

1. Installation of Docker Desktop on windows :

[Installation of Docker Desktop on windows](#) ([Web view](#))

Install Docker Desktop on Windows

Estimated reading time: 10 minutes

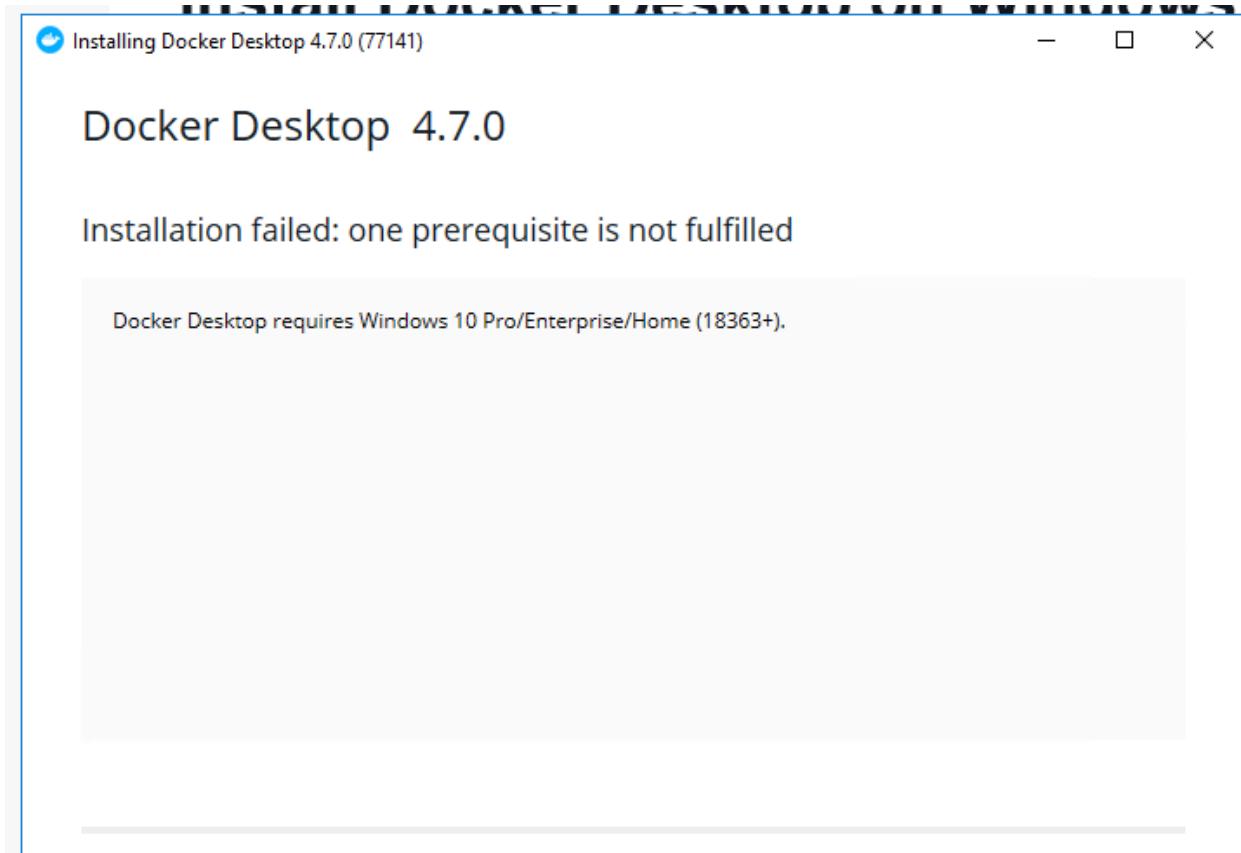
ⓘ Update to the Docker Desktop terms

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than \$10 million USD in annual revenue) now requires a paid subscription. The grace period for those that will require a paid subscription ends on January 31, 2022. [Learn more](#).

Welcome to Docker Desktop for Windows. This page contains information about Docker Desktop for Windows system requirements, download URL, instructions to install and update Docker Desktop for Windows.

ⓘ Download Docker Desktop for Windows

[Docker Desktop for Windows](#)



Name	Description	Status	Startup Type	Log
Docker Desktop Service		Running	Automatic	Local
Downloaded Maps Manager	Windows se...		Automatic (...)	Netw...
Embedded Mode	The Embed...		Manual (Trig...	Loca...
Executing File System (EFS)	Provider th...	Running	Manual (Trig...	Loca...

```
C:\Users\dhgundra>docker --version
Docker version 20.10.14, build a224086
```

```
C:\Users\dhgundra>
```

docker --version

All Docker Command Links : <https://docs.docker.com/engine/reference/commandline/push/>

Good Documents : <https://docs.docker.com/engine/reference/commandline/stop/#examples>

Docker Pull Command : https://hub.docker.com/_/microsoft-windows-servercore-iis

docker pull mcr.microsoft.com/windows/servercore/iis

```
C:\Users\dhgundra>docker pull mcr.microsoft.com/windows/servercore/iis
Using default tag: latest
latest: Pulling from windows/servercore/iis
Digest: sha256:d1821f5d785e5e17f4cb4194525dbcb57b7ec2e819d4db4738c14b6f2f2c2ad0
Status: Image is up to date for mcr.microsoft.com/windows/servercore/iis:latest
mcr.microsoft.com/windows/servercore/iis:latest
```

Docker Commands :

- From
- Label
- Copy
- Entry point
- Run
- CMD
- WORKDIR
- Expose
- Volume
- User

