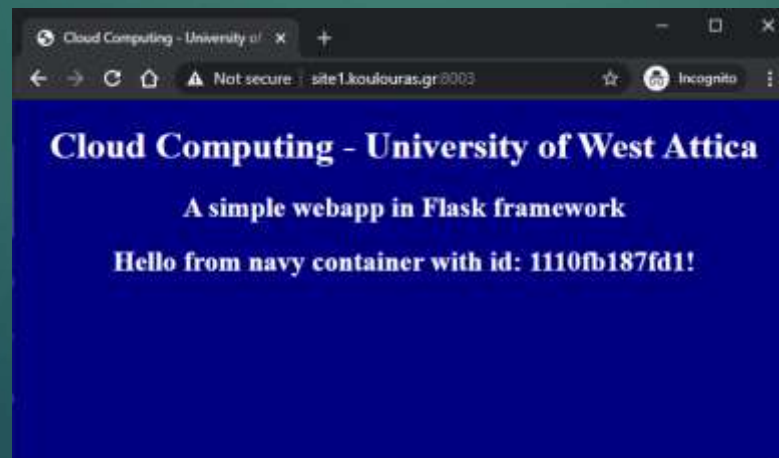
Environmental Variables in docker (4/6)

13

University of West Attica - Dr. Grigorios Koulouras

- Για τις ανάγκες του μαθήματος υλοποιήθηκε ένα πολύ απλό Web Application σε Flask, το οποίο σερβίρει μια απλή σελίδα Web. Κάθε φορά που ξεκινάει ένα νέο instance της εφαρμογής, εμφανίζεται ένα τυχαίο Background Color ανάμεσα από 6 χρώματα μίας λίστας.

```
user@snf-15678:~$ docker run -p 8003:8000 gregkoul/webapp-flask-bgcolor:2.0
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
user@snf-15678:~$
```



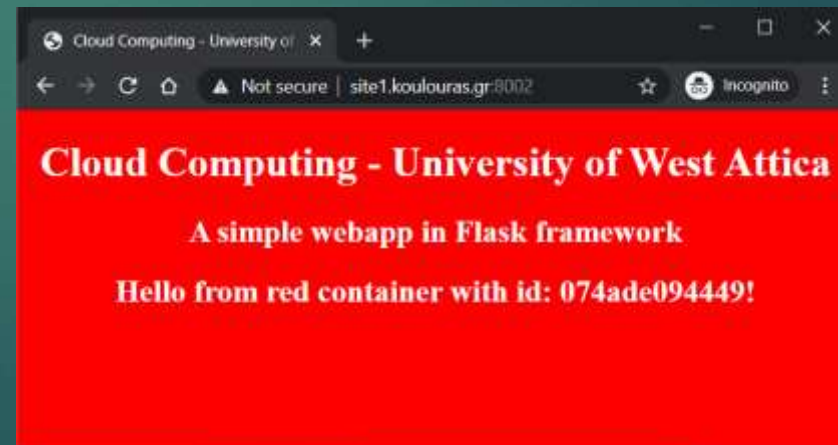
4th Docker Tip

Environmental Variables in docker (5/6)

14

- ▶ Πως μπορεί ο χρήστης να επιλέξει το Background Color στο instance της εφαρμογής που θέλει να εκκινήσει;
 - ▶ Ένας τρόπος είναι η χρήση των Environmental Variables. Ας δούμε το παρακάτω παράδειγμα και ας συζητήσουμε πως δουλεύει.

```
user@snf-15678:~$ docker run -d -p 8001:8000 -e APP_COLOR=navy gregkoul/webapp-flask-bgcolor:2.0  
bcce102d743cf9e1f987b9a66906a5c3c6ae61f76e1db07e404b18800a5ee3f0  
user@snf-15678:~$ docker run -d -p 8002:8000 -e APP_COLOR=red gregkoul/webapp-flask-bgcolor:2.0  
074ade094449ad540babdb6a8eaff2274534a1597b9d560b8e2e643843b3cbdd
```



