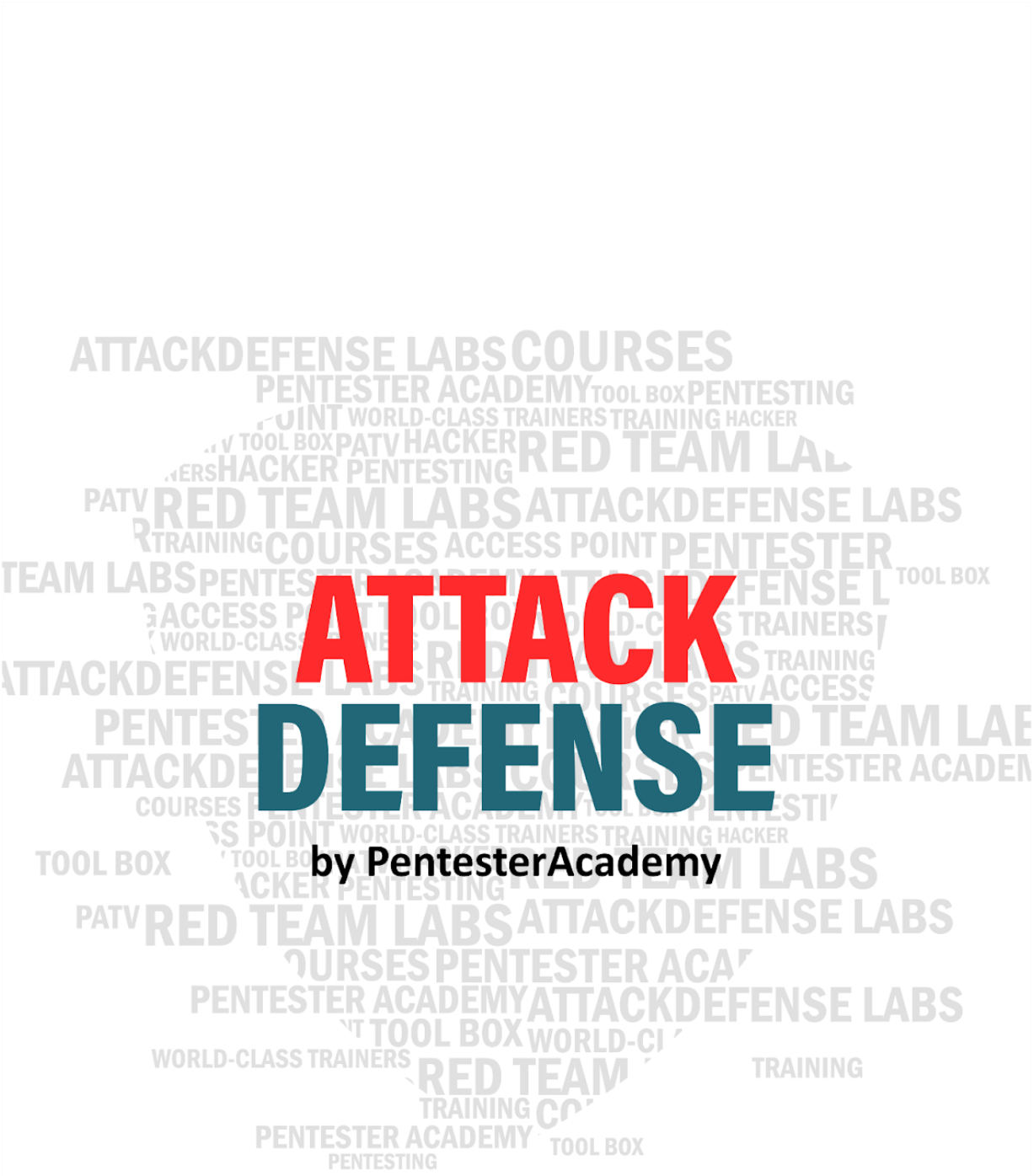

ATTACK
DEFENSE
by PentesterAcademy

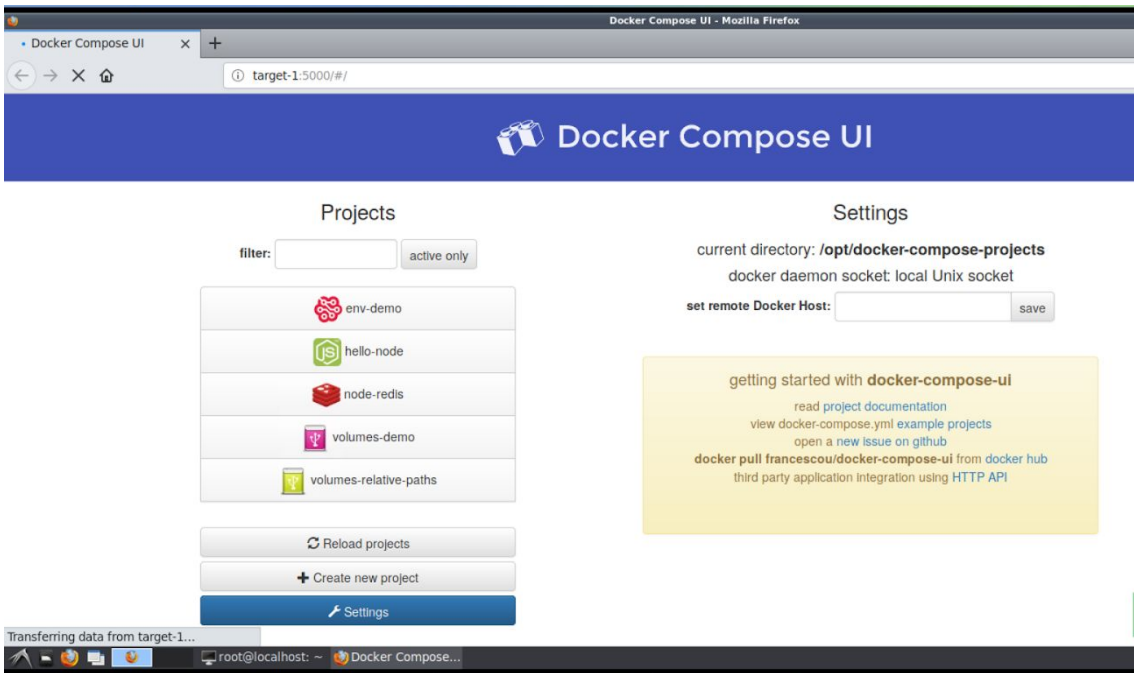
Name	Tool: Docker Compose UI
URL	https://attackdefense.com/challengedetails?cid=1418
Type	DevSecOps : Docker Tools

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Deploy "hello-node" project from Docker compose UI and check the content being served by the node-server!

Solution:

Step 1: Navigate to http://target-1:5000 to access Docker compose UI.



Step 2: SSH into target-1 machine to also check the activity of docker compose on the system.

Credentials
Username: root
Password: root

Command: ssh root@target-1:5000

```
root@attackdefense:~# ssh root@target-1
The authenticity of host 'target-1 (192.214.189.3)' can't be established.
ECDSA key fingerprint is SHA256:iLK0xAbC1+tuYXjMowYHxiRCY57WBF0dXLonlWGghuY.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'target-1,192.214.189.3' (ECDSA) to the list of known hosts.
root@target-1's password:
Welcome to Ubuntu 18.04 LTS (GNU/Linux 5.0.0-20-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Keen to learn Istio? It's included in the single-package MicroK8s.

     https://snapcraft.io/microk8s
Last login: Tue Nov 26 08:11:10 2019 from 192.214.189.1
root@localhost:~#
```