Docker Help :

Ex : docker run --help

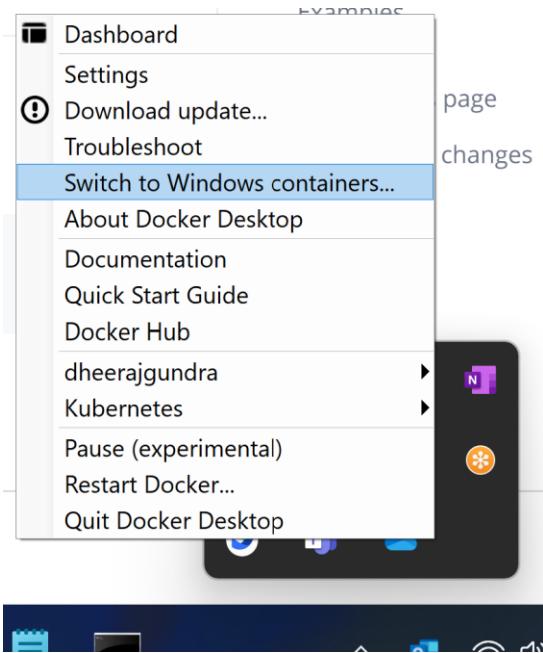
Sample of Windows Docker :

```
-----  
# escape=`  
FROM mcr.microsoft.com/windows/servercore/iis  
RUN powershell -Command `'  
New-Item -Path 'c:\' -Name 'Chandu' -ItemType 'Directory';`  
$pythonurl='https://www.python.org/ftp/python/3.8.1/python-3.8.1-amd64.exe';`  
$output='c:\Chandu\python-3.8.1-amd64.exe';`  
[Net.ServicePointManager]::SecurityProtocol =[Net.SecurityProtocolType]::Tls12;`  
Invoke-WebRequest -Uri $pythonurl -OutFile $output;`  
New-Item -Path 'C:\inetpub\wwwroot\index.html' -ItemType File -Force;`  
Add-Content C:\inetpub\wwwroot\index.html 'Welcome to learning docker and deploying a IIS  
Application';  
  
EXPOSE 80
```

Save the file name as **dockerfile** with no extensions like .txt or others.



If you are running any windows containers, please switch this to windows containers.



Note :

Open Cmd

C:\Users\dhgundra>**docker build -t testiis:v1 c:\docker**

If any error occurs, the result will be like

```
x.html 'Welcome to learning docker and deploying a IIS Application';
--> Running in a1ec66f08d96
The string is missing the terminator: ".
+ CategoryInfo          : ParserError: () [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : TerminatorExpectedAtEndOfString

The command 'cmd /S /C powershell -Command New-Item -Path "c:\" -Name "Chandu" -ItemType "Directory";$pythonurl
="https://www.python.org/ftp/python/3.8.1/python-3.8.1-amd64.exe";$output="c:\Chandu\python-3.8.1-amd64.exe";[Net.ServicePointManager]::SecurityProtocol =[Net.SecurityProtocolType]::Tls12;Invoke-WebRequest-Uri $pythonurl-OutputFile $output;New-Item -Path 'C:\inetpub\wwwroot\index.html' -ItemType File Force;Add-Content C:\inetpub\wwwroot\index.html 'Welcome to learning docker and deploying a IIS Application';' returned a non-zero code: 1
```

If no Errors ,

```
Microsoft Windows [Version 10.0.22000.613]
(c) Microsoft Corporation. All rights reserved.
```

```
C:\Users\dhgundra>docker build -t testiis:v1 c:\docker
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM mcr.microsoft.com/windows/servercore/iis
latest: Pulling from windows/servercore/iis
18ae6bebe21d: Downloading 397.9MB/1.505GB
44a67477f452: Downloading 221.7MB/1.196GB
05fb313bfe29: Download complete
4c76990685f7: Download complete
76b228c93fc3: Download complete
```

```
C:\Users\dhgundra>docker build -t testiis:v1 c:\docker
Sending build context to Docker daemon 2.56kB
Step 1/3 : FROM mcr.microsoft.com/windows/servercore/iis
latest: Pulling from windows/servercore/iis
18ae6bebe21d: Already exists
99cf2a8aa05b: Downloading [=====] 510.4MB/1.23GB
df66a4a51a9f: Verifying Checksum
5f96f7d12fb1: Download complete
9d5dde92beb3: Download complete
```

```
C:\Users\dhgundra>docker build -t testiis:v1 c:\docker
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM mcr.microsoft.com/windows/servercore/iis
latest: Pulling from windows/servercore/iis
18ae6bebe21d: Pull complete
44a67477f452: Pull complete
05fb313bfe29: Pull complete
4c76990685f7: Pull complete
76b228c93fc3: Pull complete
Digest: sha256:cae9cf27d3716ad286dc8d2094f79acd9ea9b72b6e186d8851c8f94529ca482b
Status: Downloaded newer image for mcr.microsoft.com/windows/servercore/iis:latest
--> 89cfa669fda8
Step 2/4 : RUN powershell -Command New-Item -Path 'C:\inetpub\wwwroot\index.html' -ItemType File -Force;Invoke-WebRequest -UseBasicParsing -Uri "https://dotnetbinaries.blob.core.windows.net/servicemonitor/2.0.1.6/ServiceMonitor.exe" -OutFile "C:\ServiceMonitor.exe"
--> Running in a5b63628b2ac
```

What happens ?