4th Docker Tip

Environmental Variables in docker (6/6)

15

- ▶ Συνεπώς, με την χρήση των Environmental Variables στην τεχνολογία Docker, μπορεί ένας χρήστης να επιλέξει να μεταβάλλει κάποιες από τις παραμέτρους που αφορούν τη λειτουργία κάθε instance της εφαρμογής.
- ▶ Στο συγκεκριμένο παράδειγμα που παρουσιάστηκε, ο χρήστης μπορεί να αλλάξει το Background Color και το μήνυμα καλωσορίσματος σε κάθε instance της εφαρμογής που εκκινεί, με την χρήση των Environmental Variables.

docker

5th Docker Tip

A simple flask webapp demonstrates the Command Line Arguments in docker.

5th Docker Tip

Command Line Arguments in docker (1/8)

17

- ▶ Όπως έχουμε αναφέρει και νωρίτερα, πολλές φορές, είναι απαραίτητο για να μπορέσει να γίνει scale-up ένα Web Application, να μπορεί να ρυθμιστεί διαφορετικά το κάθε instance της εφαρμογής.
- ▶ Ένας ακόμα από τους τρόπους που μπορεί να λυθεί το συγκεκριμένο πρόβλημα, εκτός από τη χρήση των Environmental Variables που εξετάσαμε νωρίτερα, είναι η χρήση των Command Line Arguments.

5th Docker Tip

Command Line Arguments in docker (2/8)

18

- ▶ Για να καταλάβουμε πως δουλεύουν τα Command Line Arguments στο Docker, θα πρέπει να γνωρίζουμε πως δουλεύουν γενικά τα Command Line Arguments στην γραμμή εντολών.
 - ▶ Αν γράψουμε στην γραμμή εντολών (cli) την εντολή **sleep 5**, στην πραγματικότητα τρέχει το πρόγραμμα sleep που παίρνει μια παράμετρο ή αλλιώς argument, για να κάνει τόση καθυστέρηση όση και η τιμή αυτής της παραμέτρου σε δευτερόλεπτα.

```
user@snf-15678:~$ sleep 5
user@snf-15678:~$
```

5th Docker Tip

Command Line Arguments in docker (3/8)

19

- ▶ Αντίστοιχα αν θέλουμε να δούμε μια μικρή βοήθεια για την σύνταξη της παραπάνω εντολής, θα γράφαμε `sleep --help`. Το argument εδώ είναι το `--help`. Τις παραμέτρους αυτές τις χρησιμοποιεί εσωτερικά το πρόγραμμα για να ρυθμίσει την λειτουργία του.

```
user@snf-15678:~$ sleep --help
Usage: sleep NUMBER[SUFFIX]...
       or:  sleep OPTION
Pause for NUMBER seconds.  SUFFIX may be 's' for seconds (the default),
'm' for minutes, 'h' for hours or 'd' for days.  Unlike most implementations
that require NUMBER be an integer, here NUMBER may be an arbitrary floating
point number.  Given two or more arguments, pause for the amount of time
specified by the sum of their values.

       --help      display this help and exit
       --version   output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/sleep>
or available locally via: info '(coreutils) sleep invocation'
user@snf-15678:~$
```

5th Docker Tip

Command Line Arguments in docker (4/8)

- ▶ Με τον ίδιο τρόπο δουλεύουν όλα τα προγράμματα που δέχονται παραμέτρους από την γραμμή εντολών και για να προσαρμόσουν την λειτουργία τους.

