**Docker compose**

* This is tool of docker which is used for executing multiple docker commands from one point of control.
* This file is created using yamil.
* Yamil always takes the data in the format of key andvalues.

**---**

**intelliq:**

**- traniers:**

**- sai:**

**- devops: 12000**

**- selenium: 5000**

**- sheshi:**

**- devops: 12000**

**- aws: free**

**- receptionist**

**- shailaja**

**...**

To validated our yamil syntax <http://www.yamllint.com>

* To install docker compose on Linux
* <https://docs.docker.com/compose/install/#install-compose>’
* sudo curl -L https://github.com/docker/compose/releases/download/1.16.1/docker-compose-`uname -s`-`uname -m` -o /usr/local/bin/docker-compose
* sudo chmod +x /usr/local/bin/docker-compose
* Docker-compose –version

21/09/2017

**Use case**

**1) Create a docker compose file for starting a word press container and mysql container**

* Create a file called docker-compose.yml
* # vim docker-compose.yml
* Go into insertion mode by pressing “I”

Version: ‘3’

Services:

Mysql:

Image: mysql

Environment:

MYSQL\_ROOT\_PASSWORD: mypassword

Wordpress:

Images: wordpress

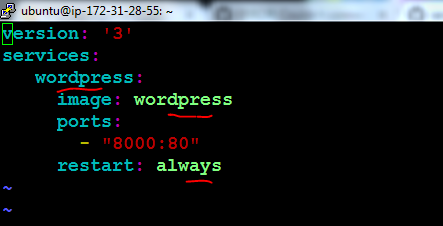
Ports:

- 8080:80

* **Save and quit**
* **Escape : wq enter**
* **To run the above file**
* **# docker-compose up**
* **To stop the services using our compose file**
* **# docker-compose down**
* **# docker –compose –f filename up**

1. **Installing wordpress services using docker-compose**

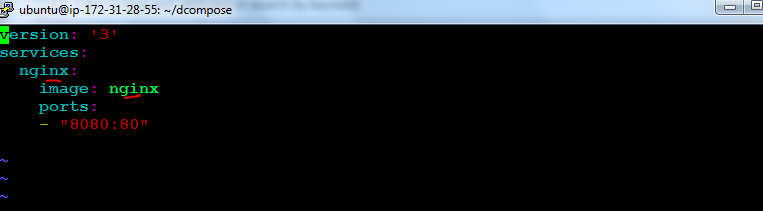
**Sudo vim docker-compose.yml**

****

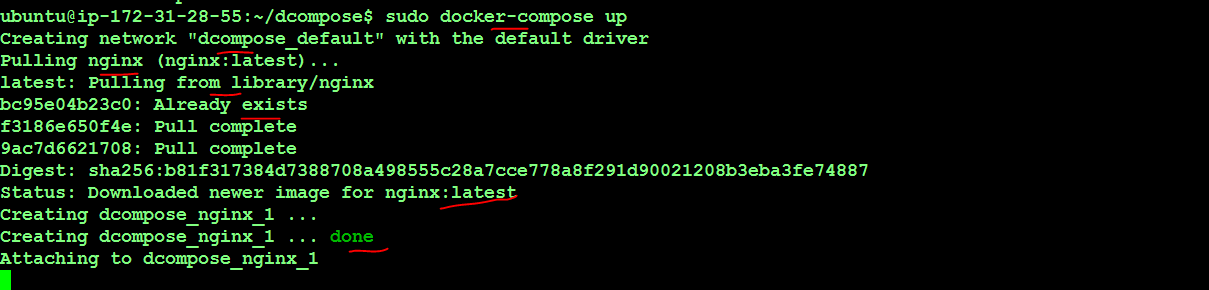
**Sudo docker-compose up**

1. **Installing nginx services using docker-compose**

**Sudo vim docker-compose1.yml**

****

**Sudo docker-compose up**

****

**Building docker images using docker file**

Docker file is a simple test file which uses specific commands using which it is possible to create our own images.

It uses the following keywords to create or modify images

* + FROM
  + MAINTAINER
  + CMD
  + ENTERYPOINT
  + RUN
  + COPY
  + EXPOSE
  + ADD
  + USER
  + VOLUME
  + WORKDIR
  + ENV
  + LABEL

To create images via docker file we should perform the below two steps

1. Create the docker file with above commands
2. Build an image using that file

**Use case**

1) Create a docker file using the base image Ubuntu and specify the name of the author

* # vim dockerfile
* Go into insert mode by pressing ‘i ‘

FROM ubuntu

MAINTAINER Sai

* Save and quit(:wq)

To build an image using the above t…..

