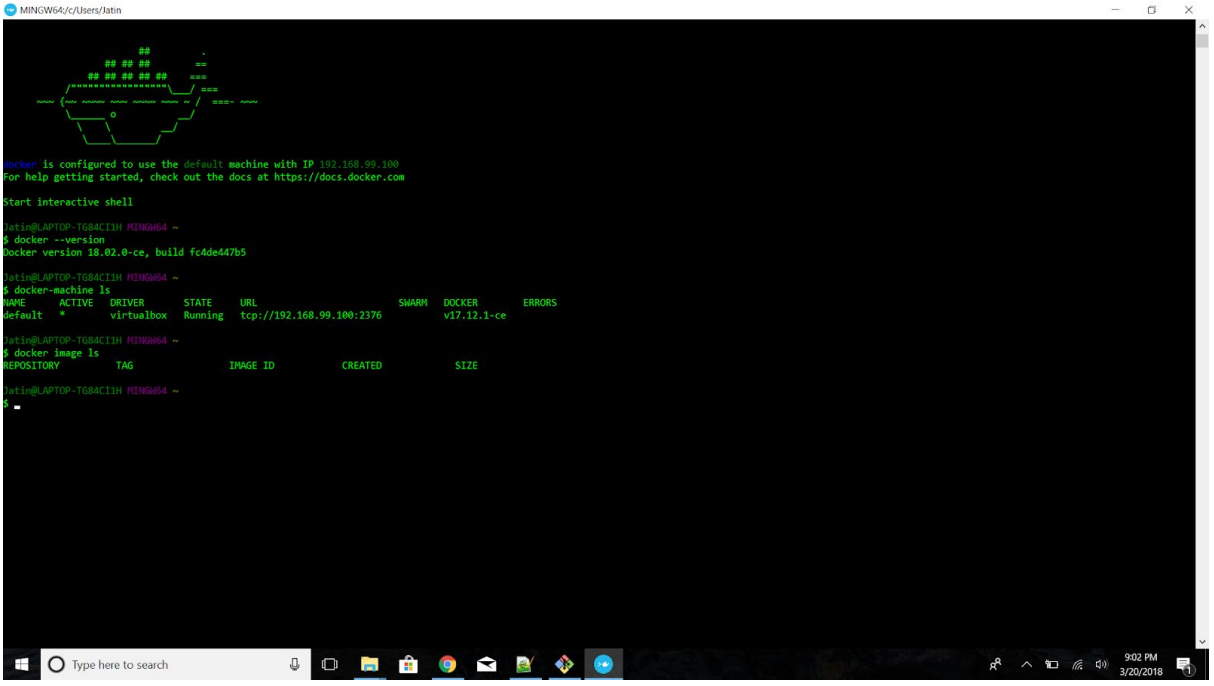
Docker Install - Use Docker Toolbox
https://docs.docker.com/toolbox/toolbox_install_windows/

First login to Docker Toolbox will give below results:



Simple commands to check first run:

docker --version	To check the version
docker-machine ls	To see the VM running docker - would be default
docker image ls	To see current images existing which would be empty

Lets create First Image

Create a directory newPackage(It can be any name)

```
mkdir newPackage
cd newPackage
```

Create following files inside it:

1. Dockerfile (file which creates the image)
2. requirements.txt (file which we have referenced in Dockerfile to keep our requirement softwares separate)
3. app.py (Actual Project/application)

Files are present in the project. Keep them in your directory newPackage at C://User/Name/

To build the image

```
docker build -t firstimage .
```

Check if image is created

```
docker image ls
```

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker image ls
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
firstimage          latest              154f028a8c46        About a minute ago  150MB
python              2.7-slim            55d87c387a8c        19 hours ago        139MB
```

firstimage is the image you just created. Python is downloaded as parent image mentioned in Dockerfile.

Running the image we just created - Containerization

docker run -d -p 4000:80 firstimage

-d = running in background

-p = giving port 4000 of local machine to link to 80 port of VM/docker image

Run below command to get IP address of your VM

docker-machine ls

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker run -p 4000:80 firstimage
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)

Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ERRORS
default   *        virtualbox   Running   tcp://192.168.99.100:2376           v17.12.1-ce
```

Open page http://IPADDRESS:4000/

Redis hasn't been installed that is why we get the error for Redis.