The image "mcr.microsoft.com/windows/servercore/iis" is pulled from Docker HUB into your laptop, then the Script in the docker file will execute & it will create a New image in your System. Note, Image is created , but the container is not created.

Docker images

```
C:\Users\dhgundra>docker images
REPOSITORY           TAG      IMAGE ID      CREATED        SIZE
testiis              v1       42dec3726b9c   21 seconds ago  5.74GB
mcr.microsoft.com/windows/servercore/iis    latest   89cfa669fda8   2 days ago   5.73GB
```

How to create a container from Local images :

```
C:\Users\dhgundra>docker run -d -p 8084:80 testiis:v1
9cb0512aaa672960765f9fee0b82a86d5b4403a0c042fac9b8f61a476e75fee9
```

Cmd --> docker run -d -p 801:80 testiis:v1

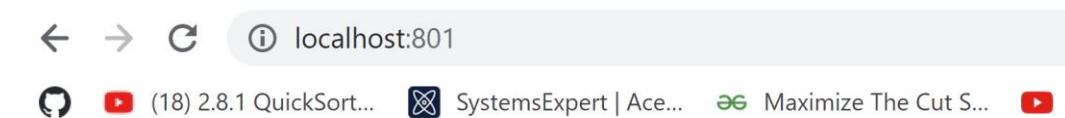
If Version is not defined, then it will auto take Latest

```
C:\Users\dhgundra>docker images
REPOSITORY           TAG      IMAGE ID      CREATED        SIZE
testiis              v1       16de615d247d   3 minutes ago  5.87GB
mcr.microsoft.com/windows/servercore/iis    latest   57b089f4fc4d   4 weeks ago   5.81GB

C:\Users\dhgundra>docker run -d -p 801:80 testiis:v1
8320a9da1a16acd2b2d9259ce+6f3e6cfe4d5685bb95b45a6dc1cf7c3a3961bd

C:\Users\dhgundra>docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED        STATUS        PORTS     NAMES
8320a9da1a16   testiis:v1   "C:\\ServiceMonitor.e..."  47 seconds ago   Up 43 seconds  0.0.0.0:801->80/tcp  distracte_hertz
```

After container is created, the Check the Application



To find out how many Containers are running in --> docker ps

```
C:\Users\dhgundra>docker ps
CONTAINER ID   IMAGE          COMMAND       CREATED        STATUS        PORTS     NAMES
8320a9da1a16   testiis:v1   "C:\\ServiceMonitor.e..."  5 minutes ago   Up 5 minutes  0.0.0.0:801->80/tcp  distracte_hertz

C:\Users\dhgundra>
```

To get the Full properties of Container like network, host & log details :

CONTAINER ID	IMAGE	COMMAND	CREATED
NAMES			
cb1ddd6a04b8	testapp	"C:\\ServiceMonitor.e..."	46 hours ago

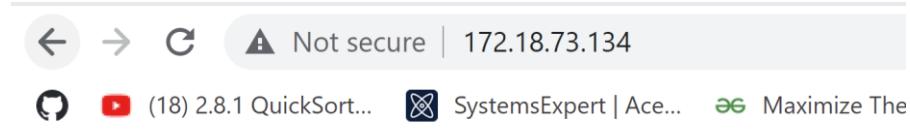
To find the Container IP Address : **docker inspect 8320a9da1a16**

C:\Users\dhgundra>docker inspect 8320a9da1a16

```

"Networks": {
    "nat": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID": "ccbcfd01ec4af3bb04682975b2d5eafa6099fb047e3c143c6dfc7cac1b6c277",
        "EndpointID": "ac49a23555f7639b9a9e66d2e85feb8dbe187d9d0c42515d4e41c9e884c93292",
        "Gateway": "172.22.144.1",
        "IPAddress": "172.22.153.5",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "00:15:5d:b2:cc:b7",
        "DriverOpts": null
    }
}

```



Welcome to learning docker and deploying a IIS Application

Login into POD(Container) : **docker exec -ti 8320a9da1a16 Powershell**

C:\Users\dhgundra>docker exec -ti cb1ddd6a04b8 PowerShell

Exit the Container Session :

PS C:\inetpub\wwwroot> **exit**

Get Container Process details :

```
docker top 8baef63aac71
```

Stop a Container : docker stop 8320a9da1a16

```
c:\docker\linux>docker stop 8320a9da1a16  
8320a9da1a16
```

Again Start a Container : docker start 8320a9da1a16

```
c:\docker\linux>docker start 8320a9da1a16  
8320a9da1a16
```

Delete a container :

```
C:\Users\dhgundra>docker rmi -f f69c910adab5
```

Delete a Docker image : docker rmi -f cc04e4b5dd0a

```
C:\Users\dhgundra>docker rmi -f 42dec3726b9c  
Untagged: testiis:v1  
Deleted: sha256:42dec3726b9c521ef4d034b7100d6564771cacf9879ad45f6dcde840a47c76e9  
Deleted: sha256:3720bf2415515e80e11ad091c6a8eb929f08c444f94478fa96657ff86980a1d9  
Deleted: sha256:3c141db94407c7e843e487df71a742fdd1717e6d5ef29a9dac2e9467d0194cd9  
Deleted: sha256:c15d516a1f297fd1049f9a459e71256a4b9244c68c5697ee655184df4e65f36d  
Deleted: sha256:2ac9e9c13bf939134416ec404bda4eff7f80bb2a777443e4653e77584c993d3b  
Deleted: sha256:77857b8d38d4608d1ab6f6079794f0354bad08347765a59ce59528b04d7dcc13
```