# docker build –t newdocker .

Note: -t is used for specifying a name for our image. represents current working directories i.e it will build an image based on the docker file present in our working directory.

2) Create a docker file based on alpine Linux image and executes some Linux commands in it

# vim dockerfile

Go into insert mode by pressing ‘I’

FROM alpine

MAINTAINER Sai

CMD [“date”]

CMD [“ls”, “-la”]

:wq

# dockerbuild –t newalpne .

**ENTEYPOINT**

This command is used for taking which comes from CMD as arguments

**Use case**

Create a docker file from busybox image set the ENTEYPOINT as cat command and open a file called /etc/passwd”.

# vim dockerfile

FROM busybox

MAINTAINER Sai

ENTERYPOINT [“/bin/cat”]

CMD [“/etc/passwd”]

# docker build -t newbusybox .

# docker run newbusybox

22/09/2017

**Use case**

1.Download Ubuntu image and then install git and maven ping and curl init.

Perform the above action thorough a docker files.

# vim dockerfile

FROM ubuntu

RUN apt-get update && apt-get install –y git \

Maven \

Oputils\* \

Curl \

:wq

# docker build –t myUbuntu .

Run the myUbuntu image created using the docker file

# docker run –it myUbuntu

# git –version

# mvn –version

2. Create 5 docker images using the same dockerfile that we have created in the previous use case.

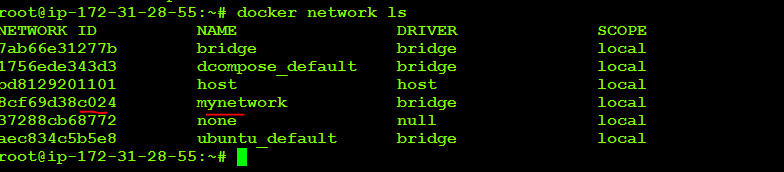
* Create a shell script with the name ‘myscript.sh’
* # vim myscript.sh
* Go Into insert mode by pressing ‘I’
* For I in {1..5}
* do
* docker build –t myUbuntu$i .
* :wq
* Give execute permissions on the above shell script
* # chmodu+xmyscript.sh
* Run the shell script using
* ./myscripts.sh

03/10/2017

**Docker Networking**

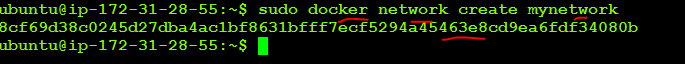
To see the list of networks available

# docker network ls



To create new network

# docker network create network\_name



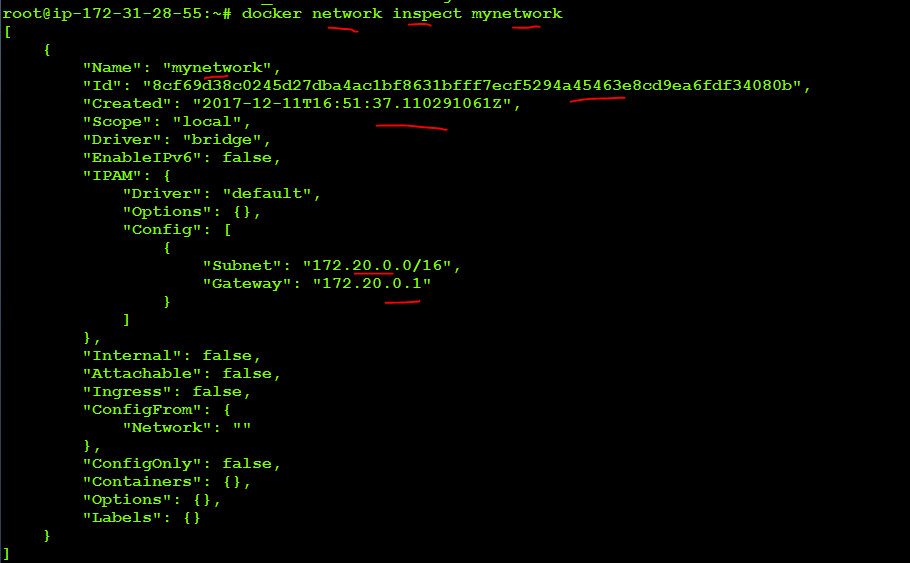
To remove/delete a network



# docker network rm network\_name

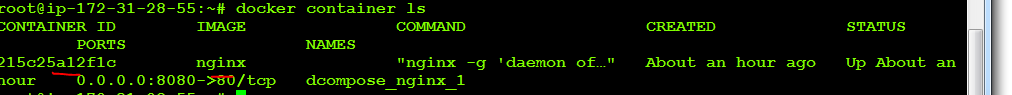
To find the information about the network

