As discussed with the Professor, I changed the scope of Docker and Containers with the following course

1. Docker Quick Start – Total Time 02:27:01 - https://linuxacademy.com/cp/modules/view/id/86
2. Docker Certified Associate Prep Course – Total Time 18:30:53 - <https://linuxacademy.com/cp/modules/view/id/150>

**Docker Quick Start**

**Lecture: Running Containers**

1. To get a list of all running containers:

docker ps

1. To get a list of all past running containers

docker ps -a

1. To get a list of base images

docker images

1. To run a container in interactive mode

docker run -it centos:latest /bin/bash

Here you are logged in as a user in the container and working inside the container. You can exit with exit command.

1. To run the container without being connected to it

docker run -d centos:latest /bin/bash

Here it executed /bin/bash and exited.

1. To inspect a running docker container

docker inspect <name> or <container id>

1. To name your container image

docker run -d --name =<Your name> nginx:latest

1. To stop the container

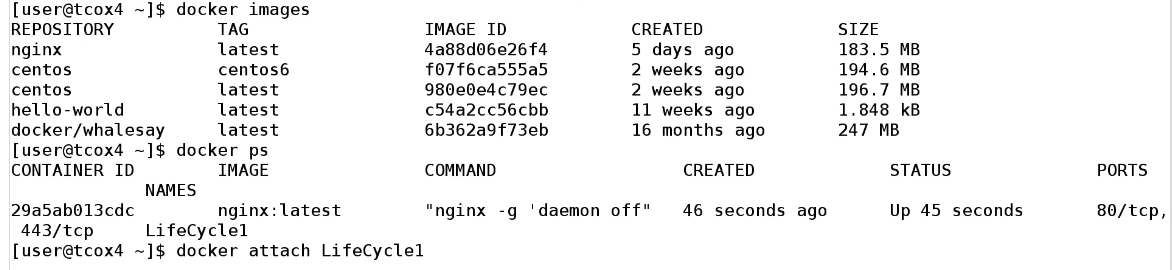
docker stop <name> or <container id>

Total Time: 26 minutes

**Lecture: The Container Lifecycle**

1. To attach to the running container

docker attach <name> or <container id>



Attached with the process. If exited then container also exits

1. To restart / start the previous running container

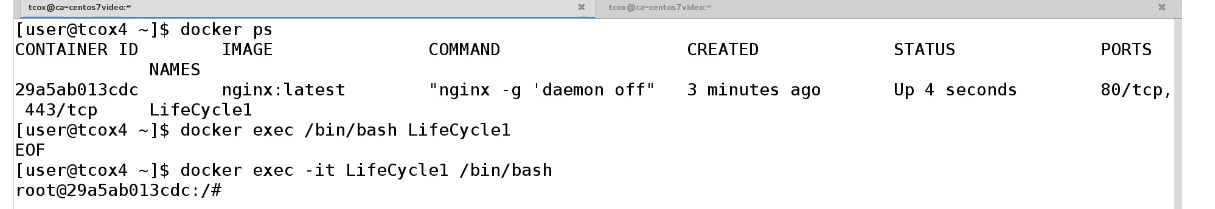
docker start <Name> or <container id>

docker restart <Name> or <container id>



1. To execute in a running container without affecting command or the process

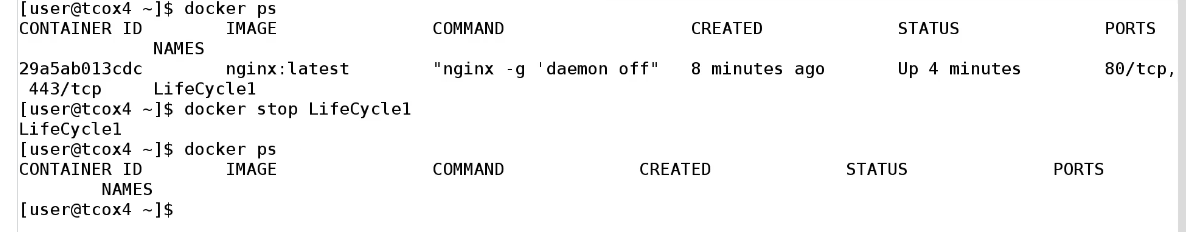
docker exec -it <name> or <id> <command>



However since I am not attached here, if I exist here, the container will not stop

1. To stop the container

docker stop <name> or <id>



Total Time: 15 minutes

**Lecture: Image and Container Management**

1. To delete the container image

docker rmi <repository:tag> or <image id>



However, it cant remove image previously run. To remove first remove containers that are independacy based on that image.