5th Docker Tip

Command Line Arguments in docker (5/8)

21

- ▶ Στην Python για παράδειγμα ο παρακάτω κώδικας θα τύπωνε την τιμή της μεταβλητής `--color` στο τερματικό μας εφόσον είχε αποκτήσει τιμή κατά την εκτέλεση του προγράμματος από το cli.

testarg.py

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

parser = argparse.ArgumentParser()
parser.add_argument('--color', required=True)
args = parser.parse_args()
if args.color:
    print(args.color)
```

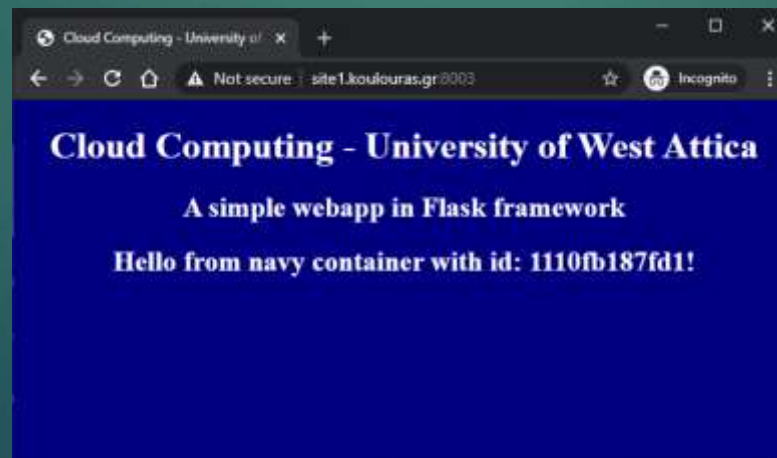
```
user@snf-15678:~$ python3 testarg.py
usage: testarg.py [-h] --color COLOR
testarg.py: error: the following arguments are required: --color
user@snf-15678:~$ python3 testarg.py --color=red
red
```

5th Docker Tip

Command Line Arguments in docker (6/8)

- ▶ Για τις ανάγκες του μαθήματος υλοποιήθηκε ένα πολύ απλό Web Application σε Flask, το οποίο σερβίρει μια απλή σελίδα Web. Κάθε φορά που ξεκινάει ένα νέο instance της εφαρμογής, εμφανίζετε ένα τυχαίο Background Color ανάμεσα από 6 χρώματα μίας λίστας.

```
user@snf-15678:~$ docker run -p 8003:8000 gregkoul/webapp-flask-bgcolor:2.0
* Running on http://0.0.0.0:8000/ (Press CTRL+C to quit)
user@snf-15678:~$
```



5th Docker Tip

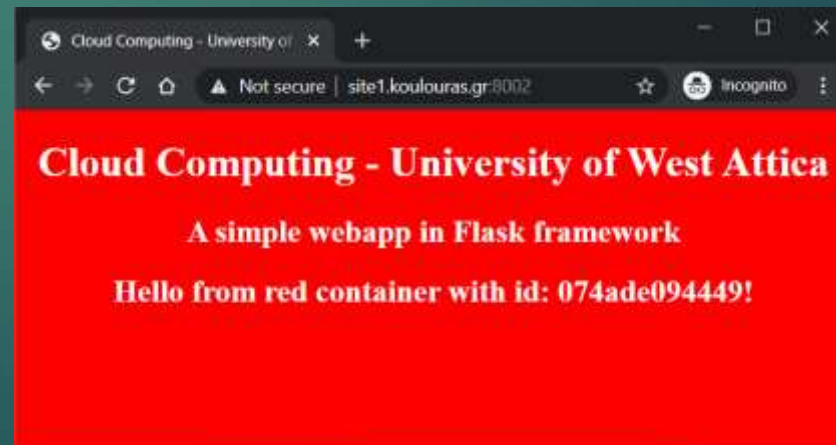
Command Line Arguments in docker (7/8)

23

University of West Attica - Dr. Grigorios Koulouras

- ▶ Πως μπορεί ο χρήστης να επιλέξει το Background Color στο instance της εφαρμογής που θέλει να εκκινήσει;
 - ▶ Ένας τρόπος είναι η χρήση των Command Line Arguments. Ας δούμε το παρακάτω παράδειγμα και ας συζητήσουμε πως δουλεύει.