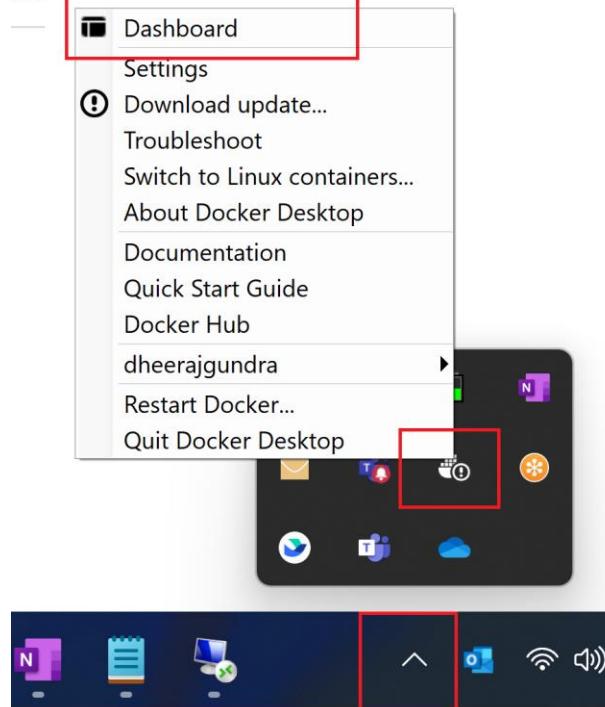
Delete with -Force : **docker rmi 89cfaf669fda8 -f**

```
C:\Users\dhgundra>docker rmi 89cfaf669fda8 -f
```

GUI Dashboard :

You can also get same information from this Dashboard.

The



A screenshot of the Docker Images dashboard. On the left, there are navigation links for Images, Volumes, and Dev Environments (with a PREVIEW button). The main area is titled "LOCAL REMOTE REPOSITORIES". It features a search bar and an "In Use only" checkbox. A table lists local images:

NAME ↑	TAG	IMAGE ID	CREATED	SIZE
mcr.microsoft.com/window...	latest	57b089f4fc4d	about 1 month ago	5.81 GB
testiis	IN USE v1	16de615d247d	34 minutes ago	5.87 GB

A screenshot of the Docker image details view for "testiis". The image size is listed as 5.87 GB. To the right, there are three buttons: a blue square with three dots ("..."), a blue button with a white play arrow labeled "RUN", and a blue button with a white square labeled "Inspect". Below these are four options: "Pull", "Push to Hub", and "Remove" (in red text).

Docker for Linux :

Installation : yum install docker -y

```
[root@ip-172-31-25-208 ec2-user]# yum install docker
```

```
Installed:
  docker.x86_64 0:20.10.17-1.amzn2

Dependency Installed:
  containerd.x86_64 0:1.6.6-1.amzn2
  pigz.x86_64 0:2.3.4-1.amzn2.0.1
  libcgroup.x86_64 0:0.41-21.amzn2
  runc.x86_64 0:1.1.3-1.amzn2

Complete!
```

service docker status

Systemctl start docker

Systemctl enable docker

```
[root@ip-172-17-0-205 ec2-user]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
  Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: d
isabled)
  Active: inactive (dead)
    Docs: https://docs.docker.com
[root@ip-172-17-0-205 ec2-user]#
```

service docker start

```
[root@ip-172-17-0-205 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
[root@ip-172-17-0-205 ec2-user]# █
```

You can pull the images directly. Check https://hub.docker.com/_/httpd

Docker pull httpd

```
[root@ip-172-31-25-208 /]# docker pull httpd█
```

```
[root@ip-172-31-25-208 /]# docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
Digest: sha256:70999c4a17c796dd28f86f9c847b30f28abaed6ef1fd72a44282b1c941238804
Status: Image is up to date for httpd:latest
docker.io/library/httpd:latest
```

```
[root@ip-172-31-25-208 /]# docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
<none>          <none>    1f28f0aa6ac1  11 minutes ago  231MB
<none>          <none>    73003ec14684  36 minutes ago  114MB
ubuntu          latest    2dc39ba059dc  8 days ago   77.8MB
httpd           latest    a981c8992512  2 weeks ago   145MB
centos          latest    5d0da3dc9764  11 months ago  231MB
```

```
[ ]# docker run -d -p 801:80 httpd█
```

docker run -d -p 801:80 httpd

Docker ps

```
[root@ip-172-31-25-208 /]# docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS
NAMES
8baef63aac71    httpd      "httpd-foreground"  5 minutes ago  Up 5 minutes  0.0.0.0:801->80/tcp
, :::801->80/tcp      jovial robinson
```

Docker ps -a



It works!

How to change the page in Apache 2

Login to the container.

```
docker exec -it bfb772364272 bash  
[root@ip-172-17-0-50 ec2-user]# docker exec -it bfb772364272 bash
```

Find HTML file :

```
root@bfb772364272:/usr/local/apache2# find -name '*.html'  
  
find -name '*.html'  
cd htdocs/  
Cat index.html  
cat -> index.html  
root@bfb772364272:/usr/local/apache2/htdocs# cat -> index.html  
Hello man..this is a page...
```

Enter Control + C

Restart the Apache by stop & start the container :

```
[root@ip-172-17-0-50 ec2-user]# docker stop bfb772364272  
bfb772364272  
[root@ip-172-17-0-50 ec2-user]# docker start bfb772364272  
bfb772364272
```

Another Example :

```
docker pull ubuntu
```

Docker pull centos

Interactive Mode :

```
docker run -it ubuntu
```

```
[root@ip-172-17-0-50 ec2-user]# docker run -it ubuntu
root@400fb7e1aacd:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root  sbin  sys  usr
root@400fb7e1aacd:/#
```

Another way :

```
[root@ip-172-17-0-50 ec2-user]# docker exec -it bfb772364272 bash
```

```
root@bfb772364272:/usr/local/apache2# ls -a
. .. bin build cgi-bin conf error htdocs icons include logs modules
root@bfb772364272:/usr/local/apache2# uname -r
5.10.144-127.601.amzn2.x86_64
```

To find OS details : uname -r

To exit Container : exit

```
root@400fb7e1aacd:/# exit
exit
[root@ip-172-17-0-50 ec2-user]#
```

To start the container :

Docker start <containerid>

What is docker attach?

