Lesson 3 Demo 4: Configure a Registry

This section will guide you to:

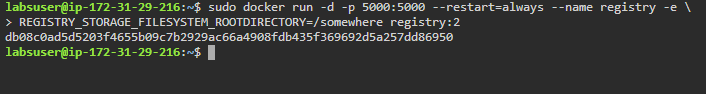
* Configure a registry in Docker

**Step 1:** Create an environment variable named **REGISTRY\_variable** where **variable** is the name of the configuration option and **\_** represents indentation level

* Use the following command to override the root directory value by specifying a configuration variable from the environment using the **-e** argument:

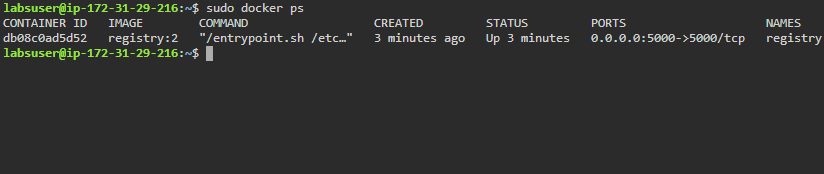
*sudo docker run -d -p 5000:5000 --restart=always --name registry -e \*

*REGISTRY\_STORAGE\_FILESYSTEM\_ROOTDIRECTORY=/somewhere registry:2*



* List the running containers to check the newly created registry with new Storage filesystem root directory

*sudo docker ps*

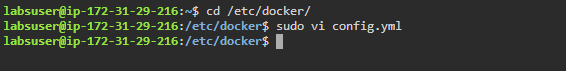


**Step 2:** Override the entire configuration file by creating a new file named **config.yml**

* Navigate to **/etc/docker/** folder and create a **config.yml** file

*cd /etc/docker*

*sudo vi config.yml*

****

* Add the following code in the **config.yml** file

*version: 0.1*

*log:*

*fields:*

*service: registry*

*storage:*

*cache:*

*blobdescriptor: inmemory*

*filesystem:*

*rootdirectory: /var/lib/registry*

*http:*

*addr: :5000*

*headers:*

*X-Content-Type-Options: [nosniff]*

*auth:*

*htpasswd:*

*realm: basic-realm*

*path: /etc/registry*

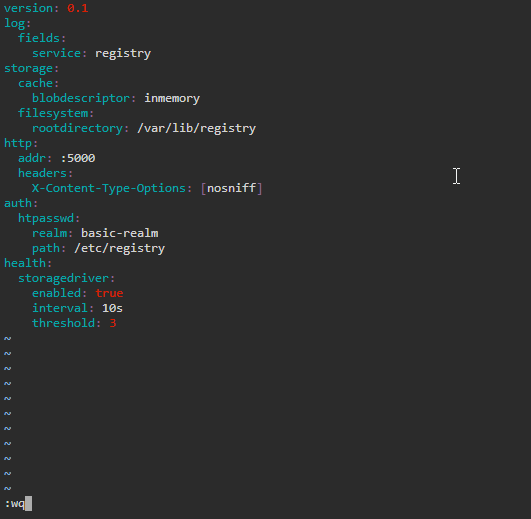
*health:*

*storagedriver:*

*enabled: true*

*interval: 10s*

*threshold: 3*

****

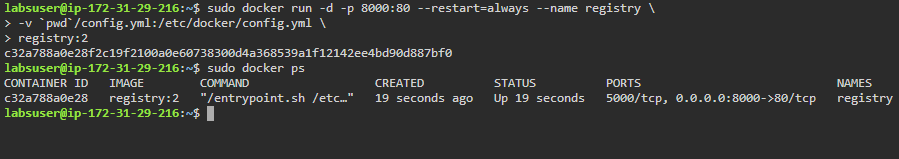
**Note:** After writing the above mentioned code, press the ESC button and enter **:wq** to save the file and exit from the editor.

* Use the following command to run a registry container with a new config.yml file specified in it:

*sudo docker run -d -p 8000:80 --restart=always --name registry2 \*

*-v `pwd`/config.yml:/etc/docker/config.yml \*

*registry:2*



**Note:** You can use *the sudo docker ps* command to check the running containers.