```
user@snf-15678:~$ docker run -d -p 8001:8000 gregkoul/webapp-flask-bgcolor:2.0 --color=navy  
bcce102d743cf9e1f987b9a66906a5c3c6ae61f76e1db07e404b18800a5ee3f0  
user@snf-15678:~$ docker run -d -p 8002:8000 gregkoul/webapp-flask-bgcolor:2.0 --color=red  
074ade094449ad540babdb6a8eaff2274534a1597b9d560b8e2e643843b3cbdd
```



5th Docker Tip

Command Line Arguments in docker (8/8)

24

- ▶ Συνεπώς, με την χρήση των Command Line Arguments στην τεχνολογία Docker, μπορεί ένας χρήστης να επιλέξει να μεταβάλλει κάποιες από τις παραμέτρους που αφορούν τη λειτουργία κάθε instance της εφαρμογής.
- ▶ Στο συγκεκριμένο παράδειγμα που παρουσιάστηκε, ο χρήστης μπορεί να αλλάξει το Background Color και το μήνυμα καλωσορίσματος σε κάθε instance της εφαρμογής που εκκινεί με την χρήση των Command Line Arguments.

docker

Δημιουργία του δεύτερου Docker

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (1/15)

- ▶ Στο σημείο αυτό, θα δούμε την λειτουργία μιας απλής εφαρμογής flask webapp που απεικονίζει μια σελίδα web με δυναμικό έγχρωμο background και δυναμικό μήνυμα καλωσορίσματος.
 - ▶ 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-webapp-flask-bgcolor.git
Cloning into 'Dockerized-webapp-flask-bgcolor'...
remote: Enumerating objects: 38, done.
remote: Counting objects: 100% (38/38), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 38 (delta 15), reused 17 (delta 4), pack-reused 0
Unpacking objects: 100% (38/38), 20.29 KiB | 1.07 MiB/s, done.
user@snf-15678:~$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (2/15)

- ▶ Μπαίνουμε στο φάκελο `Dockerized-webapp-flask-bgcolor`.

```
user@snf-15678:~$ cd ~/Dockerized-webapp-flask-bgcolor
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```

- ▶ Βλέπουμε τα περιεχόμενα του φακέλου.

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$ ls -la
total 60
drwxrwxr-x 4 user user 4096 Dec 7 05:25 ./
drwxrwxr-x 3 user user 4096 Dec 7 05:25 ../
-rw-rw-r-- 1 user user 186 Dec 7 05:25 Dockerfile
drwxrwxr-x 8 user user 4096 Dec 7 05:25 .git/
-rw-rw-r-- 1 user user 35149 Dec 7 05:25 LICENSE
-rw-rw-r-- 1 user user 2536 Dec 7 05:25 README.md
drwxrwxr-x 3 user user 4096 Dec 7 05:25 webapp-flask-bgcolor/
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (3/15)

- Στη συνέχεια μπαίνουμε στο φάκελο του Project.

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$ cd webapp-flask-bgcolor
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$
```

- Βλέπουμε τα περιεχόμενα του φακέλου.

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$ ls -la
total 20
drwxrwxr-x 3 user user 4096 Dec  7 05:25 ./
drwxrwxr-x 4 user user 4096 Dec  7 05:25 ../
-rw-rw-r-- 1 user user 3264 Dec  7 05:25 app.py
-rw-rw-r-- 1 user user  13 Dec  7 05:25 requirements.txt
drwxrwxr-x 2 user user 4096 Dec  7 05:25 templates/
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (4/15)

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

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$ cat app.py
from flask import Flask, render_template
import socket
import random
import os
import argparse

app = Flask(__name__)
.....
# Run Flask Application
app.run(host="0.0.0.0", port=8000)
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (5/15)

- ▶ Στη συνέχεια τυπώνουμε το περιεχόμενο του αρχείου `requirements.txt`, για να δούμε τα προαπαιτούμενα πακέτα της python εφαρμογής μας.
- ▶ Για να τρέξει κάποιος την εφαρμογή τοπικά, θα πρέπει προηγουμένως να γίνει εγκατάσταση αυτών των πακέτων τοπικά.

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$ cat requirements.txt
Flask==1.1.1
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (6/15)

- ▶ Αυτό **ΔΕΝ** θα το κάνουμε σε αυτό το σημείο, καθώς δεν επιθυμούμε να γεμίσουμε τον τοπικό μας υπολογιστή με «σκουπίδια».
- ▶ Σε κάθε περίπτωση, αν θέλαμε να τις εγκαταστήσουμε τοπικά, θα εκτελούσαμε την παρακάτω εντολή.

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$ pip install -r requirements.txt
.....
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$
```

