How to change Normal user to Docker user ?

-- Here we have to add particullar user to docker daemon.

-- with goldentech user we have to type

```
docker ps
```

first we have to check what are the groups associated with the user id ?

```
id goldentech
usermod goldentech -aG docker
id goldentech
```

Now login with goldentech user and check connecting docker or not with following command

```
docker images
```

if you get below line output means docker has not been started.

-- Cannot connect to the Docker daemon. Is the docker daemon running on this host?

-- That means we are unable to connect docke engine.

then type sudo su - (convert my user id to root user)

```
sudo su -
```

How to start docker engine

systemctl start docker

we can make the docker availability at the time of reboot use following the command.

```
systemctl enable docker
```

Now am going to run container with specified command and immediate i want to delete that container.

docker run --rm centos pwd

--pwd means it is present working directory it is a "/" will be the output.

so once you enter into the conatiner the default directory is "/" that is what it is telling.

and just check the conatiner deleted or not ?

docker ps -a -- i can see container has been removed after running immediately.

```
root@ubuntu:~# docker run --rm centos pwd
/
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES
c75bc2abf07e        centos              "/bin/bash"         18 hours ago
Exited (0) 18 hours ago                         prickly_goldwasser
421e0d31f99a        centos              "ls -lrt"           18 hours ago
Exited (0) 18 hours ago                         desperate_volhard
667bc8763fd5        centos              "ls"                18 hours ago
Exited (0) 18 hours ago                         awesome_euclid
90553ff825ba        centos              "/bin/bash"         18 hours ago
Exited (0) 18 hours ago                         elegant_shannon
```

Now i want to use another command called ping command wether am able to connect to or not.

docker run --rm centos ping -c 3 google.com

once you enter above command you have run some process and giving output and am done with container and container can be deleted. docker ps –a

```
root@ubuntu:~# docker run --rm centos ping -c 3 google.com
PING google.com (216.58.196.110) 56(84) bytes of data.
64 bytes from maa03s19-in-f14.1e100.net (216.58.196.110): icmp_seq=1 ttl=127 tim
e=27.2 ms
64 bytes from maa03s19-in-f14.1e100.net (216.58.196.110): icmp_seq=2 ttl=127 tim
e=26.9 ms
64 bytes from maa03s19-in-f14.1e100.net (216.58.196.110): icmp_seq=3 ttl=127 tim
e=24.3 ms

--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 24.344/26.180/27.266/1.305 ms
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS                      PORTS               NAMES
c75bc2abf07e        centos              "/bin/bash"         18 hours ago
Exited (0) 18 hours ago                         prickly_goldwasser
421e0d31f99a        centos              "ls -lrt"           18 hours ago
Exited (0) 18 hours ago                         desperate_volhard
667bc8763fd5        centos              "ls"                18 hours ago
Exited (0) 18 hours ago                         awesome_euclid
90553ff825ba        centos              "/bin/bash"         18 hours ago
Exited (0) 18 hours ago                         elegant_shannon
root@ubuntu:~#
```

Now I want to run some non interactive commands while running commands.

I want to do yum install on container using following command

docker run centos yum install httpd

so here httpd install using yum package requires our input while installing httpd.

```
Transaction Summary
================================================================================
Install  1 Package (+5 Dependent packages)

Total download size: 24 M
Installed size: 32 M
Is this ok [y/d/N]: Exiting on user command
Your transaction was saved, rerun it with:
 yum load-transaction /tmp/yum_save_tx.2017-10-17.22-08.8cvEkA.yumtx
root@ubuntu:~#
```

It didn't wait for me to provide input it was automatically exited without user input.

If you just type command "w" -- it is will just display the currently logged in users list.

```
root@ubuntu:~# w
 15:25:41 up 12:36,  1 user,  load average: 0.25, 0.18, 0.14
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
goldente tty7     :0               Mon19   20:05m 12:59   0.41s /bin/sh /usr/li
root@ubuntu:~#
```

So above screen currently logged in only one user called goldentech user. So here admin is logged in from which ip address with my internet ip address and here you can find out few more details like idle time and Login details.

And here you can find out one more word called "TTY". It is a special file in the system which accepts the input from you.so you are typing via keyboard means hardware and someone is taking this input from hardware in linux they use TTY special file. That's all about TTY file.

So Now am going to run container with interactive mode like below.

'docker run –I centos yum install httpd'

This also wont work ofcourse you are enabling interaction but you cant provide interaction why because there no one to take the input from user.you will get same above window