Docker stats

```
[ser]# docker stats
```

Stats Output :

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %	NET I/O	BLOCK I/O	PIDS
89165716b301	charming_taussig	0.00%	7.941MiB / 7.76GiB	0.10%	4.42kB / 2.41kB	0B / 0B	109

Docker system df

```
[root@ip-172-17-0-50 ec2-user]# docker system df
TYPE          TOTAL      ACTIVE      SIZE      RECLAMABLE
Images         2          2          223MB      0B (0%)
Containers     3          1          10B        8B (80%)
Local Volumes  0          0          0B         0B
Build Cache    0          0          0B         0B
[root@ip-172-17-0-50 ec2-user]#
```

If you want to create a container with custom name

Docker run --name kumarubuntu -it ubuntu

```
[root@ip-172-17-0-50 ec2-user]# docker run --name kumarubuntu -it ubuntu
```

Once, we run this command, we will be inside the container...

****Most dangerous command

Docker system prune -a

```
[root@ip-172-17-0-50 ec2-user]# docker system prune -a
WARNING! This will remove:
 - all stopped containers
 - all networks not used by at least one container
 - all images without at least one container associated to them
 - all build cache
```

Top get process details of containers

```
[root@ip-172-31-25-208 /]# docker top 8baef63aac71
UID          PID    PPID      C
TTY           TIME   CMD
root          6095   6071      1
              ?        00:00:00  httpd -DFOREGROUND
33            6123   6095      0
              ?        00:00:00  httpd -DFOREGROUND
33            6124   6095      0
              ?        00:00:00  httpd -DFOREGROUND
33            6125   6095      0
              ?        00:00:00  httpd -DFOREGROUND
```

Get all the process list in linux : ps -aux

```
[root@ip-172-31-25-208 /]# ps -aux
```

```
root      5777  0.0  0.0      0      0 ?          1  09:17  0:00 [kworker/0:4-eve]
root      6051  0.0  0.3 1164232 3088 ?          Sl  09:20  0:00 /usr/bin/docker-proxy -proto tc
root      6056  0.0  0.3 1164232 3068 ?          Sl  09:20  0:00 /usr/bin/docker-proxy -proto tc
root      6071  0.0  0.9 712208 8996 ?          Sl  09:20  0:00 /usr/bin/containerd-shim-runc-v
root      6095  0.3  0.4  5940  4652 ?          Ss  09:20  0:00 httpd -DFOREGROUND
33       6123  0.0  0.3 807160 3516 ?          Sl  09:20  0:00 httpd -DFOREGROUND
33       6124  0.0  0.3 807160 3516 ?          Sl  09:20  0:00 httpd -DFOREGROUND
33       6125  0.0  0.3 807160 3516 ?          Sl  09:20  0:00 httpd -DFOREGROUND
root      6238  0.0  0.3 162320 3904 pts/0   R+  09:21  0:00 ps -aux
root      6541  0.0  4.4 1369764 44084 ?         Ssl 08:13  0:03 /usr/bin/containerd
root      6552  0.1  7.2 1437300 71784 ?         Ssl 08:13  0:08 /usr/bin/dockerd -H fd:// --con
```

```
[root@ip-172-31-25-208 /]# docker stop 8baef63aac71
8baef63aac71
```

After stopping the container, all the processes die off.

root	5565	0.0	0.0	0	0	?	I	08:50	0:00
root	5633	0.0	0.0	0	0	?	I	09:10	0:00
root	5717	0.0	0.0	0	0	?	I	09:15	0:00
root	5777	0.0	0.0	0	0	?	I	09:17	0:00
root	6292	0.0	0.0	0	0	?	I	09:22	0:00
root	6293	0.0	0.3	162320	3856	pts/0	R+	09:22	0:00
root	6541	0.0	4.4	1369764	44084	?	Ssl	08:13	0:03
root	6552	0.1	7.2	1437300	71788	?	Ssl	08:13	0:08
root	6572	0.0	0.0	0	0	?	I	08:13	0:00
root	6573	0.0	0.0	4228	740	?	S	08:13	0:00
root	7961	0.0	0.0	0	0	?	I	08:45	0:00

Docker Logs :