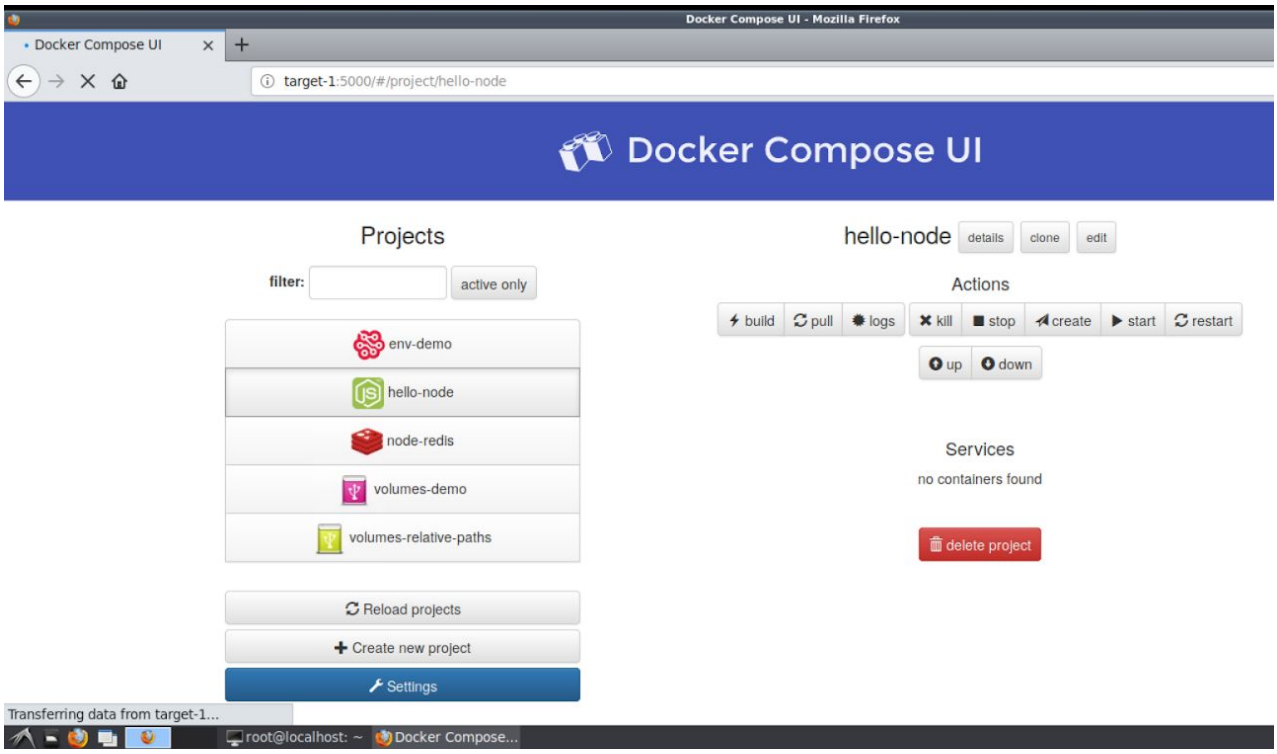
Step 3: Check running containers on the machine.

Command: docker ps

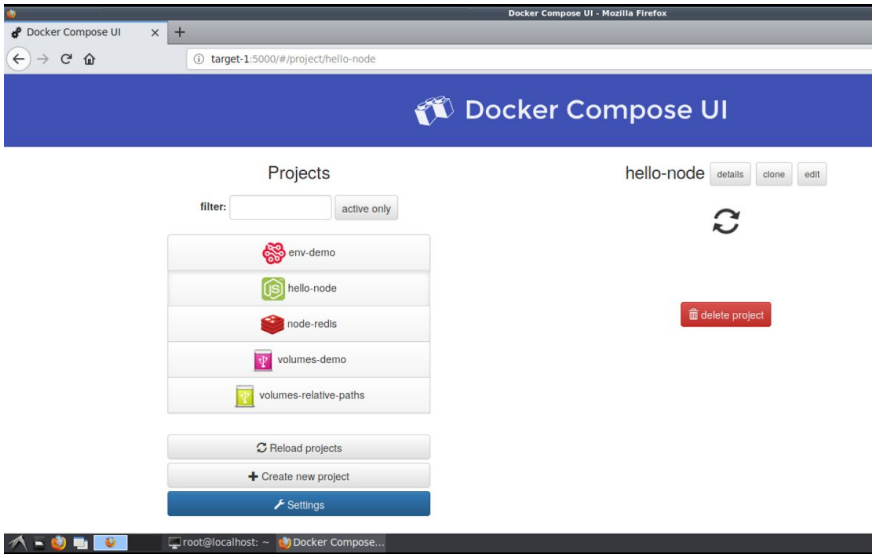
```
root@localhost:~#
root@localhost:~# docker ps
CONTAINER ID        IMAGE                                  COMMAND                  CREATED
    NAMES
7b4ca5056ab2        francescou/docker-compose-ui:1.13.0    "/env/bin/python /ap"   2 minutes ago
tcp   hopeful_kapitsa
root@localhost:~#
root@localhost:~#
root@localhost:~#
```

Only docker compose UI container is running on the system.

