



AWS DEPLOYMENT USING EC2 AND DOCKER (TIES 4560 - TASK 5.1)

By, Md Ashrafuzzaman, Md Jahidul Islam Razin,
Mohammad Imranul Ferdoues, Md Raufur Rahman and Md
Mahadi Hasan



AGENDA

Introduction
3

AWS EC2 Setup
4

Docker Containerization
5-7

Application Deployment
8-10

Difficulties and Solutions
11

INTRODUCTION

- Deployment of applications (task2 and task4)
- AWS (EC2) used for scalable computing and Docker for containerization.
- The objective is to achieve a reliable and manageable cloud-based solution.

AWS EC2 SETUP

- EC2 instance type (t2.micro)
- Security groups and network configuration (launch-wizard-6)
- Public IP (54.173.190.7)
- SSH key (group7)
- Environment preparation (Linux)

DOCKER CONTAINERIZATION

- Image build for task2 (raufurrahman/soa_and_cloud:task2)

```
You, yesterday | 1 author (You)
### official base image
FROM python:3.10-alpine

## Set working directory
WORKDIR /usr/src/app

### Install system dependencies
RUN apk update && apk add --no-cache python3-dev py3-pip

### set environment variables
ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

### Install necessary dependencies
COPY ./requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

# Copy the application code
COPY . .

# Expose the application port
EXPOSE 5000

# Run Gunicorn
CMD ["gunicorn", "--workers", "3", "--bind", "0.0.0.0:5000", "manage:app"]
```

DOCKER CONTAINERIZATION

- Image build for task4 (raufurrahman/soa_and_cloud:task4)

```
You, yesterday | 1 author (You)
### official base image
FROM python:3.10-alpine

## Set working directory
WORKDIR /usr/src/app

### Install system dependencies
RUN apk update && apk add --no-cache python3-dev py3-pip

### set environment variables
ENV PYTHONDONTWRITEBYTECODE=1
ENV PYTHONUNBUFFERED=1

### Install necessary dependencies
COPY ./requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

# Copy the application code
COPY . .

# Expose the application port
EXPOSE 5000

# Run Gunicorn
CMD ["gunicorn", "--workers", "3", "--bind", "0.0.0.0:5000", "manage:app"]
```

DOCKER CONTAINERIZATION

- Created containers with docker-compose for task2 and task4

```
version: '3.8'

services:
  app:
    container_name: dropbox_file_manager
    image: raufurrahman/soa_and_cloud:task2
    restart: always
    ports:
      - "5000:5000"
    environment:
      - DROPBOX_APP_KEY=ijem33ekgmy6wap
      - DROPBOX_APP_SECRET=zh7sad8v7c54lg3
      - DROPBOX_REDIRECT_URI=https://dropbox-test.devopscub.net/auth/callback
    networks:
      - app-network
```

```
rest-client:
  container_name: rest_application
  image: raufurrahman/soa_and_cloud:task4
  restart: always
  ports:
    - "5001:5000"
  environment:
    - SECRET_KEY=Gr@up7
    - MONGO_SERVER_NAME=mongo
    - MONGO_USER_NAME=admin
    - MONGO_PASSWORD=shimanto
  depends_on:
    - mongo
  networks:
    - app-network

mongo:
  image: mongo:latest
  restart: always
  ports:
    - "27017:27017"
  environment:
    - MONGO_INITDB_ROOT_USERNAME=admin
    - MONGO_INITDB_ROOT_PASSWORD=shimanto
  volumes:
    - mongo-data:/data/db
  networks:
    - app-network

networks:
  app-network:
    driver: bridge

volumes:
  mongo-data:
```

APPLICATION DEPLOYMENT

Commands to run and debug

- `docker-compose up -d`
- `docker logs -f <container_id>`
- `docker exec -it <container_id> sh`
- `docker ps`

```
ubuntu@ip-172-31-43-207:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
91c29525b194	raufurrahman/soa_and_cloud:task1	"python app.py"	33 hours ago	Up 33 hours	0.0.0.0:5002->5000/tcp, [::]:5002->5000/tcp	country-info
04939d9612f7	raufurrahman/soa_and_cloud:task2	"unicorn --workers ..."	33 hours ago	Up 33 hours	0.0.0.0:5000->5000/tcp, :::5000->5000/tcp	dropbox_file_manager
620547c0cf19	raufurrahman/soa_and_cloud:task4	"unicorn --workers ..."	34 hours ago	Up 34 hours	0.0.0.0:5001->5000/tcp, [::]:5001->5000/tcp	rest_application
8f996f8ee4f8	mongo:latest	"docker-entrypoint.s..."	35 hours ago	Up 35 hours	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp	ubuntu_mongo_1

ISSUES/DIFFICULTIES AND SOLUTIONS

- Issues/Difficulties
 - Error in SSL configuration (no valid domain)
- Solutions
 - Valid Domain
 - Automatic SSL configuration with Caddy reverse proxy

```
:80 {  
    # Set this path to your site's directory.  
    root * /usr/share/caddy  
  
    # Enable the static file server.  
    file_server  
  
    # Another common task is to set up a reverse proxy:  
    # reverse_proxy localhost:8080  
  
    # Or serve a PHP site through php-fpm:  
    # php_fastcgi localhost:9000  
}  
  
# Refer to the Caddy docs for more information:  
# https://caddyserver.com/docs/caddyfile  
dropbox-test.devopsclub.net {  
    reverse_proxy localhost:5000  
}
```



THANK YOU 😊