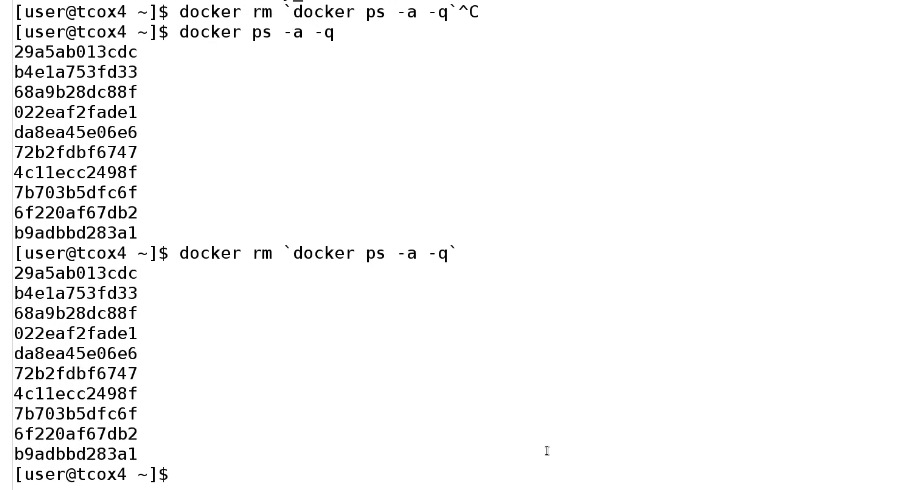
docker rm <container id> or <name>



However, all container with reference must be removed and not just the latest one.

1. To remove all containers at a time

docker rm ‘docker ps -a -q’



Now there will be no containers with dependency to images

1. To forcefully remove the image

docker rmi -f <repositoryname> or <repository name: tag> or <image id>



However, this wont remove the container (snapshot of the image) and you can still start the container. You should remove container if not needed since it consumes space.

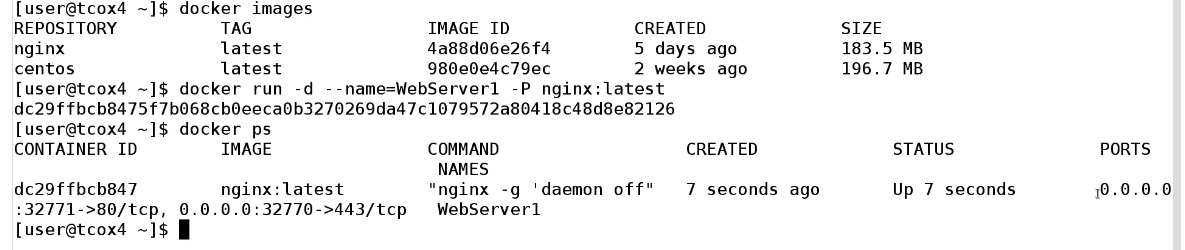
Total Time: 15 minutes

**Lecture: Redirection - Ports and Volumes**

We need to redirect container ports and volumes to underlying host

1. To redirect any ports that are exposed in the container to make available through the host on random port between 32768 and 65000 which is port range available to docker

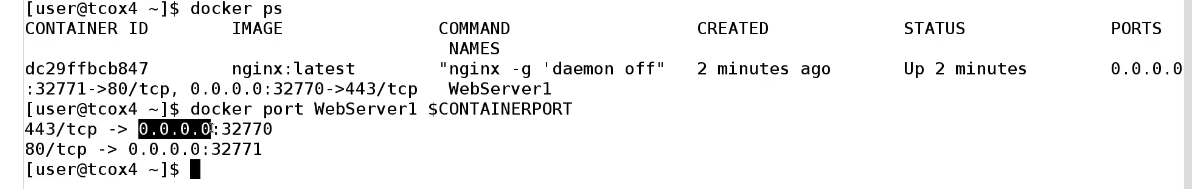
docker run -d --name = WebServer1 -P nginx:latest



