

# KPMG Microservices Training

## Docker Nginx Exercise Solution

```
docker pull alpine:latest
docker images
docker run -it --rm alpine:latest /bin/sh
```

Use Linux commands to explore the container.

Exit the container by typing control-D.

## Docker Nginx Solution

```
docker pull nginx:latest
docker images
docker run -d --name nginx -p 8080:80 nginx:latest
```

Access the website using a browser `http://localhost:8080` or using `curl`.

```
curl http://localhost:8080
```

Start a shell in the current container. There is no editor in the container, so install vim. Edit the index page and make changes. Close the shell.

```
docker exec -it nginx /bin/sh
apt update && apt install -y vim
vi /usr/share/nginx/html/index.html
control-D
```

Access the website using a browser `http://localhost:8080` or using `curl`.

```
curl http://localhost:8080
```

You should see the changed page.

Save the container.

```
docker commit nginx nginx:v1
```

Delete the container.

```
docker rm -f nginx
```

Start a container from the new image.

```
docker run -d --name nginx -p 8080:8080 nginx:v1
curl http://localhost:8080
docker exec -it nginx /bin/sh
vi /usr/share/nginx/html/index.html
control-D
docker rm -f nginx
```

See that `vi` is installed and the changes made to `index.html` are preserved in the new image.

## Docker Flask Exercise Solution

Go to the `KPMG/Flask` directory and look at the files.

- `Server.py` implements a Flask Web Server.
- The `static` directory contains static content files.
- `Dockerfile` is the Docker build file.

Build a Docker image.

```
docker build -t pyserver:latest .
```

Run a container.

```
docker run -d --name pyserver -p 8080:8080 pyserver:latest
```

View the Web page at `http://localhost:8080`.

Delete the container.

```
docker rm -f pyserver
```

## Reverse Proxy Exercise Solution

Go to the `KPMG/Nginx` directory.

Look at the Nginx configuration file `default.conf` and the `Dockerfile`.

Build a Docker image.

```
docker build -t nginx:proxy .
```

Go to the `KPMG` directory.

Look at the file `compose.yml`.

Run the containers.

```
docker compose up -d
```

Access the Flask server via the Nginx proxy using the URL `http://localhost:8080/flask/static/index.html`.

What does the URL `http://localhost:8080/flask/` return and why?