Lesson 5 Demo 11: Convert an Application Deployment into a Stack

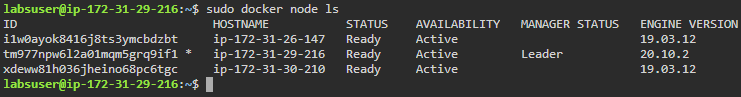
This section will guide you to:

* Convert an application deployment into a stack using a file named *docker-compose.yml*

**Step 1:** Drain the worker nodes in the swarm cluster to make sure the registry service runs on the manager node

* List all the nodes present in the swarm cluster and ensure that all nodes are in **Active** state

*sudo docker node ls*

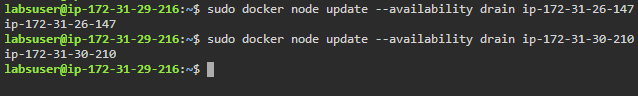


**Note:** Copy the HOSTNAME of worker nodes

* Use the following command to drain the worker nodes:

*sudo docker node update --availability drain hostname\_Worker\_Node*

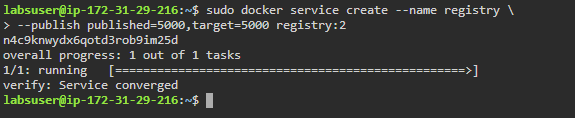
***Note:*** *Replace hostname\_Worker\_Node with the HOSTNAME copied in previous step*



**Step 2:** Start the registry as a service on your swarm

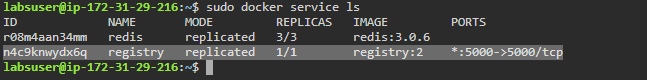
*sudo docker service create --name registry \*

*--publish published=5000,target=5000 registry:2*



**Step 3:** List the running services to check the status of *registry* service

*sudo docker service ls*



**Step 4:** Check if *registry* service is working with *curl*

*curl http://localhost:5000/v2/*

****

**Step 5:** Create a directory for the project

*mkdir stackdemo*

*cd stackdemo*



**Step 6:** Create a file called *app.py* in the **stackdemo** directory

* Use the following command to create a projectfile:

*nano app.py*

**

* Add the following code in the *app.py* file:

*from flask import Flask*

*from redis import Redis*

*app = Flask(\_\_name\_\_)*

*redis = Redis(host='redis', port=6379)*

*@app.route('/')*

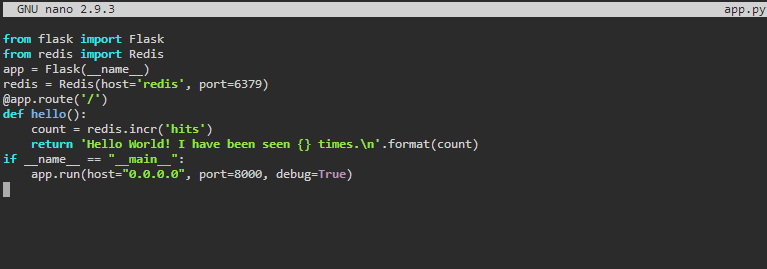
*def hello():*

*count = redis.incr('hits')*

*return 'Hello World! I have been seen {} times.\n'.format(count)*

*if \_\_name\_\_ == "\_\_main\_\_":*

*app.run(host="0.0.0.0", port=8000, debug=True)*



**Note**: Press **Ctrl+X** to exit the editor. Then type **Y** and press **Enter** to save the file.

**Step 7:** Create a file called *requirements.txt*

* Use the following command to create and open *requirements.txt*:

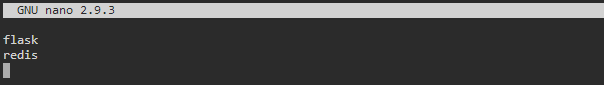
*nano requirements.txt*

**

* Add the following text in the *requirements.txt* file:

*flask*

*redis*



**Note**: Press **Ctrl+X** to exit the editor. Then type **Y** and press **Enter** to save the file.

**Step 8:** Create a file called Dockerfile

* Use the following command to create a *Dockerfile*:

*nano Dockerfile*



* Add the following code in the *Dockerfile*:

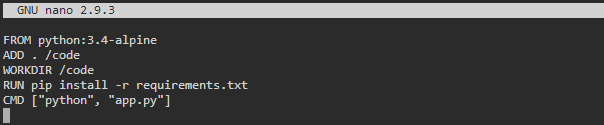
*FROM python:3.4-alpine*

*ADD . /code*

*WORKDIR /code*

*RUN pip install -r requirements.txt*

*CMD ["python", "app.py"]*



**Note**: Press **Ctrl+X** to exit the editor. Then type **Y** and press **Enter** to save the file.

**Step 9:** Create a file named docker-compose.yml

* Use the following command to create the *docker-compose.yml* file:

*nano docker-compose.yml*



* Add the following code in the *docker-compose.yml* file:

*version: "3.3"*

*services:*

*web:*

*image: 127.0.0.1:5000/stackdemo*

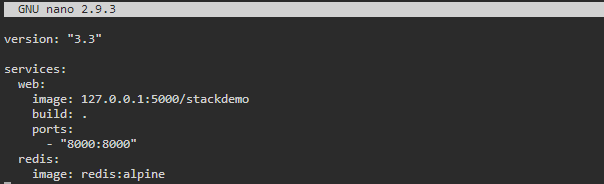
*build: .*

*ports:*

*- "8000:8000"*

*redis:*

*image: redis:alpine*



**Note**: Press **Ctrl+X** to exit the editor. Then type **Y** and press **Enter** to save the file.

**Step 10:** Start the application

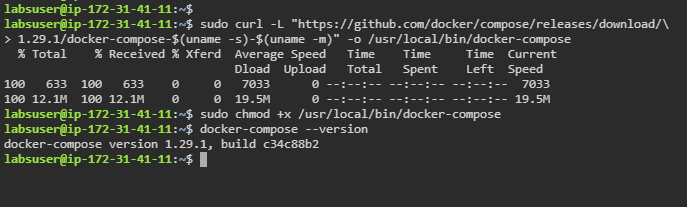
* Use the following commands to install *docker-compose*:

*sudo curl -L "https://github.com/docker/compose/releases/download/\*

*1.29.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose*

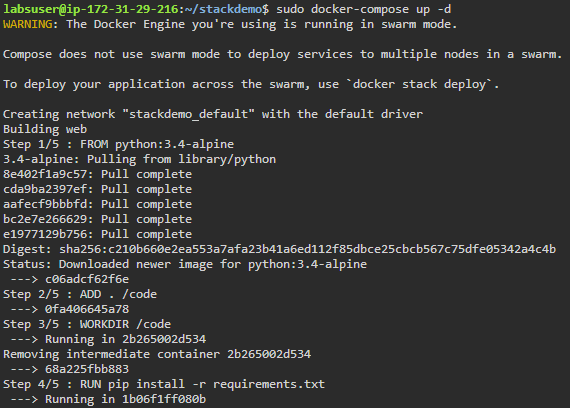
*sudo chmod +x /usr/local/bin/docker-compose*

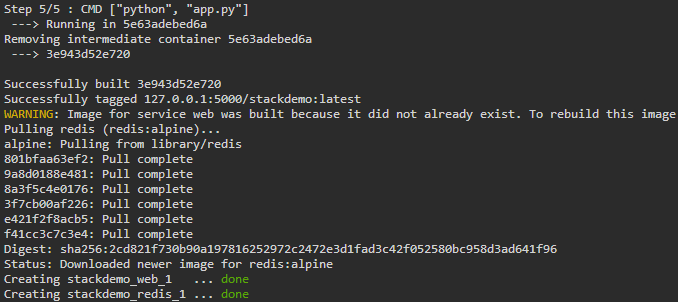
*docker-compose --version*

**

* Start *docker-compose* using the following command:

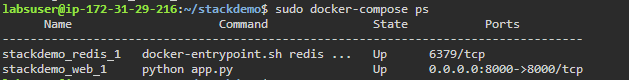
*sudo docker-compose up -d*



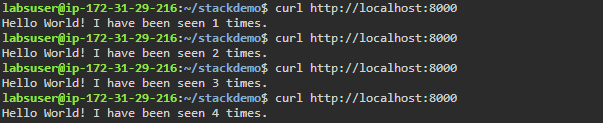


**Step 11:** Use the following commands to check whether the app is running

*sudo docker-compose ps*

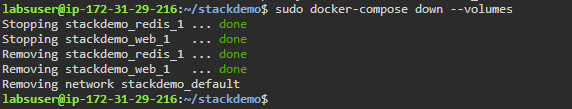


*curl* [*http://localhost:8000*](http://localhost:8000)



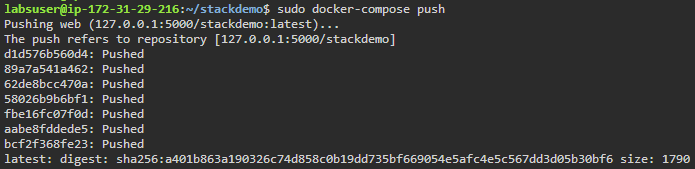
**Step 12:** Bring the application down

*sudo docker-compose down --volumes*



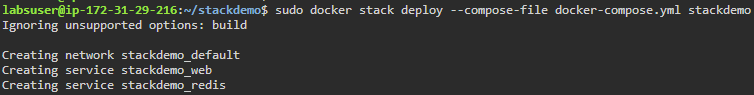
**Step 13:** Push the application to the registry

*sudo docker-compose push*



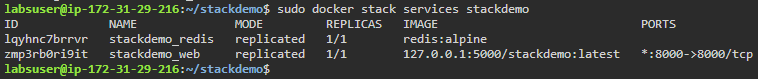
**Step 14:** Use the following command to create the stack docker stack deploy:

*sudo docker stack deploy --compose-file docker-compose.yml stackdemo*



**Step 15:** Check if the stack is running

*sudo docker stack services stackdemo*

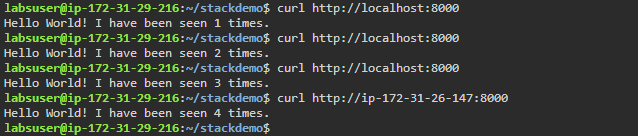


**Step 16:** Test the app again with *curl*command

*curl* [*http://localhost:8000*](http://localhost:8000)

*curl* [*http://ip-172-31-26-147:8000*](http://ip-172-31-26-147:8000)

***Note:***In step 10 while starting docker-compose if you get an error showing the port is already assigned, run the command **sudo docker ps** and kill the container with the same port and then proceed.



**Step 17:** Use the following command to bring the stack down:

*sudo docker stack rm stackdemo*

