

Assessments

Level 0

1. Setup 'NGINX' server using docker.

- NGINX is a popular lightweight web application that is used for developing server-side applications.
- Use the official nginx image from docker hub registry.
- Pull image and run the docker container on docker host.
- Open then 8080 port.
- Try accessing [http://dockerhost ip:8080](http://dockerhost_ip:8080) and check installation.

2. Setup simple hello world website using 'NGINX' server using docker.

- NGINX is a popular lightweight web application that is used for developing server-side applications.
- Use the nginx server to host your simple 'hello world' website.
- Create simple html page. Say an 'Index.html' page.
- Pull image and run the docker container on docker host with correct commands..
- Open then 8080 port.
- Try accessing [http://dockerhost ip:8080/index.html](http://dockerhost_ip:8080/index.html) and check the welcome message.

3. Setup 'MongoDB' using docker.

- MongoDB is a famous document-oriented database that is used by many modern-day web applications.
- Use the official image from docker hub registry.
- Pull image and run the docker container on docker host.
- Open then required port while running container.
- Try connecting to mongodb using desktop clients to verify.

Level 1

1. Create your own dockerfile to install Python.

- Create your own dockerfile to install python
- Expose required ports.
- Verify installation.
- Publish your image to docker hub registry.

2. Create your own dockerfile to setup Elasticsearch.

- Create your own dockerfile to setup single node ES cluster.
- Expose required ports.
- Install ES plugins.
- Verify installation by accessing through web browser.
- Publish your image to docker hub registry.

Level 2

1. Use Docker-Compose to setup multi container ELK stack.

- Create your own docker compose file to setup Elasticsearch, logstash and Kibana (ELK) stack.
- Expose required ports as needed.
- Create the stack and test it.

References:

<https://hub.docker.com/explore/>