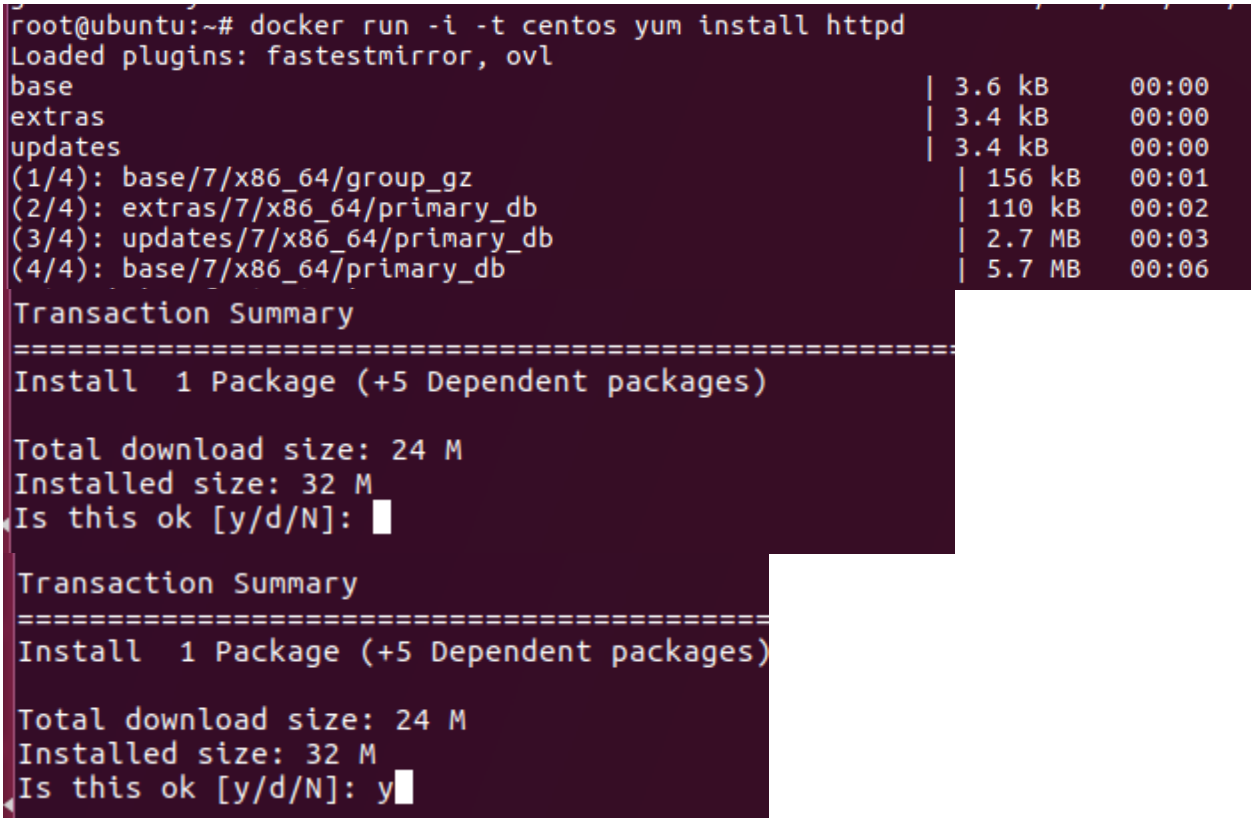
```
Transaction Summary
================================================================================
Install  1 Package (+5 Dependent packages)

Total download size: 24 M
Installed size: 32 M
Is this ok [y/d/N]: Exiting on user command
Your transaction was saved, rerun it with:
 yum load-transaction /tmp/yum_save_tx.2017-10-17.22-08.8cvEkA.yumtx
root@ubuntu:~#
```

So here it s not taking input here, you are allowing to interaction but but no one to take input from you. So to enable that option we have –it or –I –t .

Here “t” means a TTY enabled option in the docker that means you type something and TTY will take the input from you.

docker run –it centos yum install httpd



So the commands are 2 types

1. Interactive commands
2. Non-interactive commands

Now you can check the containers list now using “docker ps –a”

```
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
  STATUS                      PORTS               NAMES
ac005678ed1a        centos              "yum install httpd"   4 minutes ago
  Exited (0) 16 seconds ago                       agitated_franklin
fe603606858b        centos              "yum install httpd"   31 minutes ago
  Exited (1) 31 minutes ago                       goofy_spence
ea715056dc8f        centos              "yum install httpd"   37 minutes ago
  Exited (1) 37 minutes ago                       big_brown
c75bc2abf07e        centos              "/bin/bash"           19 hours ago
  Exited (0) 19 hours ago                         prickly_goldwasser
421e0d31f99a        centos              "ls -lrt"             19 hours ago
  Exited (0) 19 hours ago                         desperate_volhard
667bc8763fd5        centos              "ls"                  19 hours ago
  Exited (0) 19 hours ago                         awesome_euclid
90553ff825ba        centos              "/bin/bash"           19 hours ago
  Exited (0) 19 hours ago                         elegant_shannon
root@ubuntu:~#
```

Here centos the default command for running centos is "/bin/bash" that means you are going to get a shell environment. So if you are starting a shell it requires interaction and else interaction is not enabled it is exiting automatically.

Now we can run with default command with interactive mode like

docker run –it centos

now you are entering inside of container now. It will work now on default command /bin/bash

In container I want to check my script is working or not script should be anything like below

```
#!/bin/bash

[ -z "$1" ] && exit 1

id $1 &>/dev/null

if [ $? -ne 0 ] ; then

    useradd $1

else

  echo User already exists

fi
```

```
[root@c412c45b389f /]# cat useradd
#!/bin/bash

[ -z "$1" ] && exit 1

id $1 &>/dev/null

if [ $? -ne 0 ] ; then
      useradd $1
else
   echo User already exists
fi
[root@c412c45b389f /]#
```

```
[root@c412c45b389f /]# ./useradd krishna
[root@c412c45b389f /]# ./useradd krishna
User already exists
[root@c412c45b389f /]#
```

Iam done with my script and my script is working fine and my container job also done. I can come out using exit command. So when you use container in interaction mode and if you say exit then you are exted from the container.

But the same container I want to run this in background so we have one more option called “-d” daemon mode or detached mode.

docker run –d centos

docker ps

It is not running backgroung coming out from container since we have not given any command why because the default command for centos is /bin/bash so there is command to run so if we provide default command like below it is going to be run in background.

docker run –d centos sleep 5000

docker ps

Or else if you want to run /bin/bash default we have to write like below command.

docker run –itd centos

docker ps

with out –it and if you are not providing any commands to container it will exit because /bin/bash will need interaction as well as terminal, if you have those options only this can be run when we have default command is /bin/bash even in the background.

