



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 LAB13
«Παραδείγματα με Docker Compose»

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

- ▶ Παραδείγματα με Docker Compose
 - ▶ 1st Docker Compose Tip – Nginx and volume mounting
 - ▶ 2nd Docker Compose Tip – Web Hit Count
- ▶ Conclusions
- ▶ References

docker

Παραδείγματα για Docker Compose

docker-compose

1st Docker Compose Tip

This example demonstrates the “volume mounting” for persistent data with Docker Compose

1st Docker Compose Tip

Nginx and volume mounting (1/5)

- ▶ Σε αυτό το παράδειγμα θα προσπαθήσουμε να δημιουργήσουμε ένα dockerized Nginx web server, με την βοήθεια του Docker Compose.
- ▶ Αρχικά, δημιουργούμε ένα φάκελο για το project “dc-website3” μέσα στο home directory του χρήστη.

```
user@snf-15678:~$ mkdir -p ~/myprojects/dc-website3
user@snf-15678:~$ cd ~/myprojects/dc-website3
user@snf-15678:~/myprojects/dc-website3$
```

- ▶ Στη συνέχεια, δημιουργούμε το αρχείο **docker-compose.yml** και αντιγράφουμε το περιεχόμενο που απεικονίζεται στην επόμενη διαφάνεια.

```
user@snf-15678:~/myprojects/dc-website3$ nano docker-compose.yml
```

1st Docker Compose Tip

Nginx and volume mounting (2/5)

- Το περιεχόμενο του αρχείου `docker-compose.yml` απεικονίζεται παρακάτω.

```
version: '3.2'
services:
  web:
    container_name: 'website3'
    image: 'nginx:1.19.5'
    ports:
      - '9003:80'
    volumes:
      - website3-vol1:/usr/share/nginx/html
    environment:
      - NGINX_HOST=website3.example.com
      - NGINX_PORT=80
    restart: unless-stopped
volumes:
  website3-vol1:
```

1st Docker Compose Tip

Nginx and volume mounting (3/5)

7

- ▶ Αφού έχουμε αποθηκεύσει το αρχείο, μπορούμε να ξεκινήσουμε την εφαρμογή μας, με μόνο μια εντολή!!!

```
user@snf-15678:~/myprojects/dc-website3$ docker-compose up -d
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm. All containers will be
scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

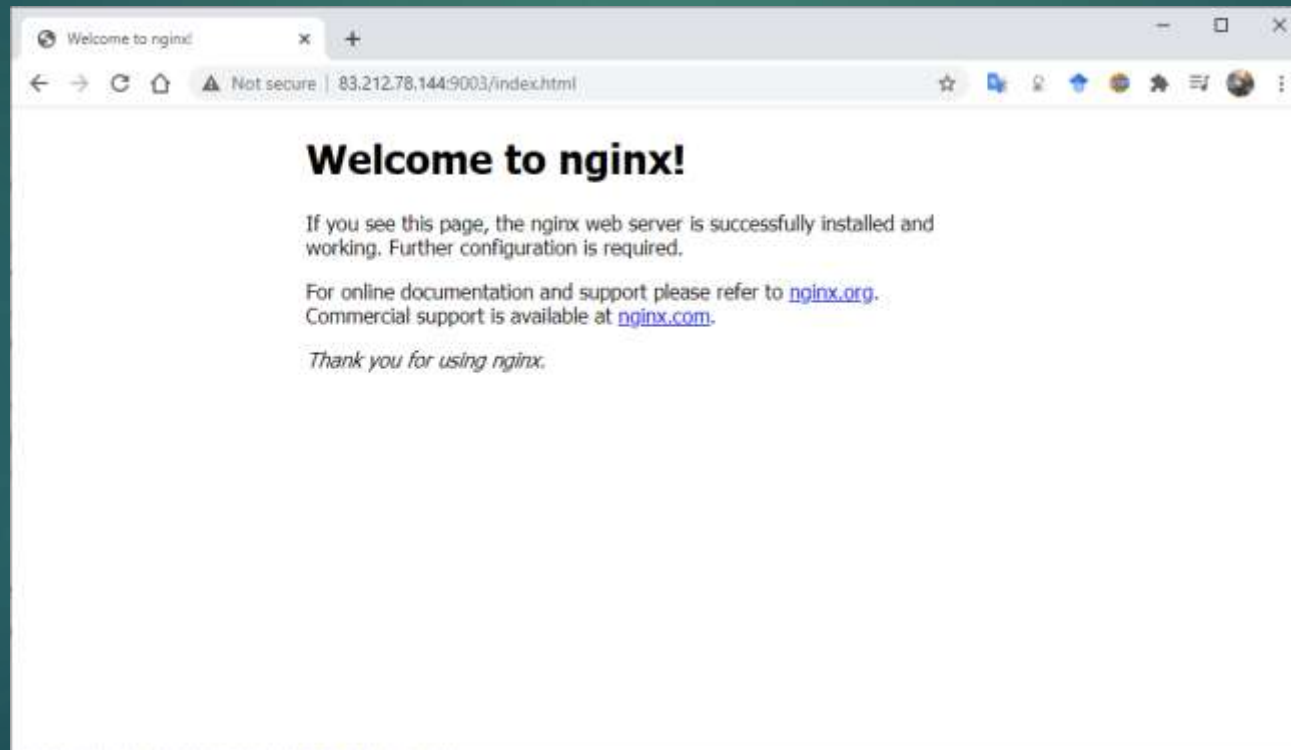
Creating network "dc-website3_default" with the default driver
Creating volume "dc-website3_website3-vol1" with default driver
Creating website3 ... Done
user@snf-15678:~/myprojects/dc-website3$
```