32771 localhost = container 80 and 32770 localhost = container 443

1. To list ports of container

docker port Webserver1 $CONTAINERPORT



1. To bind the port to specific port rather than random port

docker run -d -p <host port>:<container port>,<host port>:<container port> --name=WebServer2 nginx:latest



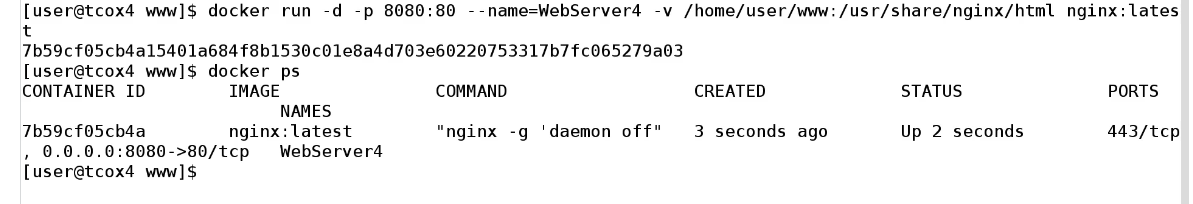
1. To mount directories from underlying host inside the container

docker run -d -p 8080:80 --name=Webserver3 -v /mnt/data nginx:latest



1. To mount directories from underlying host to particular folder inside the container

docker run -d -p 8080:80 --name=Webserver3 -v /mnt/data:/usr/share/nginx/html nginx:latest