Docker logs -f <container ID>

```
[root@ip-172-17-0-205 ec2-user]# docker logs -f 72cf73fa6c1d
AH00558: httpd: Could not reliably determine the server's fully qualified domain name
, using 172.18.0.2. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name
, using 172.18.0.2. Set the 'ServerName' directive globally to suppress this message
[Fri Sep 16 10:02:12.295261 2022] [mpm_event:notice] [pid 1:tid 140345411677504] AH00
489: Apache/2.4.54 (Unix) configured -- resuming normal operations
[Fri Sep 16 10:02:12.295446 2022] [core:notice] [pid 1:tid 140345411677504] AH00094:
Command line: 'httpd -D FOREGROUND'
167.220.238.142 - - [16/Sep/2022:10:18:20 +0000] "GET / HTTP/1.1" 200 45
167.220.238.142 - - [16/Sep/2022:10:18:21 +0000] "GET /favicon.ico HTTP/1.1" 404 196
167.220.238.142 - - [16/Sep/2022:10:19:11 +0000] "-" 408 -
167.220.238.142 - - [16/Sep/2022:10:29:48 +0000] "GET / HTTP/1.1" 304 -
167.220.238.142 - - [16/Sep/2022:10:29:52 +0000] "GET / HTTP/1.1" 304 -
167.220.238.142 - - [16/Sep/2022:10:29:54 +0000] "GET / HTTP/1.1" 304 -
167.220.238.142 - - [16/Sep/2022:10:29:57 +0000] "GET / HTTP/1.1" 304 -
167.220.238.142 - - [16/Sep/2022:10:30:23 +0000] "GET /dd HTTP/1.1" 404 196
167.220.238.142 - - [16/Sep/2022:10:30:34 +0000] "GET / HTTP/1.1" 304 -
167.220.238.142 - - [16/Sep/2022:10:30:37 +0000] "GET / HTTP/1.1" 304 -
```

Important Points :

```
[root@ip-172-17-0-205 ec2-user]# docker pull nanoserver/iis
Using default tag: latest
latest: Pulling from nanoserver/iis
bce2fbc256ea: Downloading
e7a598ed389e: Downloading
330fc9a0a84f: Download complete
8accfc5d2068: Waiting
ce0579b0380c: Waiting
363f5ae71fe9: Waiting
bdadd5a88cb5: Waiting
73f9c45f3462: Waiting
image operating system "windows" cannot be used on this platform
[root@ip-172-17-0-205 ec2-user]#
```

yes, it will not work, as container uses kernel space from host os
So host os for windows container should be windows

CloudWatch logs with Dockers :

```
docker run --name nginx --log-driver="awslogs" --log-opt awslogs-region=us-east-1 --log-opt
awslogs-group="myapp/dev" --log-opt awslogs-stream="myapp-log-stream" -p 80:80 --restart
unless-stopped -id nginx:latest
```

#Run Docker outside AWS & push logs to AWS.

```
#aws configure
```

```
edit /lib/systemd/system/docker.service file.
```

```
#In [Service] section add:
```

```
#[Service]
#Environment=AWS_SHARED_CREDENTIALS_FILE=<path_to_aws_credentials_file>

#
#Environment=AWS_SHARED_CREDENTIALS_FILE=/root/.aws/credentials
#systemctl daemon-reload
#systemctl restart docker
#
```

```
#docker run --log-driver=awslogs \
```

```
#--log-opt awslogs-region=us-east-1 \
#--log-opt awslogs-group=cloudgeeks \
#--log-opt awslogs-create-group=true \
```

Basic Docker Networks

Docker network ls

```
[root@ip-172-17-0-205 ec2-user]# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
b623e2b313e4    bridge    bridge      local
10890ef98626    host      host       local
d8d873d588db    none      null       local
[root@ip-172-17-0-205 ec2-user]#
```

Types of Virtual Networks :

1. Bridge Network --> Default

Ans : When you create a containers, a default Bridge is created with some subnet & all the containers are part of that bridge. 99%

Bridge is the default network in docker which is also called as docker0.

Command : docker network inspect bridge

1. Host Network :

Ans : If you need to Attach the Host network(Docker Server) to the container, then we use Host network. 1%. We may get port binding issues.

1. None Network : Self isolated network. So, No communication is established. 0%.

1. Overlay Network : If you to communicate containers from one Docker Host to Other Docker Host, then this OverLay Network is used. Genrally, Container communicate within same Docker host.

Container communicate to each other with Namespaces not with ip Address. Whenever you create a Containers, a default DNS in 127.0.0.11 is created & it will maintain the Ip resolutions.

How Communication is happening between Docker container & Docker Host(Servers)??

Physical network & Virtual network communication is done with NAT. This will happen by default.
When every Virtual Network is created, Automatically NAT is created
NAT is called as ip Mask reading.

```
docker inspect my_container
```

```
Docker network ls
```

```
[root@ip-172-17-0-50 ec2-user]# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
9792b247f2e8    bridge    bridge      local
1578695db61a   host      host       local
56272fb49501   none      null       local
```

```
docker exec -it my_container ping -w3 google.com
```

```
Docker network inspect bridge
```

```
Docker network inspect host
```

```
Docker network inspect none
```

```
[root@ip-172-17-0-205 ec2-user]# docker network inspect none
[{"Name": "none",
 "Id": "d8d873d588dbdf4f0b2950de37196dd9972ac5c7245737f8fbf282538f1b357b",
 "Created": "2022-09-16T09:57:42.864257533Z",
 "Scope": "local",
 "Driver": "null",
 "EnableIPv6": false,
 "IPAM": {
     "Driver": "default",
     "Options": null,
     "Config": []
 },
 "Internal": false,
 "Attachable": false,
 "Ingress": false,
 "ConfigFrom": {
     "Network": ""
 },
 "ConfigOnly": false,
 "Containers": {},
 "Options": {},
 "Labels": {}
}]
```