1st Docker Compose Tip

Nginx and volume mounting (4/5)

8

- ▶ Πλοηγούμαστε στη διεύθυνση: <http://your-server-real-ip:9003>



1st Docker Compose Tip

Nginx and volume mounting (5/5)

- ▶ Συγχαρητήρια!!
- ▶ Μόλις ολοκληρώσατε τη δημιουργία ενός dockerized Nginx web server, με την βοήθεια του Docker Compose!
- ▶ Το επόμενο βήμα είναι να αλλάξετε το περιεχόμενο των σελίδων. Εφόσον δεν έχουν ενσωματωθεί μέσα στο ίδιο το container, απλά αντιγράφουμε τα αρχεία της σελίδας μας στο Volume με το όνομα dc-website3_website3-vol1 που δημιουργήθηκε αυτόματα από το Docker Compose.

docker-compose

2nd Docker Compose Tip

This example demonstrates a web app with two microservices by using Docker Compose

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (1/11)

11

- ▶ Η εφαρμογή που θα υλοποιηθεί ονομάστηκε “Web Hit Count”.
- ▶ Σκοπός της είναι να διατηρεί ένα μετρητή επισκεψιμότητας της σελίδας μας με τη βοήθεια του Redis.
- ▶ Αυτή η απλή web εφαρμογή αποτελείται από δύο μικροϋπηρεσίες.
 - ▶ Η πρώτη μικροϋπηρεσία της εφαρμογής μας, είναι ένα web site γραμμένο στη γλώσσα Python και πιο συγκεκριμένα έχει υλοποιηθεί με την βοήθεια του Framework Flask.
 - ▶ Η δεύτερη μικροϋπηρεσία της εφαρμογής μας, είναι το γνωστό σε όλους Redis. Το Redis μπορεί να χρησιμοποιηθεί και σαν in-memory Database.

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (2/11)

12

- ▶ Με την βοήθεια του docker-compose θα δημιουργηθεί αυτή η απλή web εφαρμογή που αποτελείται από δύο μικροϋπηρεσίες.
- ▶ Παρόλο που αυτό το παράδειγμα χρησιμοποιεί το Framework Flask της Python και την υπηρεσία Redis, οι έννοιες που θα παρουσιαστούν εδώ, θα είναι κατανοητές ακόμα και σε αυτούς που δεν είναι εξοικειωμένοι με αυτά.