Ok

docker ps –a

docker rm $(docker ps –a -q)

docker run centos

It will be exited --- docker ps –a

docker run –itd centos

docker ps

You can see the container is up and running in background.

docker ps –a

If we want to go to inside of container which was running in background I have to use attach command

docker attach <container id>

if you want come out of container we have to use ctrl+p plus ctrl+q.

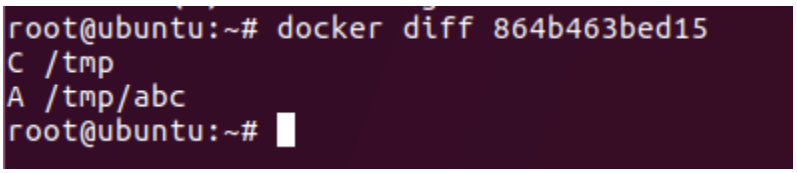
How to commit the container.

docker run centos yum install httpd –y

docker run centos touch /tmp/abc

if you want to check the differences between the conatiners you can use

docker diff <latest container id>



If we want to create new image with these differences we have to use commit command.

Syntax wil given below

docker commit -m "ubuntupython" -a "siva" f56be454c884 sivadevops/ubuntupython:v1

docker push sivadevops/ubuntupython:v1

docker pull sivadevops/ubuntupython:v1

```
root@ubuntu:~# docker commit
"docker commit" requires at least 1 and at most 2 argument(s).
See 'docker commit --help'.

Usage:  docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]

Create a new image from a container's changes
root@ubuntu:~#
```

```
root@ubuntu:~# docker commit 864b463bed15 siva/centos-abc
sha256:00cbc2782d63168342831fa026557548391d306149617b3584bc7d859f921854
root@ubuntu:~# docker images
REPOSITORY              TAG                 IMAGE ID            CREATED
     SIZE
siva/centos-abc         latest              00cbc2782d63        7 seconds ago
     196.6 MB
```

Here container id I can use or firstname_lastname I can use.

Cp command :

```
root@ubuntu:~# docker cp
"docker cp" requires exactly 2 argument(s).
See 'docker cp --help'.

Usage:  docker cp [OPTIONS] CONTAINER:SRC_PATH DEST_PATH|-
        docker cp [OPTIONS] SRC_PATH|- CONTAINER:DEST_PATH

Copy files/folders between a container and the local filesystem
root@ubuntu:~#
```

```
root@ubuntu:~# docker cp ac005678ed1a:/etc/httpd/conf/httpd.conf .
root@ubuntu:~# ls -lrt
total 4184
drwxr-xr-x 2 root root    4096 May 25 20:22 docker_Goldentech
drwxr-xr-x 2 root root    4096 May 28 23:36 dockerfile
-rw------- 1 root root 4236288 May 29 22:56 test.tar
drwxr-xr-x 2 root root    4096 Aug  4 23:29 dockerimage1
drwxr-xr-x 2 root root    4096 Aug  5 00:07 apacheserverimage
drwxr-xr-x 2 root root    4096 Aug 17 08:58 testimage1
drwxr-xr-x 2 root root    4096 Aug 17 09:05 testimage2
drwxr-xr-x 2 root root    4096 Aug 17 09:17 testimage3
drwxr-xr-x 2 root root    4096 Aug 17 09:40 testimage4
-rw-r--r-- 1 root root   11753 Oct 11 10:07 httpd.conf
root@ubuntu:~#
```

We can copy the files from container to local system.

Create command:

It wll just create a container from an image.

Docker create centos

```
root@ubuntu:~# docker create centos
ab45a8a80e9bc5209ea56111accb8f7c65e79cedcdf50c45b5fa66f3ae066909
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
  STATUS                      PORTS               NAMES
ab45a8a80e9b        centos              "/bin/bash"              5 seconds ago
  Created                                         infallible_lamport
864b463bed15        centos              "touch /tmp/abc"         13 minutes ago
  Exited (0) 13 minutes ago                       awesome_thompson
```

It will just create a container and it is not ready for use so then we have to start the container using start command

Docker start <containerid>

```
root@ubuntu:~# docker start ab45a8a80e9b
ab45a8a80e9b
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
  STATUS                      PORTS               NAMES
ab45a8a80e9b        centos              "/bin/bash"              2 minutes ago
  Exited (0) 8 seconds ago                        infallible_lamport
864b463bed15        centos              "touch /tmp/abc"         15 minutes ago
  Exited (0) 15 minutes ago                       awesome_thompson
```

Events command:

docker events -- it will just display the events ok

go to another terminal window and type this command docker events

```
root@ubuntu:~# docker events
```

It is having nothing to show you

And then goto our main terminal and delete one container and check the second terminal for events

```
root@ubuntu:~# docker rm ab45a8a80e9b
ab45a8a80e9b
root@ubuntu:~#
```

```
root@ubuntu:~# docker events
2017-10-17T16:57:32.027560382-07:00 container destroy ab45a8a80e9bc5209ea56111ac
cb8f7c65e79cedcdf50c45b5fa66f3ae066909 (build-date=20170911, image=centos, licen
se=GPLv2, name=infallible_lamport, vendor=CentOS)
```

And again goto main terminal and try to create one container

docker run centos echo hello

```
root@ubuntu:~# docker run centos
root@ubuntu:~#
```