- ▶ Εφόσον έχουμε εγκαταστήσει τις βιβλιοθήκες τοπικά, μπορούμε να εκτελέσουμε και την εφαρμογή μας τοπικά.

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$ python app.py
.....
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (7/15)

- ▶ Αφού είδαμε πως λειτουργεί και πως εκτελείται τοπικά η εφαρμογή webapp-flask-bgcolor, θα δούμε πως δημιουργούμε ένα docker image με αυτή την εφαρμογή.
 - ▶ Επιστρέφουμε στο φάκελο **Dockerized-webapp-flask-bgcolor**.

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor/webapp-flask-bgcolor$ cd ..  
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```

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

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$ ls -la  
total 60  
drwxrwxr-x 4 user user 4096 Dec 7 05:25 ./  
drwxrwxr-x 3 user user 4096 Dec 7 05:25 ../  
-rw-rw-r-- 1 user user 186 Dec 7 05:25 Dockerfile  
drwxrwxr-x 8 user user 4096 Dec 7 05:25 .git/  
-rw-rw-r-- 1 user user 35149 Dec 7 05:25 LICENSE  
-rw-rw-r-- 1 user user 2536 Dec 7 05:25 README.md  
drwxrwxr-x 3 user user 4096 Dec 7 05:25 webapp-flask-bgcolor/  
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```


Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (8/15)

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

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$ cat Dockerfile
FROM python:3.6-alpine
MAINTAINER gregkoul@gmail.com
COPY webapp-flask-bgcolor/. /opt/
WORKDIR /opt
RUN pip install -r requirements.txt
EXPOSE 8000
ENTRYPOINT ["python", "app.py"]
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```

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

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (9/15)

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

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$ docker build . -t webapp-flask-bgcolor:2.0
Sending build context to Docker daemon 175.1kB
Step 1/7 : FROM python:3.6-alpine
---> 32ec068d92f7
Step 2/7 : MAINTAINER gregkoul@gmail.com
---> a37d7b0171d7
Step 3/7 : COPY webapp-flask-bgcolor/. /opt/
---> ae30dd96c897
Step 4/7 : WORKDIR /opt
---> ad61add12853
Step 5/7 : RUN pip install -r requirements.txt
---> 4616e37e2a25
Step 6/7 : EXPOSE 8000
---> e520821cb3c6
Step 7/7 : ENTRYPOINT ["python", "app.py"]
---> 954b00e3b15b
Successfully built 954b00e3b15b
Successfully tagged webapp-flask-bgcolor:2.0
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (10/15)

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

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$ docker image ls
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
webapp-flask-bgcolor	2.0	954b00e3b15b	8 minutes ago	51.5MB
python	3.6-alpine	32ec068d92f7	12 days ago	40.8MB
alpine	latest	d6e46aa2470d	5 weeks ago	5.57MB
docker/whalesay	latest	6b362a9f73eb	5 years ago	247MB

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```

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

```
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$ docker run -d -p 8000:8000 webapp-flask-bgcolor:2.0
560895d96ebe836d6c99740235e58c78a39081f92fd0d5a39fd1ea308e24280f
user@snf-15678:~/Dockerized-webapp-flask-bgcolor$
```

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (11/15)

36

- ▶ Στην συγκεκριμένη εφαρμογή ο χρήστης μπορεί να επιλέξει δυναμικά το Background Color καθώς και το μήνυμα καλωσορίσματος στο κάθε instance της εφαρμογής που εκκινεί. Αυτό μπορεί να γίνει με δύο διαφορετικούς τρόπους:
 - a) Ο ένας τρόπος είναι με τη χρήση των Environmental Variables.
 - b) Ο άλλος τρόπος είναι με τη χρήση των Command Line Arguments.