# docker network inspect network\_name



To attach a container to a network

# docker network connect network\_name container\_name



To disconnect the network to containere

# docker network disconnect network\_name container\_name

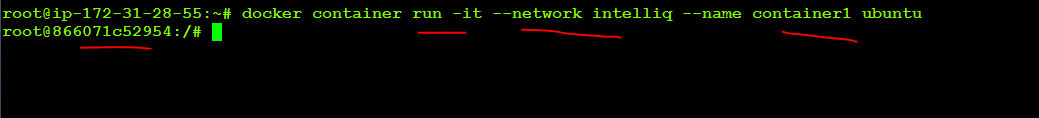
**Use case:**

Create two Ubuntu containers with container1, container2. Create a n/w intelliq. Assign these two containers to intelliq network.

Docker network create intelliq



Create Ubuntu container with name as container1



Press Ctrl+P+Q and Come out of container1 without exiting



Create Ubuntu container with name as container2

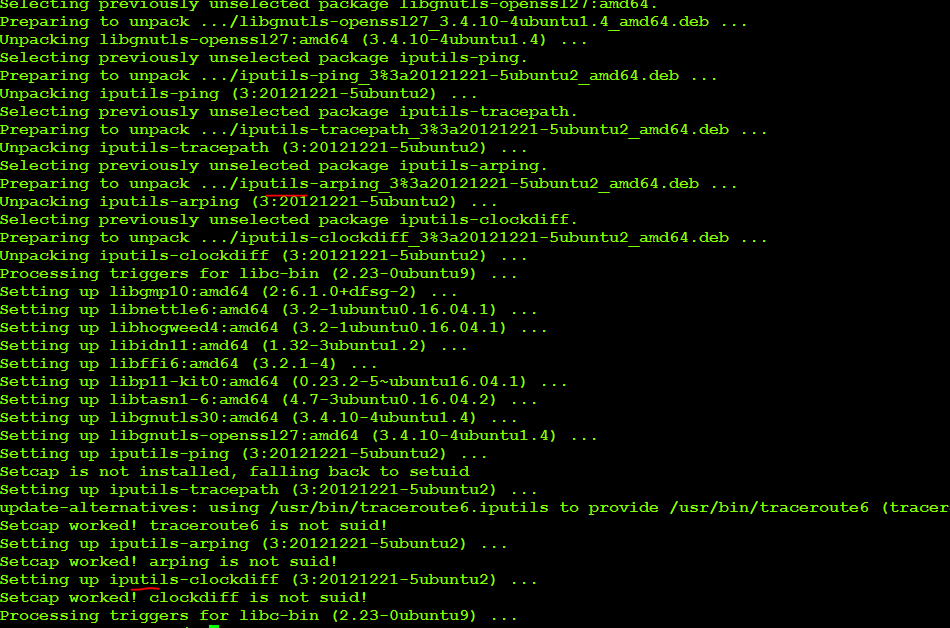




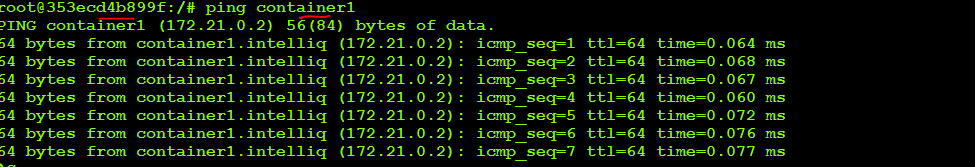
Now install ping package in container2

Apt-get update

Apt-get install –y iputils\*



Now ping to container1 from container2 to check connectivity between two containers. If both connected then it means both containers are in same network.



