

## Module-3: Docker – I Assignment - 2

You have been asked to:

- Save the image created in Assignment 1 as a Docker image
- Launch container from this new image and map the port to 81
- Go inside the container and start the apache2 service
- Check if you are able to access it on the browser

### Solution:

- 1) Save the image created in Assignment 1 as a Docker image:  
List running containers to find the container ID or name: docker ps

```
ubuntu@ip-172-31-15-134:~$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS                               NAMES
acc29f734773   ubuntu    "/bin/bash"             20 minutes ago Up 20 minutes  0.0.0.0:80->80/tcp, :::80->80/tcp  cool_dubinsky
ubuntu@ip-172-31-15-134:~$
```

Commit the container to a new image: docker commit [container\_id\_or\_name] my-apache-image

```
ubuntu@ip-172-31-15-134:~$ docker commit acc29f734773 my-apache-image
sha256:fbf310460a98d88d7836df85cd615aab89160b4c2e345352b5dea4287e0460be
ubuntu@ip-172-31-15-134:~$
```

- 2) Launch container from this new image and map the port to 81:

Run the new container and map port 81: docker run -dit -p 81:80 --name my-apache-container my-apache-image

-p 81:80 maps port 81 on your local machine to port 80 in the container.

--name my-apache-container names the container.

my-apache-image is the name of the image created in the previous step.

```
ubuntu@ip-172-31-15-134:~$ docker run -dit -p 81:80 --name my-apache-container my-apache-image
7de3728bc482d792c70099e450b0d13ec7325a666b584eb4d31f6740e2c5f084
ubuntu@ip-172-31-15-134:~$
```

- 3) Go inside the container and start the apache2 service:

Access the running container: docker exec -it my-apache-container bash

```
ubuntu@ip-172-31-15-134:~$ docker exec -it my-apache-container bash
root@7de3728bc482:/#
```

**Start Apache2 service:** service apache2 start

```
root@7de3728bc482:/# service apache2 start
 * Starting Apache httpd web server apache2
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
 *
root@7de3728bc482:/#
```

4) **Check if Apache2 is Accessible in the Browser:** Open your web browser.

Navigate to: <http://localhost:81> or <http://<your-server-ip>:81>.



← → × ⓘ 35.154.37.94:81



## Apache2 Default Page

# Ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/`