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (12/15)

37

University of West Attica - Dr. Grigorios Koulouras

- ▶ Docker Containers με χρήση των Environmental Variables.

```
user@snf-15678:~$ docker run -d -p 8001:8000 -e APP_COLOR=navy webapp-flask-bgcolor:2.0  
bcce102d743cf9e1f987b9a66906a5c3c6ae61f76e1db07e404b18800a5ee3f0  
user@snf-15678:~$ docker run -d -p 8002:8000 -e APP_COLOR=red webapp-flask-bgcolor:2.0  
074ade094449ad540babdb6a8eaff2274534a1597b9d560b8e2e643843b3cbdd
```



Δημιουργία του δεύτερου Docker

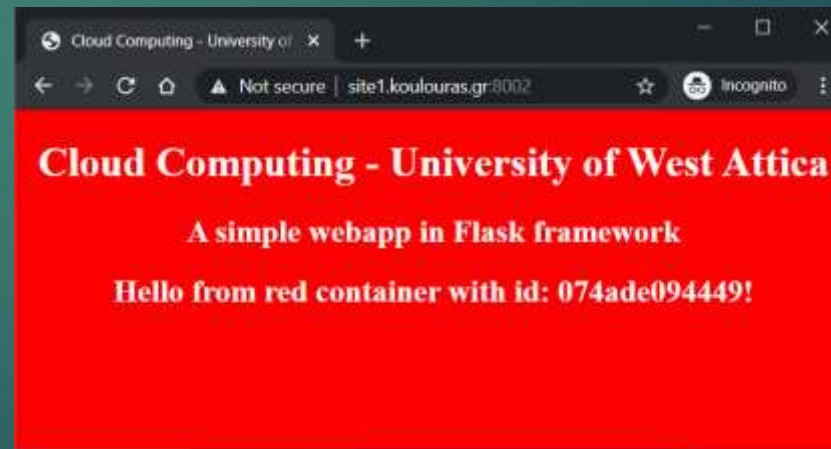
Dockerize a simple flask webapp (13/15)

38

University of West Attica - Dr. Grigorios Koulouras

- Docker Containers με χρήση των Command Line Arguments.

```
user@snf-15678:~$ docker run -d -p 8001:8000 webapp-flask-bgcolor:2.0 --color=navy  
bcce102d743cf9e1f987b9a66906a5c3c6ae61f76e1db07e404b18800a5ee3f0  
user@snf-15678:~$ docker run -d -p 8002:8000 webapp-flask-bgcolor:2.0 --color=red  
074ade094449ad540babdb6a8eaff2274534a1597b9d560b8e2e643843b3cbdd
```



Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (14/15)

- ▶ Παρατηρούμε ότι μπορούμε να αλλάξουμε το Background Color στο κάθε instance της εφαρμογής που εκκινεί με δύο διαφορετικούς τρόπους:
 - a) Επηρεάζοντας την Environmental Variable **APP_COLOR**
 - b) Κάνοντας χρήση του Command Line Argument **--color**
- ▶ Καλείστε να δημιουργήσετε 2 νέα instances της εφαρμογής, αλλάζοντας αυτή την φορά ταυτόχρονα και το μήνυμα καλωσορίσματος. Για το σκοπό αυτό θα γίνει χρήση των:
 - a) Environmental Variable **APP_TITLE**
 - b) Command Line Argument **--title**

Δημιουργία του δεύτερου Docker

Dockerize a simple flask webapp (15/15)

- ▶ Συγχαρητήρια!!
- ▶ Μόλις ολοκληρώσατε επιτυχώς το «πακετάρισμα» σε docker image, της δεύτερης σας εφαρμογής, αυτή τη φορά γραμμένη στο Framework Flask της Python.

Αναφορές

41

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