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (3/11)

13

- ▶ Δημιουργούμε το φάκελο του project “Web Hit Count”.

```
user@snf-15678:~$ mkdir -p ~/myprojects/dc-website4-hitcnt  
user@snf-15678:~$ cd ~/myprojects/dc-website4-hitcnt  
user@snf-15678:~/myprojects/dc-website4-hitcnt$
```

- ▶ Δημιουργούμε το αρχείο `app.py` και αντιγράφουμε το περιεχόμενο που απεικονίζεται στην επόμενη διαφάνεια.

```
user@snf-15678:~/myprojects/dc-website4-hitcnt$ nano app.py
```

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (4/11)

14

- Το περιεχόμενο του αρχείου `app.py` του project “Web Hit Count”.

```
import time
import redis
from flask import Flask

app = Flask(__name__)
cache = redis.Redis(host='redis', port=6379)

def get_hit_count():
    retries = 5
    while True:
        try:
            return cache.incr('hits')
        except redis.exceptions.ConnectionError as conerr:
            if retries == 0:
                raise conerr
            retries -= 1
            time.sleep(0.5)

@app.route('/')
def hello():
    count = get_hit_count()
    return 'Hello World! I have been seen {} times.\n'.format(count)
```

app.py

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (5/11)

15

- ▶ Δημιουργούμε το αρχείο **requirements.txt** και αντιγράφουμε το περιεχόμενο που απεικονίζεται στη συνέχεια.

```
user@snf-15678:~/myprojects/dc-website4-hitcnt$ nano requirements.txt
```

- ▶ Το αρχείο **requirements.txt**, περιέχει τις βιβλιοθήκες της Python που έχουν χρησιμοποιηθεί στο project “Web Hit Count”.

```
redis==3.5.3  
Flask==1.1.2
```

requirements.txt

- ▶ <https://pypi.org/project/redis/>
- ▶ <https://pypi.org/project/Flask/>

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (6/11)

16

- ▶ Δημιουργούμε το αρχείο **Dockerfile** και αντιγράφουμε το περιεχόμενο που απεικονίζεται στη συνέχεια.

```
user@snf-15678:~/myprojects/dc-website4-hitcnt$ nano Dockerfile
```

- ▶ Δημιουργία του Dockerfile του project “Web Hit Count”.

```
FROM python:3.7-alpine
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
WORKDIR /code
COPY requirements.txt /code/requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY app.py /code/app.py
CMD ["flask", "run"]
```

Dockerfile

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (7/11)

17

- ▶ Δημιουργούμε το αρχείο `docker-compose.yml` και αντιγράφουμε το περιεχόμενο που απεικονίζεται στη συνέχεια.

```
user@snf-15678:~/myprojects/dc-website4-hitcnt$ nano docker-compose.yml
```

- ▶ Δημιουργία του `docker-compose.yml` file του project “Web Hit Count”.

```
version: "3.9"
services:
  web:
    build: .
    ports:
      - "9004:5000"
  redis:
    image: "redis:6.0.10-alpine3.12"
```

`docker-compose.yml`

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (8/11)

18

- Τα αρχεία του τελικού Project.

```
user@snf-15678:~/myprojects/dc-website4-hitcnt$ ls -la
total 24
drwxrwxr-x  2 user user 4096 Jan 18 05:41 .
drwxrwxr-x 20 user user 4096 Jan 18 05:28 ..
-rw-rw-r--  1 user user  520 Jan 18 05:31 app.py
-rw-rw-r--  1 user user  111 Jan 18 05:41 docker-compose.yml
-rw-rw-r--  1 user user  253 Jan 18 05:39 Dockerfile
-rw-rw-r--  1 user user   24 Jan 18 05:33 requirements.txt
user@snf-15678:~/myprojects/dc-website4-hitcnt$
```

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (9/11)