Create a network called intelliq1 another network called intelliq2

Create 3 busy box containers container1 container2 container3

Start container1 and container2 on intelliq1 network and check if they are pining not

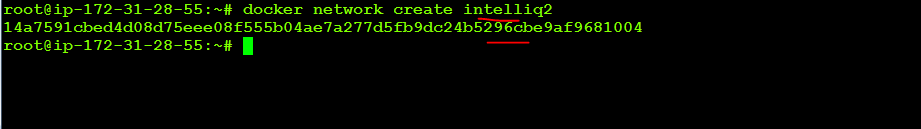
Start container3 is intelliq2 network and check that it cannot ping to containere1 and container2

Now container1 should be able to communicate with container2 but cannot communicate with container3 similarly container3 should be able to communicate with container2 but not with container1

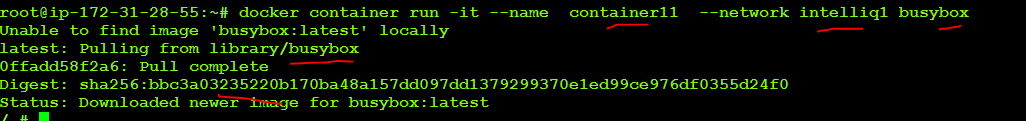
# docker network create intelliq1



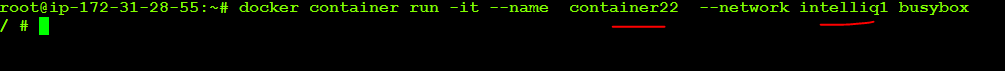
# docker network create intelliq2



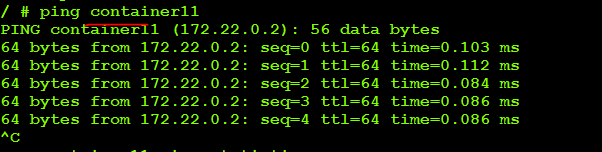
# docker run –itd –name container11 –network intelliq1 **busybox**

****

**# docker run –it –name container22 –network intelliq1 busybox in container22**

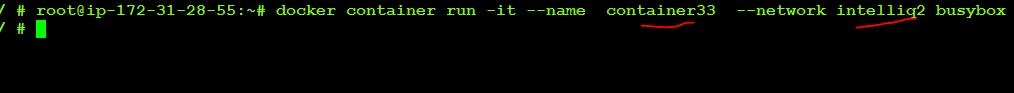
****

**Ping contaienr1: it should successfully ping**

****

**Ctrl+p+q to come out of container22**

**# docker run –it --name container33 –network intelliq2 busybox**

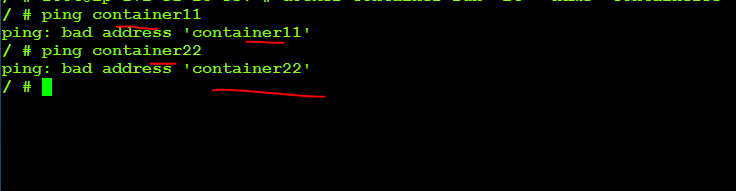
****

**In container33**

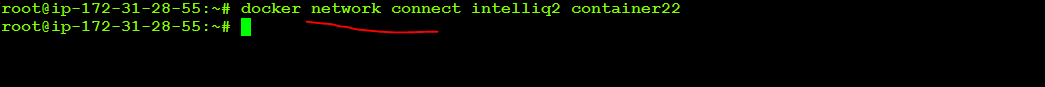
**Ping container11**

**Ping container22**

**It should not be able to ping because contaiener1 and containere2 are running intelliq1 network**

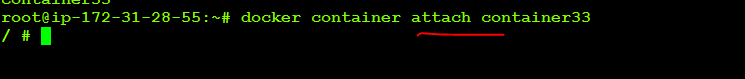
****

**# docker network connect intelliq2 container2**

****

**The above command will attach container2 to intelliq2 network…container2 is now running on both intelliq1 and intelliq2 network**

**Docker container attach container3**

****

**Ping container2: it will ping successfully**

**Building docker images using docker file**

Docker file is a simple test file which uses specific commands using which it is possible to create our own images.