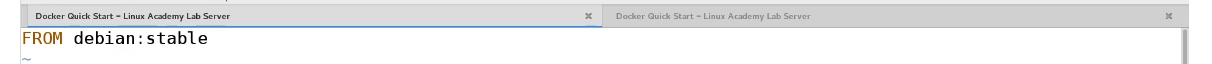
to create your own base image

Total Time: 25 minutes

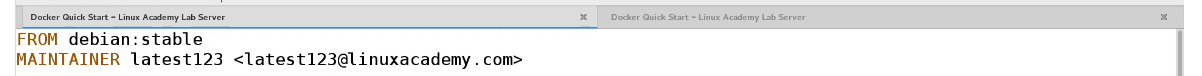
**Lecture: The Dockerfile**

We need to learn how to create your own base images

1. First element of the docker file is where is the image coming from like a base image, tar gz or a public repository?



1. Then list who is the maintainer of this file . The docker username , email address

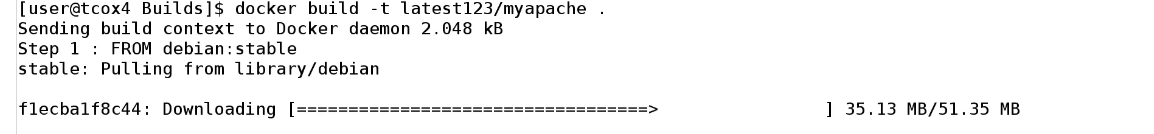


1. Next is the list of commands you want to run like the RUN command

RUN apt-get update

1. Before running, you need to build the image

docker build -t <name of the image> <location of the docker file>

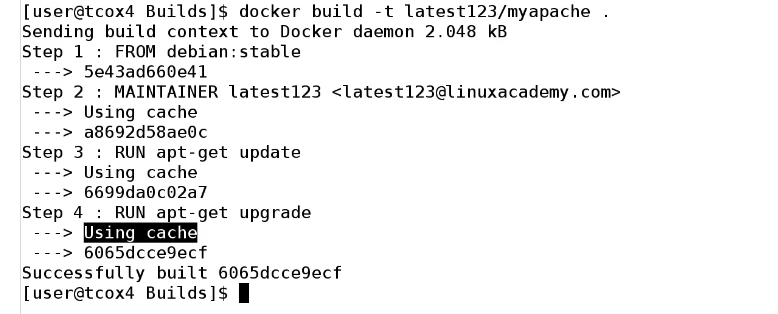


However if there is an error, then it wont build

1. Since each line consumes memory as it creates new layers, the best practice will be to combine them

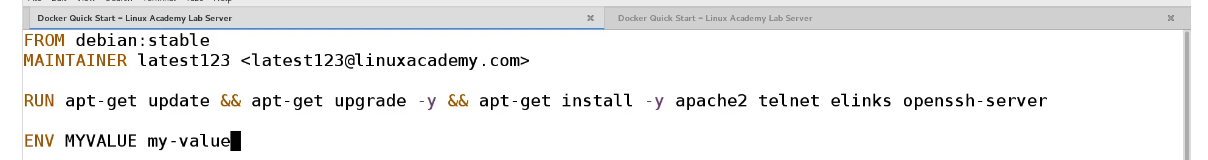
RUN apt-get update && apt-get upgrade -y && apt-get install -y apache2 telnet elinks openssh-server

1. On rerun, the docker file will use cache to rebuild itself.



1. To add an environment variable

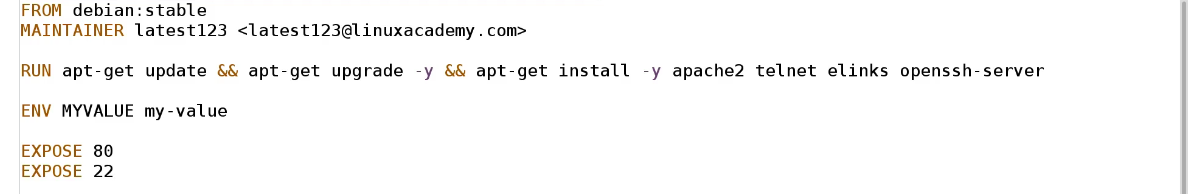
ENV MYVALUE <value>



1. To expose port,

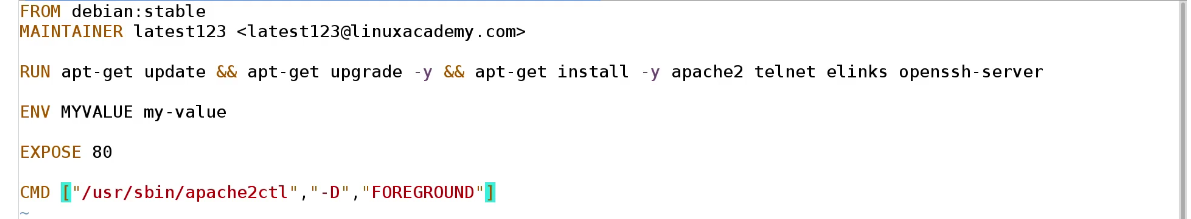
EXPOSE <port number>

EXPOSE 80



1. To run any command when the container is instantiated

CMD <list of command as a list>



Total Time: 30 minutes

**Lecture: Containers vs. Services in Docker Swarm**

Containers has given us the flexibility, portability, granularity, and abstraction to get the most out of our environments and deployments. However, it has quickly becoming ‘too limited’. There is a need to dploy complex configurations in Highly available and scalable implementations. This can be done through Orchestration or cluster management and control software (like Docker Swarm or Kubernetes) to work directly with Docker containers.

‘Docker service’ is the solution to managing containers deployed in highly available, easily scalable cluster implementations. Whereas containers are limited to the single host they are started on, services are containers that live on a scalable number of ‘workers’ in a cluster of systems. Docker Swarm handles access to, and the availability of, that service across those worker nodes, eliminating the challenges of routing and accessing individual containers. Scalability is key in the enterprise, both up and down, in order to maximize your infrastructure spend. The scalability services allow you to take granular control of your CPU, Memory, Disk, Network, and more.



While containers require manual steps to expose functionality outside of the host system (ports, network and/or volumes), functionality of the services is easily accessed using features like ‘routing mesh’ outside the worker nodes. Containers need more complex configurations to use multiple instances example via proxies. Services handles this through single sets of commands.