Step 4: Select “hello-node” project from web UI.



Step 4: Multiple operation options are listed on the project page. Build the project by pressing the “build” button.



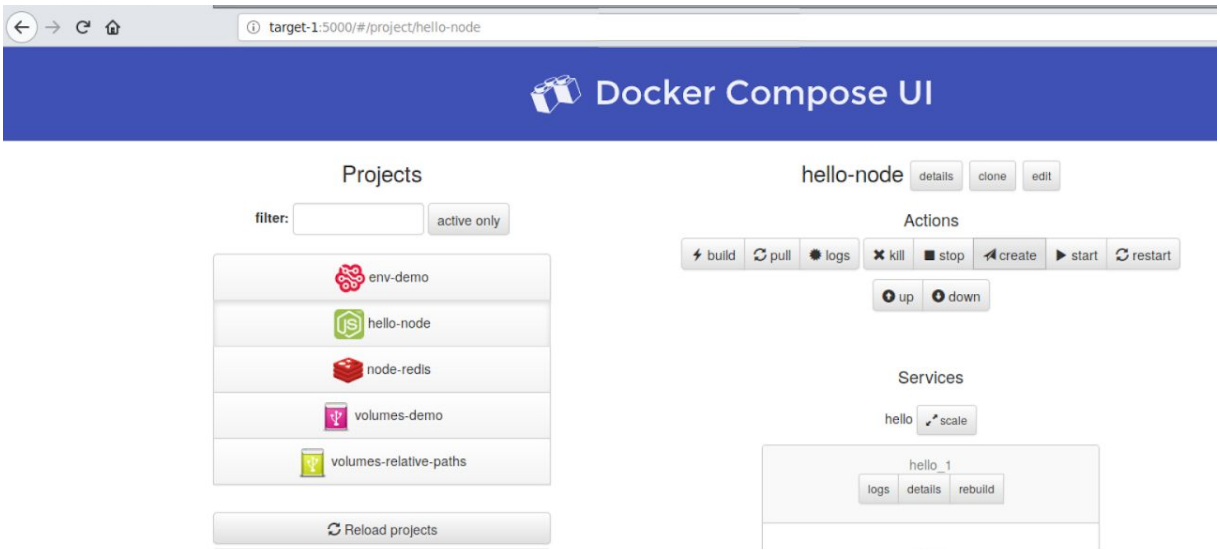
Using SSH session, check the logs of the running Docker Compose UI container.