To create a network

Docker network create <Name of network> --> this will create a network with default(bridge)

```
[root@ip-172-17-0-50 ec2-user]# docker network create testnetork
c10dc9a25e13592a04ea22282ac0f8201fa00e0a8e0c7abdec4d4c91fd045d3f
[root@ip-172-17-0-50 ec2-user]# Docker network inspect
bash: Docker: command not found
[root@ip-172-17-0-50 ec2-user]# Docker network ls
bash: Docker: command not found
[root@ip-172-17-0-50 ec2-user]# docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
9792b247f2e8    bridge    bridge      local
1578695db61a    host      host      local
56272fb49501    none      null      local
c10dc9a25e13    testnetork  bridge      local
```

Docker network inspect testnetwork

If you want to create Custom network :

```
Docker network create --driver bridge --subnet 17.30.0.0/24 <name>
```

```
Docker network create --driver bridge --subnet 17.30.0.0/24 customtestnetwork
```

```
[root@ip-172-17-0-50 ec2-user]# docker network create --driver bridge --subnet 17.30.0.0/24 customtestnetwork  
ff0650b16b55d3fe080a1d16d98b174dfdd25950cbc3a3fbe0334a490262bb4b  
[root@ip-172-17-0-50 ec2-user]# docker network ls  
NETWORK ID      NAME      DRIVER      SCOPE  
9792b247f2e8    bridge    bridge      local  
ff0650b16b55    customtestnetwork    bridge      local  
1578695db61a    host      host      local  
56272fb49501    none      null      local  
c10dc9a25e13    testnetork    bridge      local
```

Now, you can create container with this network...

```
docker run -d --name test --network customtestnetwork nginx
```

Use inspect command to check for the network...

If you want to connect this container to other network,(2 networks)

```
--> Docker network connect testnetork nginx
```

To disconnect

```
Docker network disconnect testnetork nginx
```

Create a image in Linux :

```
FROM node:12-alpine  
RUN apk add --no-cache python2 g++ make  
WORKDIR /app  
COPY ..  
RUN yarn install --production  
CMD ["node", "src/index.js"]  
EXPOSE 3000
```

```
Cd docker  
Touch dockerfile  
Vi dockerfile  
Save it as dockerfile
```

```
docker build -t testimage /home/ec2-user/docker
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
testimage	latest	86e7a40a5dc6	23 seconds ago	314MB
<none>	<none>	e3086f47f823	17 minutes ago	280MB
nginx	latest	76c69feac34e	3 days ago	142MB
httpd	latest	d16a51d08814	3 weeks ago	145MB
node	12-alpine	bb6d28039b8c	6 months ago	91MB
centos	latest	5d0da3dc9764	13 months ago	231MB

Docker Push to Docker HUB :

First Login to Docker :

```
C:\Users\dhgundra>docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, h
  https://hub.docker.com to create one.
Username: dheerajgundra
>Password:
>Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://d
com/go/access-tokens/
```

Call Docker Commands using API's :

Docker Volumes :

Why do we need Docker Volumes ?

Container are temp storage & when the Server is Stopped/restarted, the Container is gone. To have Container permanently, we can store in docker volumes.s

For login with Tokens : <https://docs.docker.com/docker-hub/access-tokens/>

New Access Token

A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. [Learn more](#)

Access Token Description *

Test

Access permissions

Read, Write, Delete



Read, Write, Delete tokens allow you to manage your repositories.

[Cancel](#)

[Generate](#)

Copy Access Token

When logging in from your Docker CLI client, use this token as a password. [Learn more](#)

ACCESS TOKEN DESCRIPTION

readWrite

ACCESS PERMISSIONS

Read & Write

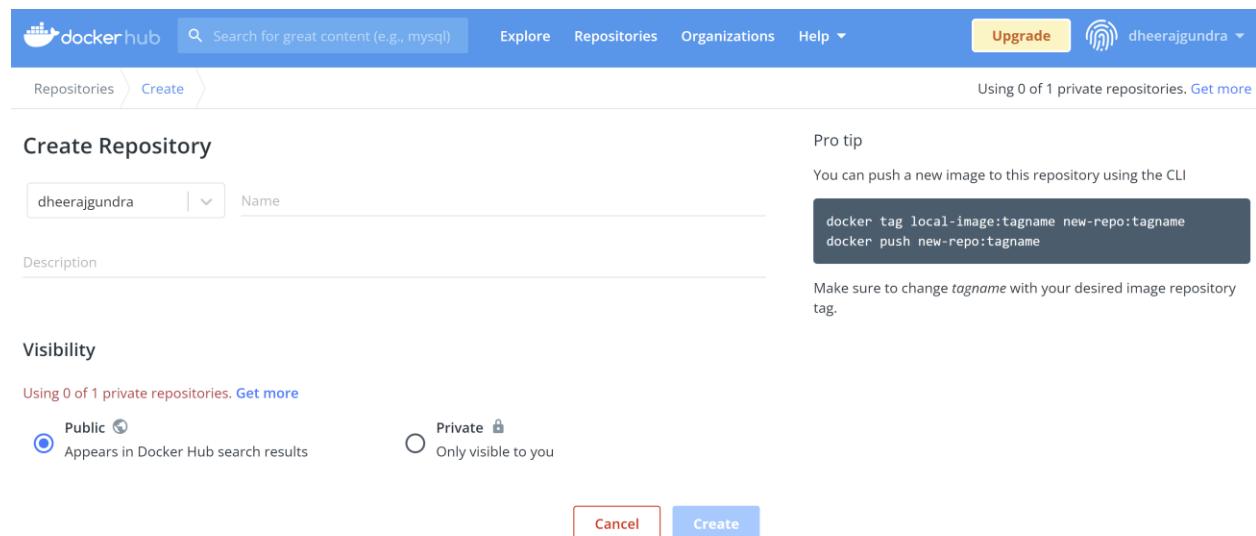
To use the access token from your Docker CLI client:

1. Run `docker login -u dheerajgundra`
2. At the password prompt, enter the personal access token.