Total Time: 20 Minutes

**Content: Docker Cheat Sheet**

One of the main takeaways of this Linuxacademy course was the Docker – Cheat Sheet which provides the following troubleshooting:

1. Docker Engine Installation
2. Pull an Image from a Registry (Using Docker Pull and Docker Images)
3. Utilize Search in a Registry
4. Use CLI Commands to Manage Images (List, Delete, Prune, RMI, etc)
5. Inspect Images and Report Specific Attributes Using Filter and Format
6. Container Basics - Running, Attaching to and Executing Commands In Containers
7. Create an Image with a Dockerfile
8. Dockerfile Options, Structure, and Efficiencies (Part I and Part II)
9. Describe and Display How Image Layers Work
10. Flatten an Image to a Single Storage Layer
11. Selecting a Docker Storage Driver
12. Prepare for a Secure Docker Registry
13. Deploy, Configure, Log In to, Push, Pull, and Delete an Image in a Registry
14. Managing Images in Your Private Repository
15. Configure Logging Drivers (Splunk, Journald, etc)
16. Complete Setup of a Swarm Mode Cluster with Managers and Worker Nodes
17. State the Differences Between Running a Container and Running a Service
18. Demonstrate Steps to Lock (and Unlock) a Cluster
19. Extend the Instructions to Run Individual Containers into Running Services Under Swarm and Manipulate a Running Stack of Services
20. Illustrate Running a Replicated vs Global Service
21. Increase and Decrease the Number of Replicas in a Service
22. Setting Up a Swarm (Backup and Restore)
23. Demonstrate the Usage of Templates with “docker service create”
24. Apply Node Labels to Demonstrate Placement of Tasks
25. Convert an Application Deployment into a Stack File Using a YAML Compose File with ‘docker stack deploy’
26. Create a Docker Bridge Network for a Developer to Use for their Containers
27. How to Configure Docker to Use External DNS
28. Publish a Port So That an Application is Accessible Externally and Identify the Port and IP It Is On
29. Deploy a Service on a Docker Overlay Network
30. Troubleshoot Container and Engine Logs to Understand Connectivity Issues Between Containers
31. Describe How Storage Can Be Used Across Cluster Nodes
32. Describe the Process of Signing an Image
33. Set Up and Configure Universal Controller Plane and Docker Trusted Registry for Secure Cluster Management
34. Complete Configuration of Backups for UCP and DTR
35. Create and Manage Users and Teams
36. Configure RBAC in UCP and Enable LDAP for Authentication in UCP

This cheat sheet is highly recommended for Beginners in Docker pursuing career in Docker.

Total Time: 5 minutes

In content available for extra reading, Linuxacademy course provides following resources:

1. About the Exam - Everything you need to know about the exam.
2. Container Network Model - Slides discussing the Container Network Model from the course lesson.
3. Containers Vs. Services - Slides from the Containers vs. Service course lesson.
4. Docker Engine and Docker Swarm Default Security - Slides from the Docker Engine and Docker Swarm Default Security lesson.
5. Introduction to Docker Enterprise and Docker Swarm - Slides from the Introduction lesson.
6. Docker Engine, UCP and DTR Communication - Slides from the Docker Engine, UCP and DTR Communication lesson.
7. Dockerized Application Communication - Slides from the Dockerized Application Communication course lesson.
8. Docker Network Driver Use Cases - Slides from the Docker Network Driver Use Cases
9. Docker Storage Layers - Slides from the Docker Storage Layer course lesson.
10. Docker Storage Drivers and Use Cases - Slides from the Docker Storage Drivers (Graph Driver) and Use Cases course lesson.
11. Importance of Quorum in a Swarm Cluster - Slides from the Importance of Quorum course lesson.
12. MTLS - Slides from the MTLS course lesson.
13. Docker Namespaces and Control Groups - Slides from the Namespaces and Control Groups course lesson.
14. Docker Discussion on Sizing Requirements for Services and Containers - Slides from the Sizing course lesson.
15. Docker Troubleshooting - Slides from the Troubleshooting lesson.
16. Docker Cheat Sheet -
17. Docker File - Pieces, Parts and Optimization - The vertical Dockerfile pieces and directives from our demonstration videos.

Went through some of the contents. However, it is beyond the scope of Beginner Level