And goto second terminal which opend for events

```
root@ubuntu:~# docker events
2017-10-17T17:01:34.252886610-07:00 container create 351492ea98e704534872b83b71a
ed12cadc922b930a8e7053e8a5db482ae3941 (build-date=20170911, image=centos, licens
e=GPLv2, name=romantic_khorana, vendor=CentOS)
2017-10-17T17:01:34.258987324-07:00 container attach 351492ea98e704534872b83b71a
ed12cadc922b930a8e7053e8a5db482ae3941 (build-date=20170911, image=centos, licens
e=GPLv2, name=romantic_khorana, vendor=CentOS)
2017-10-17T17:01:34.479122477-07:00 network connect 05bf8aabe7c588841a3e53d1a48c
cced250d340803b179da5b6ca522e51f1ea5 (container=351492ea98e704534872b83b71aed12c
adc922b930a8e7053e8a5db482ae3941, name=bridge, type=bridge)
2017-10-17T17:01:36.390380514-07:00 container start 351492ea98e704534872b83b71ae
d12cadc922b930a8e7053e8a5db482ae3941 (build-date=20170911, image=centos, license
=GPLv2, name=romantic_khorana, vendor=CentOS)
2017-10-17T17:01:36.392673358-07:00 container die 351492ea98e704534872b83b71aed1
2cadc922b930a8e7053e8a5db482ae3941 (build-date=20170911, exitCode=0, image=cento
s, license=GPLv2, name=romantic_khorana, vendor=CentOS)
2017-10-17T17:01:36.573683241-07:00 network disconnect 05bf8aabe7c588841a3e53d1a
48ccced250d340803b179da5b6ca522e51f1ea5 (container=351492ea98e704534872b83b71aed
12cadc922b930a8e7053e8a5db482ae3941, name=bridge, type=bridge)
```

Exec command :

Run a command in a running container.

```
root@ubuntu:~# docker run -itd centos
214583ba6f19a80a3f7cea6dda8dcd3179f98d6c4d1b4814d1e1c9156c5a771e
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
214583ba6f19        centos              "/bin/bash"         5 seconds ago
Up 3 seconds                            big_payne
root@ubuntu:~# docker exec
docker: "exec" requires a minimum of 2 arguments.
See 'docker exec --help'.

Usage:  docker exec [OPTIONS] CONTAINER COMMAND [ARG...]

Run a command in a running container
root@ubuntu:~# docker exec 214583ba6f19 echo hello world
hello world
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
214583ba6f19        centos              "/bin/bash"         57 seconds ago
Up 55 seconds                           big_payne
root@ubuntu:~#
```

It just ran a output inside the container.

```
root@ubuntu:~# docker exec 214583ba6f19 uname
Linux
root@ubuntu:~# docker exec 214583ba6f19 uname -a
Linux 214583ba6f19 4.10.0-37-generic #41-Ubuntu SMP Fri Oct 6 20:20:37 UTC 2017
x86_64 x86_64 x86_64 GNU/Linux
root@ubuntu:~# docker exec 214583ba6f19 hostname
214583ba6f19
root@ubuntu:~# hostname
ubuntu
root@ubuntu:~#
```

Just compare with hostname commands last command is running on my system and given output and remaining above commands ran inside of containers and given output. So if you want to execute any command inside of container I have to use exec command on running container.

History command

Show the history of an image.

```
root@ubuntu:~# docker history centos
IMAGE               CREATED             CREATED BY
        SIZE                COMMENT
196e0ce0c9fb        4 weeks ago         /bin/sh -c #(nop)  CMD ["/bin/bash"]
        0 B
<missing>           4 weeks ago         /bin/sh -c #(nop)  LABEL name=CentOS Bas
e Ima   0 B
<missing>           4 weeks ago         /bin/sh -c #(nop) ADD file:1ed4d1a29d09a
636dd   196.6 MB
root@ubuntu:~# docker history siva/centos-abc
IMAGE               CREATED             CREATED BY
        SIZE                COMMENT
00cbc2782d63        37 minutes ago      touch /tmp/abc
        0 B
196e0ce0c9fb        4 weeks ago         /bin/sh -c #(nop)  CMD ["/bin/bash"]
        0 B
<missing>           4 weeks ago         /bin/sh -c #(nop)  LABEL name=CentOS Bas
e Ima   0 B
<missing>           4 weeks ago         /bin/sh -c #(nop) ADD file:1ed4d1a29d09a
636dd   196.6 MB
root@ubuntu:~#
```

info : Display system-wide information. It shows some nformation about docker engine.

```
root@ubuntu:~# docker info
Containers: 12
 Running: 1
 Paused: 0
 Stopped: 11
Images: 24
Server Version: 1.12.6
Storage Driver: aufs
 Root Dir: /var/lib/docker/aufs
 Backing Filesystem: extfs
 Dirs: 40
 Dirperm1 Supported: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: host bridge overlay null
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Security Options: apparmor seccomp
Kernel Version: 4.10.0-37-generic
Operating System: Ubuntu 17.04
```

inspect :

Return low-level information on a container, image or task

docker inspect <image_id or container id>

Suppose your specified command is lengthy while running container example

docker run –itd centos-httpd /usr/sbin/httpd -D FOREGROUND

or

docker run –td centos-httpd touch /tmp/aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

```
root@ubuntu:~# docker run -itd centos touch /tmp/aaaaaaaaaaaaaaaaaaaaaaaaaaaaa
6e463e860d7e7108cddcb7a823eb0e313b7c1da9818e15791d4c5b642975251c
root@ubuntu:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
     STATUS                      PORTS               NAMES
6e463e860d7e        centos              "touch /tmp/aaaaaaaaa"   5 seconds ago
     Exited (0) 3 seconds ago                        tiny_kalam
d971dcdccfd1        centos              "touch /tmp/aaaaaaaaa"   25 seconds ago
     Exited (0) 22 seconds ago                       compassionate_kalam
```

IF you want to see entire command what has been used here I can go with inspect command will check the full command which was used above.

docker inspect <container_id>

```
],
"Cmd": [
    "touch",
    "/tmp/aaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
],
"Image": "centos",
```

logs command : Fetch the logs of a container

docker rm $(docker ps –a –q)

docker run –itd –p 80:80 httpd

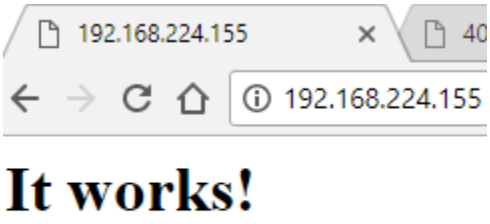
Now am going to check logs about the container

Docker logs –help -------- for parameters

docker logs <container id>

docker logs –f <container id>

Now you can go to browser and type IP address



And you can try in browser other window ip address with some other value like below

Now all the history will updated into logs terminal.