It uses the following keywords to create or modify images

* + FROM
  + MAINTAINER
  + CMD
  + ENTERYPOINT
  + RUN
  + COPY
  + EXPOSE
  + ADD
  + USER
  + VOLUME
  + WORKDIR
  + ENV
  + LABEL

To create images via docker file we should perform the below two steps

1. Create the docker file with above commands
2. Build an image using that file

**Use case**

1) Create a docker file using the base image Ubuntu and specify the name of the author and run CMD on /etc/passwd

* # vim dockerfile
* Go into insert mode by pressing ‘i ‘

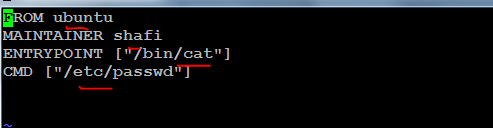
FROM ubuntu

MAINTAINER Sai

ENTRYPOINT [“/bin/cat”]

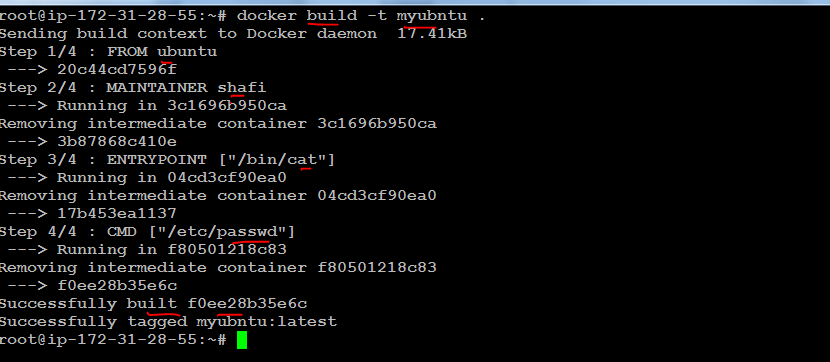
CMD [“/etc/passwd”

* Save and quit(:wq)



To build an image using the above t…..

# docker build –t newdocker .



Note: -t is used for specifying a name for our image. represents current working directories i.e it will build an image based on the docker file present in our working directory.

2) Create a docker file based on alpine Linux image and executes some Linux commands in it

# vim dockerfile

Go into insert mode by pressing ‘I’

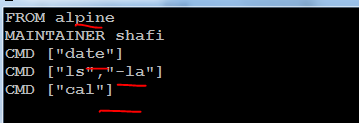
FROM alpine

MAINTAINER Sai

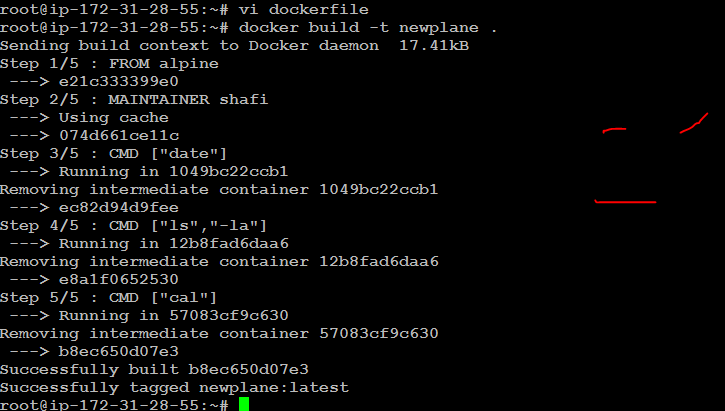
CMD [“date”]

CMD [“ls”, “-la”]

:wq



# dockerbuild –t newalpne .



**ENTEYPOINT**

This command is used for taking which comes from CMD as arguments

**Use case**

Create a docker file from busybox image set the ENTEYPOINT as cat command and open a file called /etc/passwd”.

# vim dockerfile

FROM busybox

MAINTAINER Sai

ENTERYPOINT [“/bin/cat”]

CMD [“/etc/passwd”]

# docker build -t newbusybox .

# docker run newbusybox

22/09/2017

**Use case**

1.Download Ubuntu image and then install git and maven ping and curl init.

Perform the above action thorough a docker files.

# vim dockerfile

FROM ubuntu

RUN apt-get update && apt-get install –y git \

Maven \

Oputils\* \

Curl \

:wq

# docker build –t myUbuntu .

Run the myUbuntu image created using the docker file

# docker run –it myUbuntu

# git –version

# mvn –version

2. Create 5 docker images using the same dockerfile that we have created in the previous use case.

* Create a shell script with the name ‘myscript.sh’
* # vim myscript.sh
* Go Into insert mode by pressing ‘I’
* For I in {1..5}
* do
* docker build –t myUbuntu$i .
* :wq
* Give execute permissions on the above shell script
* # chmodu+xmyscript.sh
* Run the shell script using
* ./myscripts.sh