Stop this container

docker container ls

docker stop f0f54bc28338

docker container ls

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS                   NAMES
f0f54bc28338        firstimage          "python app.py"     7 minutes ago       Up 7 minutes        0.0.0.0:4000->80/tcp    awesome_blackwell

Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker stop f0f54bc28338
f0f54bc28338

Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker container ls
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
```

Share an Image

Login to Repository and tag your image (tag spec - username/repository:tag)
docker login
docker tag firstimage jatin10arora/starter:part1
docker image ls

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username (jatin10arora):
Password:
Login Succeeded

Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker tag firstimage jatin10arora/starter:part1

Jatin@LAPTOP-TG84CI1H MINGW64 ~/newPackage
$ docker image ls
REPOSITORY            TAG                 IMAGE ID            CREATED             SIZE
firstimage            latest              154f028a8c46        About an hour ago   150MB
jatin10arora/starter  part1               154f028a8c46        About an hour ago   150MB
python                2.7-slim            55d87c387a8c        20 hours ago        139MB
```

Push this image
docker push jatin10arora/starter:part1

Pull this image
docker run -p 4000:80 jatin10arora/starter:part1

Running Services/Swarms

Single Machine swarm +stack(multiple services are in files)
Create a new directory serviceFolder
mkdir serviceFolder
cd serviceFolder

Create the docker-compose.yml file at the location. File is present in the project. Change the entry in the file to your own repository with username and tag.

Running the image as a service
docker-machine ls

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ERRORS
default   *        virtualbox   Running   tcp://192.168.99.100:2376           v17.12.1-ce
```

docker swarm init --advertise-addr 192.168.99.100
docker stack deploy -c docker-compose.yml firstservicelab
docker service ls

```
$ docker swarm init --advertise-addr 192.168.99.100
Swarm initialized: current node (uuuc78istlc9kocgwwdwmktle) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-1nibpuchhrfc5mttonhukgs2v3h6vt8feqyiuzif3ety4w6t42-e08at9gmemav7zehfvnpyvial 192.168.99.100:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.


Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker stack deploy -c docker-compose.yml firstservicelab
Creating network firstservicelab_webnet
Creating service firstservicelab_visualizer
Creating service firstservicelab_redis
Creating service firstservicelab_web

Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker service ls
ID                  NAME                      MODE                REPLICAS            IMAGE                             PORTS
lw2bqv4omwj0        firstservicelab_redis     replicated          0/1                 redis:latest                      *:6379->6379/tcp
xunaj730z1yo        firstservicelab_visualizer replicated         0/1                 dockersamples/visualizer:stable   *:8080->8080/tcp
8yp1hfyef7zx        firstservicelab_web       replicated          0/5                 jatin10arora/starter:part1        *:80->80/tcp
```

View all the tasks(container in a service) in for a service:
docker service ps firstservicelab_web

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker service ps firstservicelab_web
ID                  NAME                    IMAGE                     NODE                DESIRED STATE       CURRENT STATE            ERROR               PORTS
xhra3h7zzxft        firstservicelab_web.1   jatin10arora/starter:part1   default             Running             Running about a minute ago
s3gfc4c8e7q7        firstservicelab_web.2   jatin10arora/starter:part1   default             Running             Running about a minute ago
lcpts5ekkom8        firstservicelab_web.3   jatin10arora/starter:part1   default             Running             Running about a minute ago
lcmgedn69raa        firstservicelab_web.4   jatin10arora/starter:part1   default             Running             Running about a minute ago
m1dey8619trr        firstservicelab_web.5   jatin10arora/starter:part1   default             Running             Running about a minute ago
```

Output of the service:

Tear down the current structure:
docker stack rm firstservicelab
docker swarm leave --force

Multi Machine Swarm

Create multiple virtual machines
docker-machine create -driver virtualbox myvm1
docker-machine create -driver virtualbox myvm2

Create swarm on myvm1
docker-machine ls
docker-machine ssh myvm1 “docker swarm init --advertise-addr 192.168.99.101”

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ERRORS
default   *        virtualbox   Running   tcp://192.168.99.100:2376           v17.12.1-ce
myvm1     -        virtualbox   Running   tcp://192.168.99.101:2376           v17.12.1-ce
myvm2     -        virtualbox   Running   tcp://192.168.99.102:2376           v17.12.1-ce

Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker-machine ssh myvm1  "docker swarm init --advertise-addr 192.168.99.101"
Swarm initialized: current node (m3v6oo96xuuv7fs0m457yowxi) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-1ewj2sr1sziiyny2dah6zwkygnrev19gonljsk9hquom91tap1-3kiqbfl4lbae9wiewm1kp4zpc 192.168.99.101:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
```

Join the other vm as a worker

docker-machine ssh myvm2 “docker swarm join --token SWMTKN-1-1ewj2sr1sziiyny2dah6zwkygnrev19gonljsk9hquom91tap1-3kiqbfl4lbae9wiewm1kp4zpc 192.168.99.101:2377”

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker-machine ssh myvm2 "docker swarm join --token SWMTKN-1-1ewj2sr1sziiyny2dah6zwkygnrev19gonljsk9hquom91tap1-3kiqbfl4lbae9wiewm1kp4zpc 192.168.99.101:2377"
This node joined a swarm as a worker.
```

Switch local ssh to myvm1 ssh

docker-machine env myvm1

Run the last instruction present:

eval $("C:\Users\Jatin\Documents\Softwares\Docker Toolbox\docker-machine.exe" env myvm1)

docker-machine ls

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker-machine env myvm1
export DOCKER_TLS_VERIFY="1"
export DOCKER_HOST="tcp://192.168.99.101:2376"
export DOCKER_CERT_PATH="C:\Users\Jatin\.docker\machine\machines\myvm1"
export DOCKER_MACHINE_NAME="myvm1"
export COMPOSE_CONVERT_WINDOWS_PATHS="true"
# Run this command to configure your shell:
# eval $("C:\Users\Jatin\Documents\Softwares\Docker Toolbox\docker-machine.exe" env myvm1)

Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ eval $("C:\Users\Jatin\Documents\Softwares\Docker Toolbox\docker-machine.exe" env myvm1)

Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker-machine ls
NAME      ACTIVE   DRIVER       STATE     URL                         SWARM   DOCKER        ERRORS
default   -        virtualbox   Running   tcp://192.168.99.100:2376           v17.12.1-ce
myvm1     *        virtualbox   Running   tcp://192.168.99.101:2376           v17.12.1-ce
myvm2     -        virtualbox   Running   tcp://192.168.99.102:2376           v17.12.1-ce
```

Star marker tells you which machine is active for you.

Deploy application on group of VMs

docker stack deploy -c docker-compose.yml firstswarmlab

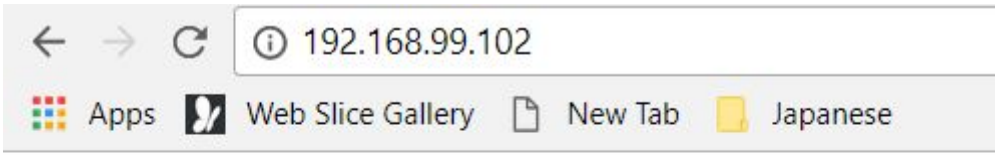
docker stack ps firsllabswarm

```
Jatin@LAPTOP-TG84CI1H MINGW64 ~/serviceFolder
$ docker stack ps firstswarmlab
ID            NAME                    IMAGE                             NODE      DESIRED STATE   CURRENT STATE            ERROR   PORTS
ndszqylu5s8y  firstswarmlab_web.1     jatin10arora/starter:part1        myvm2     Running         Preparing 47 seconds ago
j150xjh5ii3f  firstswarmlab_redis.1   redis:latest                      myvm1     Running         Preparing 52 seconds ago
ou7snicu4lm5  firstswarmlab_visualizer.1  dockersamples/visualizer:stable  myvm1  Running         Preparing 53 seconds ago
ixtnelwhbiuz  firstswarmlab_web.2     jatin10arora/starter:part1        myvm1     Running         Preparing 47 seconds ago
4wa0ji3l1c45  firstswarmlab_web.3     jatin10arora/starter:part1        myvm2     Running         Preparing 48 seconds ago
apbidxb5chno  firstswarmlab_web.4     jatin10arora/starter:part1        myvm1     Running         Preparing 48 seconds ago
lemnh2nrkrl4  firstswarmlab_web.5     jatin10arora/starter:part1        myvm2     Running         Preparing 47 seconds ago
```

Output of lab:

Hello World!

Hostname: 4814c5031dc5
Visits: *cannot connect to Redis, counter disabled*



Hello World!

Hostname: 4814c5031dc5
Visits: *cannot connect to Redis, counter disabled*

To start and stop a docker machine
docker-machine start myvm1
docker-machine stop myvm2

Redis doesn't work since it requires a directory.
docker-machine ssh myvm1 "mkdir ./data"

Currently the app.py has some error which will cause it to fail.

Visualizer output can be seen as below:


192.168.99.101:8080
Apps
Web Slice Gallery
New Tab
Japanese
firstswarmlab_visualizer
image : visualizer:stable@sha256:bc
tag : stable@sha256:bc680132f772c
updated : 20/3 23:23
e371cf76d7cb83ce15baa82ac3f375
state : running
firstswarmlab_web
image : starter:part1@sha256:b82d0
tag : part1@sha256:b82d0fede5856
updated : 20/3 23:23
4118a394d4ead91a12ab4ff3d56792
state : running
firstswarmlab_redis
image : redis:latest@sha256:26c93c
tag : latest@sha256:26c93c5b06eaa
cmd : redis-server,--appendonly,yes
updated : 20/3 23:41
de34b47aee1847272b56fe1a2fc469
state : running
firstswarmlab_web
image : starter:part1@sha256:b82d0
tag : part1@sha256:b82d0fede5856
updated : 20/3 23:23
4814c5031dc53f80a8298a2812de79
state : running
firstswarmlab_web
image : starter:part1@sha256:b82d0
tag : part1@sha256:b82d0fede5856
updated : 20/3 23:23
3d8cc7219fb5d38f71604127b3e6f0
state : running
firstswarmlab_web
image : starter:part1@sha256:b82d0
tag : part1@sha256:b82d0fede5856
updated : 20/3 23:23
1c40d55b0605f0e567b9022208eb7
state : running
firstswarmlab_web
image : starter:part1@sha256:b82d0
tag : part1@sha256:b82d0fede5856
updated : 20/3 23:23
fc1d45e8c880da8fb0346b37832950
state : running