And go to logs terminal window and check the logs now everything will be added.

```
root@ubuntu:~# docker logs -f c8ac99ab2d00
AH00558: httpd: Could not reliably determine the server's fully qualified domain
 name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress thi
s message
AH00558: httpd: Could not reliably determine the server's fully qualified domain
 name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress thi
s message
[Wed Oct 18 23:18:22.687688 2017] [mpm_event:notice] [pid 1:tid 139866329724800]
 AH00489: Apache/2.4.28 (Unix) configured -- resuming normal operations
[Wed Oct 18 23:18:22.688761 2017] [core:notice] [pid 1:tid 139866329724800] AH00
094: Command line: 'httpd -D FOREGROUND'
192.168.224.1 - - [18/Oct/2017:23:19:15 +0000] "GET /sample HTTP/1.1" 404 204
192.168.224.1 - - [18/Oct/2017:23:19:16 +0000] "GET /favicon.ico HTTP/1.1" 404 2
09
192.168.224.1 - - [18/Oct/2017:23:19:47 +0000] "GET / HTTP/1.1" 200 45
192.168.224.1 - - [18/Oct/2017:23:20:47 +0000] "GET /sample HTTP/1.1" 404 204
172.17.0.1 - - [18/Oct/2017:23:21:46 +0000] "GET / HTTP/1.0" 200 45
172.17.0.1 - - [18/Oct/2017:23:26:46 +0000] "GET / HTTP/1.0" 200 45
```

To follow the logs am going to use this command.

Pause command :

Docker pause all processes within one or more containers

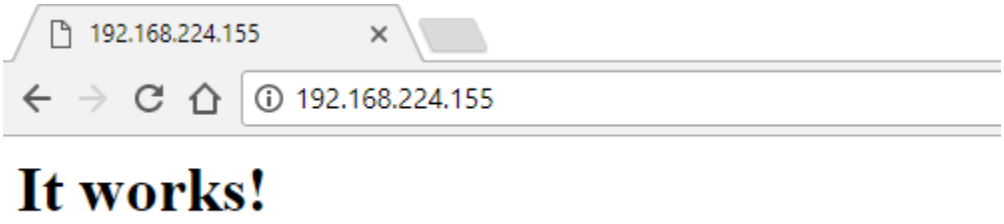
Unpause command

Docker unpause all processes within one or more containers

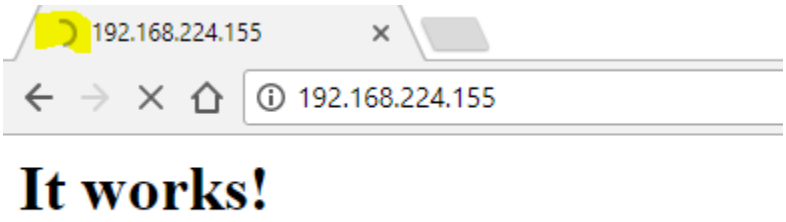
docker ps

```
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
 STATUS              PORTS               NAMES
c8ac99ab2d00        httpd               "httpd-foreground"       15 minutes ago
 Up 15 minutes       0.0.0.0:80->80/tcp   jovial_agnesi
root@ubuntu:~#
```

Goto browase and type your ipaddress like above logs example



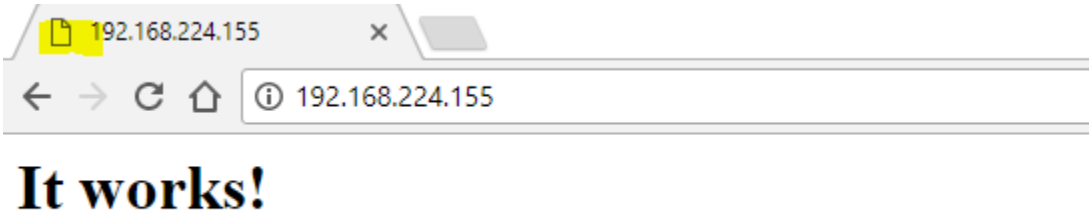

```
root@ubuntu:~# docker pause c8ac99ab2d00
c8ac99ab2d00
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
 STATUS                     PORTS               NAMES
c8ac99ab2d00        httpd               "httpd-foreground"       17 minutes ago
 Up 16 minutes (Paused)   0.0.0.0:80->80/tcp   jovial_agnesi
root@ubuntu:~#
```



Now we are unable to see the website it is just loading. Ok so we are playing a movie and we have done pause here.

Now we are unapuse the same container and website will again.