19

- ▶ Δημιουργία των docker image του project “Web Hit Count”, δημιουργία των Container και εκκίνησή τους για την υλοποίηση της εφαρμογής.

```
user@snf-15678:~/myprojects/dc-website4-hitcnt$ docker-compose up -d
WARNING: The Docker Engine you're using is running in swarm mode.

Compose does not use swarm mode to deploy services to multiple nodes in a swarm. All containers will be scheduled on the current node.

To deploy your application across the swarm, use `docker stack deploy`.

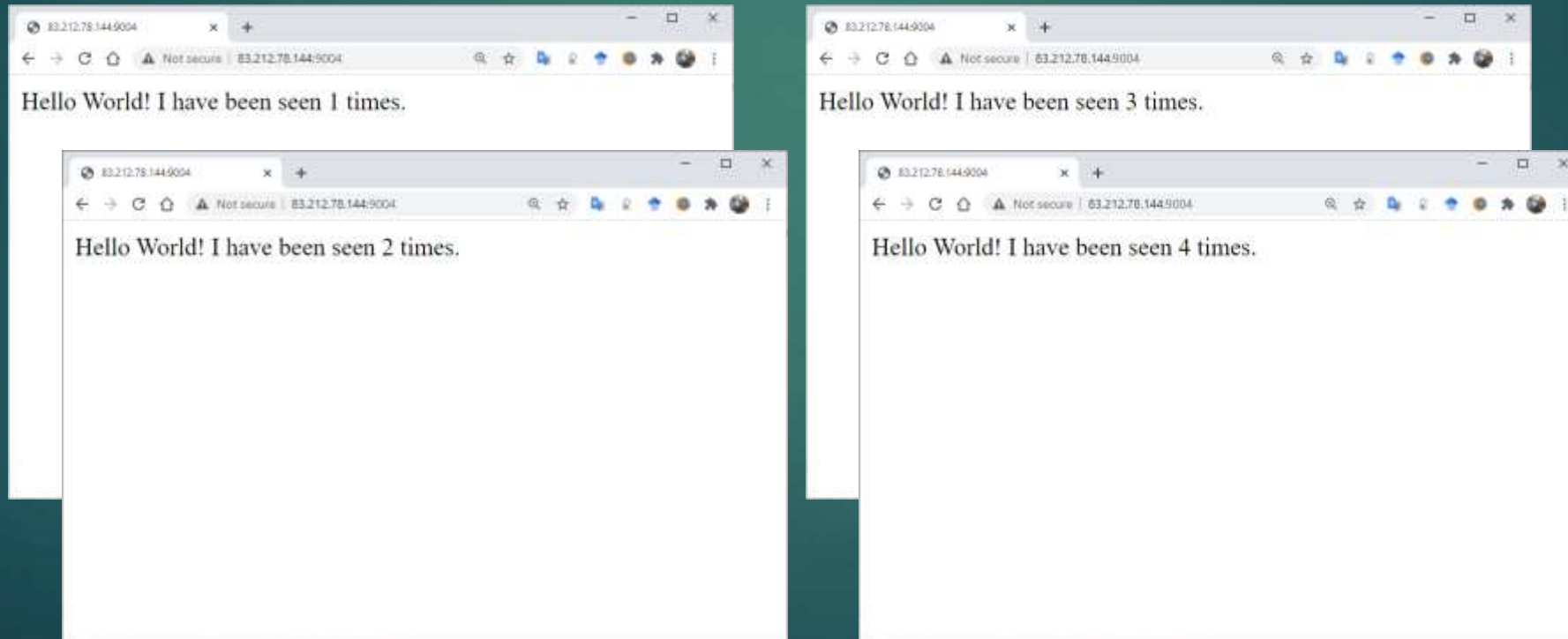
Creating network "dc-website4-hitcnt_default" with the default driver
Creating dc-website4-hitcnt_redis_1 ... done
Creating dc-website4-hitcnt_web_1   ... done
user@snf-15678:~/myprojects/dc-website4-hitcnt$
```

2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (10/11)

20

- ▶ Πλοηγούμαστε στη διεύθυνση: <http://your-server-real-ip:9004> και κάθε φορά που κάνουμε refresh βλέπουμε ότι αυξάνει κατά 1 ο μετρητής «hit count».



2nd Docker Compose Tip

Flask web-app, Redis and docker-compose (11/11)

21

- ▶ Συγχαρητήρια!!
- ▶ Μόλις ολοκληρώσατε με την βοήθεια του docker-compose τη δημιουργία μιας απλής web εφαρμογής που αποτελείται από δύο μικροϋπηρεσίες.
- ▶ Με την ίδια ευκολία, ένας χρήστης μπορεί να περιγράψει σε ένα YAML αρχείο, με το όνομα docker-compose.yml, την παραμετροποίηση πολλαπλών docker containers που συνθέτουν μια εφαρμογή. Στη συνέχεια με μόνο μια εντολή κάνει «spin up» την εφαρμογή του.

docker

Conclusions

Docker Compose

Conclusions

- ▶ Σε αυτή τη διάλεξη και στα πλαίσια της Hands-On εξάσκησης, υλοποιήθηκαν πραγματικά παραδείγματα δημιουργίας απλών εφαρμογών με το εργαλείο Docker Compose.
- ▶ Ελπίζω μέσα από αυτά τα παραδείγματα που αναλύσαμε, να έγινε κατανοητή η χρήση του εργαλείου Docker Compose.
- ▶ Ευχαριστώ!

References

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