Command: docker logs -f <container_id>

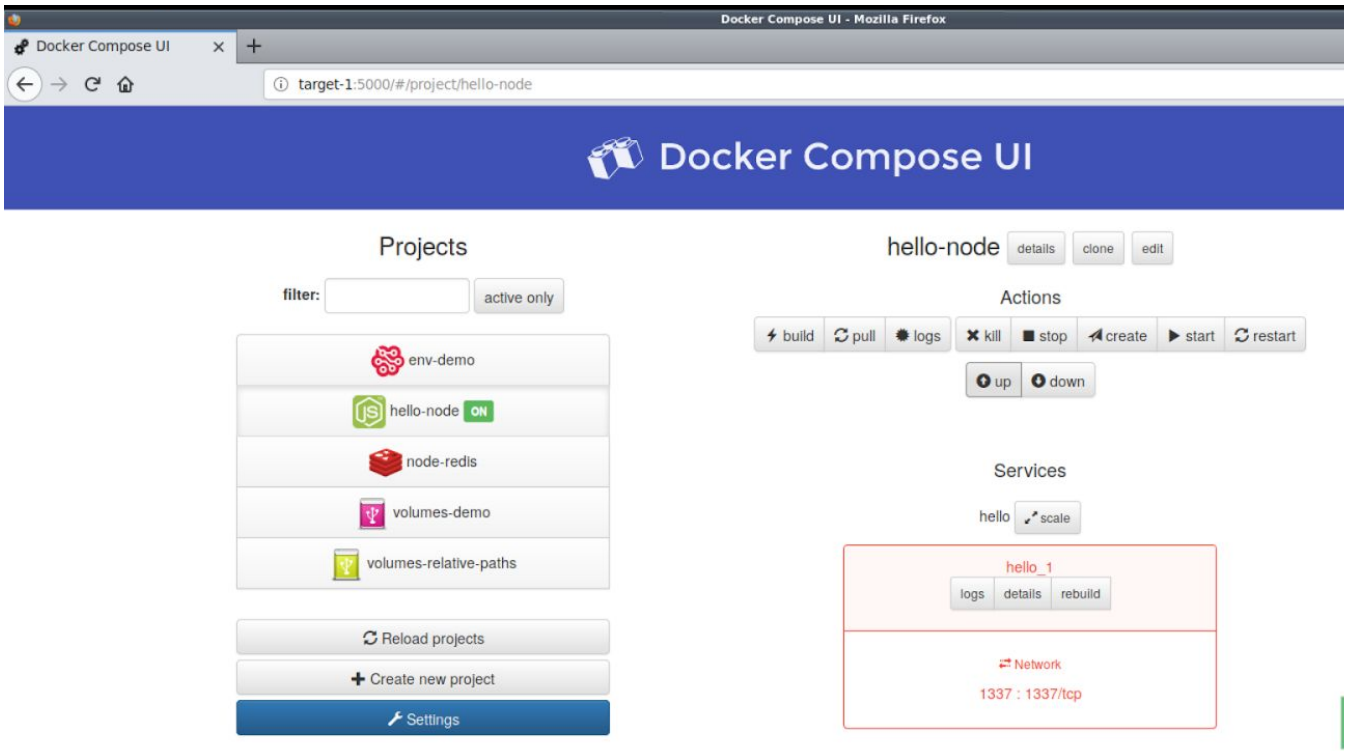
```
DEBUG:docker.api.build:Looking for auth config
DEBUG:docker.api.build:No auth config in memory - loading from filesystem
DEBUG:docker.utils.config:Trying paths: ['/root/.docker/config.json', '/root/.dockercfg']
DEBUG:docker.utils.config:No config file found
DEBUG:docker.api.build:No auth config found
DEBUG:urllib3.connectionpool:http://localhost:None "POST /v1.25/build?q=False&cachefrom=%5B%22
lse&rm=True HTTP/1.1" 200 None
Exception happened during processing of request from ('192.214.189.2', 36276)
----------------------------------------
Step 1/4 : FROM node:7-alpine
 ---> 4b72b56791f9
Step 2/4 : EXPOSE 1337
 ---> Running in b475a2a9a3e9
Removing intermediate container b475a2a9a3e9
 ---> 9eb633ea48f2
Step 3/4 : ADD ./hello.js /opt/
 ---> d99f216d4ac2
Step 4/4 : CMD ["node", "/opt/hello.js"]
 ---> Running in 61e6eaa318ea
Removing intermediate container 61e6eaa318ea
 ---> a84e34f2f8c7
Successfully built a84e34f2f8c7
Successfully tagged hellonode_hello:latest
INFO:werkzeug:192.214.189.2 - - [26/Nov/2019 08:16:18] "POST /api/v1/build HTTP/1.1" 200 -
```

One can observe that the image is being built in the backend.

Step 5: Create the project using “create” button.

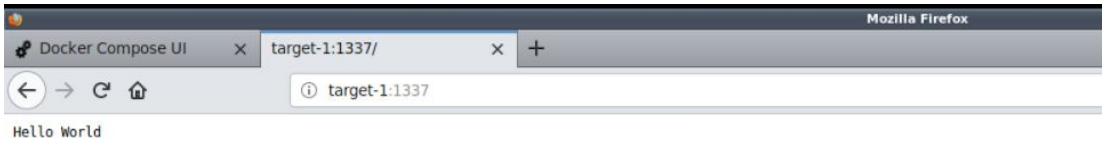


Step 5: Bring the project up using “up” button. This will essentially run the setup.



The hello-node is deployed and serving on TCP port 1337 of the machine.

Step 6: Navigate to http://target-1:1337 to check the content being served by the hello-node server.



The content is “Hello World”

References:

1. Docker (https://www.docker.com/)
2. Docker Compose UI (https://github.com/francescou/docker-compose-ui)