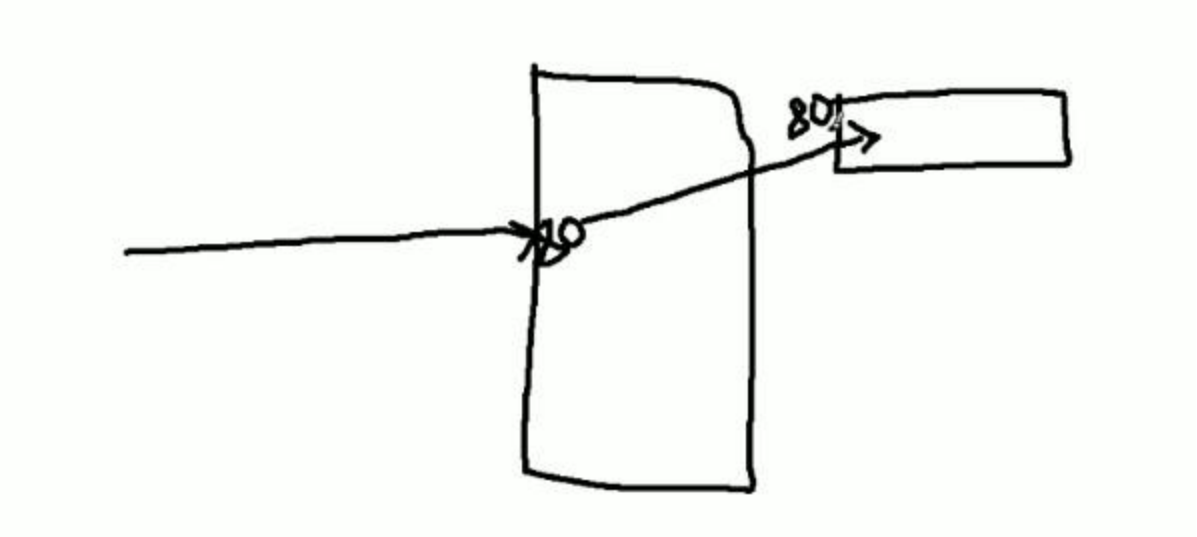
```
root@ubuntu:~# docker unpause c8ac99ab2d00
c8ac99ab2d00
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
 STATUS              PORTS               NAMES
c8ac99ab2d00        httpd               "httpd-foreground"       19 minutes ago
 Up 19 minutes       0.0.0.0:80->80/tcp   jovial_agnesi
root@ubuntu:~#
```



port ： List port mappings or a specific mapping for the container

docker port <container id>

```
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
 STATUS              PORTS               NAMES
c8ac99ab2d00        httpd               "httpd-foreground"       22 minutes ago
 Up 22 minutes       0.0.0.0:80->80/tcp   jovial_agnesi
root@ubuntu:~# docker port c8ac99ab2d00
80/tcp -> 0.0.0.0:80
root@ubuntu:~#
```



So you have a container running so here when ever user hit this server on this port number 80 so now what it will try to do is route the particular request to the particular container.

So how can we do this is we have to do some mapping saying that if any request comes to 80 port then it will redirect to particular 80 port

This configuaration we have to keep it while running the container

docker run –itd -p 80:80 httpd

so in our example we are publishing to the port number 80 my host machene to the port number 80 of my container.

Port mapping we can do while running the container later point of time we cant do that.

pull : Pull an image or a repository from a registry

push : Push an image or a repository to a registry

rename: Rename a container

restart : Restart a container

rm : Remove one or more containers

rmi : Remove one or more images

run : Run a command in a new container

search : Search the Docker Hub for images

It is going to search images bydeafult docker hub

```
root@ubuntu:~# docker search httpd
NAME                                DESCRIPTION
   STARS     OFFICIAL   AUTOMATED
httpd                               The Apache HTTP Server Project
   1294      [OK]
hypriot/rpi-busybox-httpd           Raspberry Pi compatible Docker Image with .
.  38
centos/httpd
   15                   [OK]
armhf/httpd                         The Apache HTTP Server Project
   8
macadmins/netboot-httpd             use in combination with bruienne/bsdpy
   4                    [OK]
```

We will pick always which is havng more stars and also it should be official OK.

stats : Display a live stream of container(s) resource usage statistics

we know TOP command right

```
top - 17:06:37 up 1 day,  4:32,  1 user,  load average: 0.38, 0.23, 0.17
Tasks: 257 total,   1 running, 256 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.3 us,  4.0 sy,  0.0 ni, 90.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  4025392 total,   599204 free,  1590780 used,  1835408 buff/cache
KiB Swap:  2097148 total,  2097148 free,        0 used.  2067344 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
  958 root      20   0  560668 141996  77832 S  3.6  3.5  33:19.51 Xorg
 4466 root      20   0  738132  54084  24140 S  1.6  1.3   2:26.09 dockerd
 7655 goldent+  20   0  755208  42004  29856 S  1.3  1.0   3:44.69 gnome-term+
```

So it is going to display the statistcks of current host running , where as if we want to display the statistics of my container running so we have to use STATS command.

```
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
 STATUS              PORTS               NAMES
c8ac99ab2d00        httpd               "httpd-foreground"       50 minutes ago
 Up 50 minutes       0.0.0.0:80->80/tcp  jovial_agnesi
root@ubuntu:~#
```

Docker stats <container id>

```
CONTAINER           CPU %               MEM USAGE / LIMIT     MEM %
    NET I/O             BLOCK I/O           PIDS
c8ac99ab2d00        1.16%               15.85 MiB / 3.839 GiB   0.40%
    18.81 kB / 14.8 kB   0 B / 0 B          109
```

It is just shows the stats of the container and this is what it will shows you like above screen.

What it is showing how much CPU is being use by the container how much memory consume by the container so here currently it is using 15 mb but maximum size it can use upto 3.839 GB ok that is total memory of the system.

So a single container can use your complete resources and we can limit that as well.

So memory consumption percentage .40% and network information so how much bytes currently transferred we can see here.

And if we want to increase all the values we can run some process on container and we can put lots of load here. So how to put some load is we can run small process on this container like below.

We will open new terminal and run the below command.

while true; do curl http://192.168.224.155/ ;sleep 0.5;done;

above command will do every half second the command will hit the server.

```
root@ubuntu:~# while true; do curl http://192.168.224.155/ ;sleep 0.5;done;
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
<html><body><h1>It works!</h1></body></html>
```

Now we wil go and check the stats about the same container.

```
CONTAINER           CPU %               MEM USAGE / LIMIT     MEM %
   NET I/O             BLOCK I/O           PIDS
c8ac99ab2d00        34.02%              15.9 MiB / 3.839 GiB   0.40%
   57.54 kB / 58.16 kB   0 B / 0 B           109
```

```
CONTAINER           CPU %               MEM USAGE / LIMIT     MEM %
   NET I/O             BLOCK I/O           PIDS
c8ac99ab2d00        39.49%              15.9 MiB / 3.839 GiB   0.40%
   77.34 kB / 80.38 kB   0 B / 0 B           109
```

```
CONTAINER           CPU %               MEM USAGE / LIMIT     MEM %
   NET I/O             BLOCK I/O           PIDS
c8ac99ab2d00        24.05%              15.91 MiB / 3.839 GiB   0.40%
   89.37 kB / 94.02 kB   0 B / 0 B           109
```

```
CONTAINER           CPU %               MEM USAGE / LIMIT       MEM %
    NET I/O                 BLOCK I/O           PIDS
c8ac99ab2d00        3.37%               15.91 MiB / 3.839 GiB   0.40%
    157.5 kB / 170.4 kB   0 B / 0 B             109
```

Now this command every half second it is going to hit the server and you can see that network bytes going up simultaneously if you see the cpu utilization also changing little continuously. Some times it goes up and some times it goes down.

Now am running ths command insdie my Ubuntu system but if we want to run the same command inside of container we can use exec command or I can put same code in a sample script and I will copy script file using cp command to a container.

vi sample

while true; do curl http://192.168.224.155/ ;sleep 0.5;done;

:wq

So we have to copy the script to the server(container)

docker cp sample <container_id>:/

docker exec <container_id> sh /sample

```
root@ubuntu:~# docker cp sample c8ac99ab2d00:/
root@ubuntu:~# docker exec c8ac99ab2d00 as /sample
rpc error: code = 13 desc = invalid header field value "oci runti
failed: container_linux.go:247: starting container process caused
\\\": executable file not found in $PATH\"\n"
root@ubuntu:~# docker exec c8ac99ab2d00 sh /sample
/sample: 1: /sample: curl: not found
/sample: 1: /sample: curl: not found
/sample: 1: /sample: curl: not found
/sample: 1: /sample: curl: not found
/sample: 1: /sample: curl: not found
/sample: 1: /sample: curl: not found
/sample: 1: /sample: curl: not found
/sample: 1: /sample: curl: not found
```

Curl is not available in container so we have to install curl first. Taking new container which curl is available.

```
root@ubuntu:~# docker run -itd centos
d42f80f614a50d7c46017b7b61313f6cad48b062df91bb20468d85245322aa36
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
 STATUS              PORTS               NAMES
d42f80f614a5        centos              "/bin/bash"         14 seconds ago
 Up 9 seconds                            berserk_liskov
d1e5535f4054        centos              "/bin/bash"         3 minutes ago
 Up 3 minutes                            tiny_feynman
c8ac99ab2d00        httpd               "httpd-foreground"  About an hour ago
 Up About an hour    0.0.0.0:80->80/tcp   jovial_agnesi
root@ubuntu:~#
```

```
root@ubuntu:~# docker cp sample d1e5535f4054:/
root@ubuntu:~# docker exec d1e5535f4054 sh /sample
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    45  100    45    0     0   3537      0 --:--:-- --:--:-- --:--:--  4500
<html><body><h1>It works!</h1></body></html>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    45  100    45    0     0   3989      0 --:--:-- --:--:-- --:--:--  5625
<html><body><h1>It works!</h1></body></html>
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0<h
tml><body><h1>It works!</h1></body></html>
100    45  100    45    0     0   5819      0 --:--:-- --:--:-- --:--:-- 11250
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0<h
tml><body><h1>It works!</h1></body></html>
100    45  100    45    0     0   4792      0 --:--:-- --:--:-- --:--:--  7500
```

Go to another terminal and check stats same like above

docker stats <container_id>

```
CONTAINER           CPU %               MEM USAGE / LIMIT       MEM %
   NET I/O             BLOCK I/O           PIDS
d1e5535f4054        3526.61%            12.44 MiB / 3.839 GiB   0.32%
    458.4 kB / 405.2 kB   10.29 MB / 0 B      7
```

```
CONTAINER           CPU %               MEM USAGE / LIMIT       MEM %
   NET I/O             BLOCK I/O           PIDS
d1e5535f4054        3062.62%            12.44 MiB / 3.839 GiB   0.32%
    596.2 kB / 527.9 kB   10.29 MB / 0 B      7
```

Like this it will shows stats when the container is running.

tag : Tag an image into a repository

```
root@ubuntu:~# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED
     SIZE
ubuntu                      sivaapache2         6788027f3666        4 days ago
     257.5 MB
httpd                       latest              c24f66af34b4        6 days ago
     177.3 MB
centos                      latest              196e0ce0c9fb        4 weeks ago
     196.6 MB
hello-world                 latest              05a3bd381fc2        5 weeks ago
     1.84 kB
```

Here we have these many images so we can these images and we can make new image with our own name wth tag command.

```
root@ubuntu:~# docker tag
"docker tag" requires exactly 2 argument(s).
See 'docker tag --help'.

Usage:  docker tag IMAGE[:TAG] IMAGE[:TAG]

Tag an image into a repository
root@ubuntu:~# docker tag centos goldentech/mycentos
root@ubuntu:~# docker images
REPOSITORY                  TAG                 IMAGE ID            CREATED
     SIZE
ubuntu                      sivaapache2         6788027f3666        4 days ago
     257.5 MB
httpd                       latest              c24f66af34b4        6 days ago
     177.3 MB
centos                      latest              196e0ce0c9fb        4 weeks ago
     196.6 MB
goldentech/mycentos         latest              196e0ce0c9fb        4 weeks ago
     196.6 MB
hello-world                 latest              05a3bd381fc2        5 weeks ago
     1.84 kB
```

We cant use captal letters as images names only small letters.

I just got goldentech/mycentos but that's not really my image because just copied the existing image so and named my self . Both the image ids are same. Instead of remembering public images and we can keep our names so It is easy for us to remember names. That's the only reason.

update : Update configuration of one or more containers

Beofre starts with update command we will just check the limits of docker.

If you go to run command will have lot of commands to give as a parameters.

```
root@ubuntu:~# docker run --help

Usage:  docker run [OPTIONS] IMAGE [COMMAND] [ARG...]

Run a command in a new container

Options:
      --add-host value              Add a custom host-to-IP mapping (host:ip
efault [])
  -a, --attach value                Attach to STDIN, STDOUT or STDERR (defau
])
      --blkio-weight value          Block IO (relative weight), between 10 a
000
```

Previously we have done stats command right there we have seen some limits means memory allocation has been set .

```
CONTAINER           CPU %               MEM USAGE / LIMIT       MEM %
    NET I/O             BLOCK I/O           PIDS
d1e5535f4054        3526.61%            12.44 MiB / 3.839 GiB   0.32%
    458.4 kB / 405.2 kB   10.29 MB / 0 B      7
```

So here container to use complete memory bydefault, suppose if we have 10 containers and we allow every container to use complete memory as a limit in some cases some container is misbehaving and it can be consumed complete memory so here other 9 containers will get an issue and 9 are going to effected.

That's the reason we put some threshold values usually. So threshold now we want to keep it for memory allocation to a container.

In run command help we can –m parameter to set threshold value for memory allocation.

docker run --help

docker run –itd –m 1g centos (memory can be b bytes m mbs g gbs)

```
root@ubuntu:~# docker run -itd -m 1g centos
WARNING: Your kernel does not support swap limit capabilities, memory limited wi
thout swap.
db1494385caa722b46ea6bd119255149ed8db1df15461b19ac8b94cd96c2fb34
root@ubuntu:~#
```

```
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
  STATUS              PORTS               NAMES
db1494385caa        centos              "/bin/bash"              About a minut
  Up 56 seconds                           pensive_shockley
d42f80f614a5        centos              "/bin/bash"              37 minutes ag
  Up 37 minutes                           berserk_liskov
d1e5535f4054        centos              "/bin/bash"              40 minutes ag
  Up 40 minutes                           tiny_feynman
c8ac99ab2d00        httpd               "httpd-foreground"       2 hours ago
  Up 2 hours          0.0.0.0:80->80/tcp  jovial_agnesi
root@ubuntu:~# docker stats db1494385caa
```

```
CONTAINER           CPU %               MEM USAGE / LIMIT   MEM %
NET I/O             BLOCK I/O           PIDS
db1494385caa        0.00%               1000 KiB / 1 GiB    0.10%
3.299 kB / 866 B    0 B / 0 B           1
```

Now we can the total memory utilization has been set to 1 gb.

Now if we want to update this to 2 gb so here for updations we have to use docker update.

docker update –m 2g <container_id>

docker stats <container _id>

```
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
db1494385caa        centos              "/bin/bash"         20 minutes ago
Up About a minute                       pensive_shockley
root@ubuntu:~# docker update
docker: "update" requires a minimum of 1 argument.
See 'docker update --help'.

Usage:  docker update CONTAINER [CONTAINER...]

Update configuration of one or more containers
root@ubuntu:~# docker update -m 2g db1494385caa
db1494385caa
root@ubuntu:~#
```

docker stats <container_id>

```
CONTAINER           CPU %               MEM USAGE / LIMIT   MEM %
NET I/O             BLOCK I/O           PIDS
db1494385caa        0.00%               5.938 MiB / 2 GiB   0.29%
8.712 kB / 866 B    5.059 MB / 0 B      1
```

Now we can see the updation on memory threshold.

wait Block until a container stops, then print its exit code

It will just display the exited status of the container

docker ps

```
root@ubuntu:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
db1494385caa        centos              "/bin/bash"         22 minutes ago
Up 4 minutes                            pensive_shockley
```

docker wait <container id>

```
goldentech@ubuntu:~$ docker wait db1494385caa
```

go to other terminal

docker stop <container_id>

```
root@ubuntu:~# docker stop db1494385caa
db1494385caa
root@ubuntu:~#
```

Then come to first terminal and check the output here. 137 is the exited status number.

```
goldentech@ubuntu:~$ docker wait db1494385caa
137
goldentech@ubuntu:~$
```

```
goldentech@ubuntu:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED
 STATUS                      PORTS               NAMES
db1494385caa        centos              "/bin/bash"              27 minutes ago
 Exited (137) 40 seconds ago                     pensive_shockley
d42f80f614a5        centos              "/bin/bash"              About an hour ago
 Exited (1) 9 minutes ago                        berserk_liskov
d1e5535f4054        centos              "/bin/bash"              About an hour ago
 Exited (1) 9 minutes ago                        tiny_feynman
c8ac99ab2d00        httpd               "httpd-foreground"       2 hours ago
 Exited (1) 9 minutes ago    0.0.0.0:80->80/tcp  jovial_agnesi
goldentech@ubuntu:~$
```