```
18d9fc54-398a-4ce0-8ebc-1461036c584d
```



Create a Repo :



The screenshot shows the Docker Hub interface for creating a new repository. At the top, there's a search bar and navigation links for Explore, Repositories, Organizations, Help, Upgrade, and user profile. Below the header, it says "Using 0 of 1 private repositories. [Get more](#)". The main form is titled "Create Repository" and has fields for "Name" (set to "dheerajgundra") and "Description". A "Pro tip" box contains the command "docker tag local-image:tagname new-repo:tagname" and "docker push new-repo:tagname". In the "Visibility" section, there are two options: "Public" (selected) and "Private" (unselected). The "Public" option includes a note that it "Appears in Docker Hub search results". At the bottom right are "Cancel" and "Create" buttons.

110 up

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname  
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

Docker commands

[Public View](#)

To push a new tag to this repository,

```
docker push dheerajgundra/testmyapp:tagname
```

To PUSH a image to DockerHUB

```
C:\Users\dhgundra>docker tag testapp:latest dheerajgundra/testmyapp:latest
```

```
docker tag testapp:latest dheerajgundra/testmyapp:latest
```

Then check if the image is created.

```
C:\Users\dhgundra>docker images  
REPOSITORY                                TAG      IMAGE ID      CREATED       SIZE  
dheerajgundra/testmyapp                    latest   27903dd1089b  45 hours ago  5.74GB
```

```
C:\Users\dhgundra>docker push dheerajgundra/testmyapp:latest  
The push refers to repository [docker.io/dheerajgundra/testmyapp]  
ea5fb1e35023: Pushing [=====>] 54.27kB  
0ee44f71244d: Pushing [=====>] 5.455MB  
7ddc2ce2d9f8: Pushing [=====>] 54.27kB  
0994e1d3f959: Pushing [=====>] 54.27kB  
754ec456f8ba: Pushing [====>] 9.233MB/113.5MB  
d00f903a379e: Waiting  
86b179761af9: Waiting
```

Docker push dheerajgundra/testmyapp:latest

```
C:\Users\dhgundra>docker push dheerajgundra/testmyapp:latest
The push refers to repository [docker.io/dheerajgundra/testmyapp]
ea5fb1e35023: Pushed
0ee44f71244d: Pushed
7ddc2ce2d9f8: Pushed
0994e1d3f959: Pushed
754ec456f8ba: Pushed
d00f903a379e: Skipped foreign layer
86b179761af9: Skipped foreign layer
latest: digest: sha256:eac8eaa4caf5ac448a3e544858cc41ba3d22c1f20c850ceaa939fd61c1abe7da size: 2149
```

dheerajgundra / testmyapp

This repository does not have a description



 Last pushed: a minute ago

Tags and Scans

 VULNERABILITY SCANNING - DISABLED

[Enable](#)

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
 latest		---	a minute ago

[See all](#)



dheerajgundra/testmyapp:latest

DIGEST: sha256:eac8eaa4caf5ac448a3e544858cc41ba3d22c1f20c850ceaa939fd61c1abe7da

OS/ARCH
windows/amd64

COMPRESSED SIZE ⓘ
2.54 GB

LAST PUSHED
2 minutes ago by [dheerajgundra](#)

IMAGE LAYERS ?		Command
1	Apply image 20H2-RTM-amd64	1.4 GB
2	Install update 20H2-amd64	1.11 GB
3	cmd /S /C powershell -Command	23.9 MB
4	cmd /S /C #(nop)	1.26 KB
5	cmd /S /C #(nop)	1.26 KB
6	cmd /S /C powershell -Command	378.17 KB
7	cmd /S /C #(nop)	1.3 KB

Docker Compose :

What is docker compose?

If you want to run multiple containers , it is not always recommended to run each & every commands again & again. So, for multiple complex container creations, we will create a Docker Compose file, which Will run create multiple containers at one shot.

#Docker Compose Installation

```
curl -L "https://github.com/docker/compose/releases/download/1.28.5/docker-compose-\$\(uname -s\)-\$\(uname -m\)" -o /usr/local/bin/docker-compose
```

```
chmod +x /usr/local/bin/docker-compose
```

```
ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

```
docker-compose --version
```

How to Create Docker Compose File :

Mkdir dockercomposefolder

Cd dockercomposefolder

Touch docker-compose.yml

Vi docker-compose.yml

```
services:  
  database:  
    image: redis  
  web:  
    image: nginx  
version: '2.1'
```

To check the Version , Use --> Docker-compose version

Based on the Version & add Version value.

Run this command --> Docker-compose up -d

Docker Base important Networking :

<https://docs.docker.com/config/containers/container-networking/>

By default, a container inherits the DNS settings of the host, as defined in the `/etc/resolv.conf` configuration file. Containers that use the default `bridge` network get a copy of this

file, whereas containers that use a [custom network](#) use Docker's embedded DNS server, which forwards external DNS lookups to the DNS servers configured on the host.

From <<https://docs.docker.com/config/containers/container-networking/>>

Please add Docker Compose Samples & add more details :

Please work on Docker Push to AWS ECR :

Kubernet installation :

Create a ubuntu Master Server with min 2 Core & 4 GN ram

Paste the Bash Script in User-data or run manully :

```
#To Install Prerequisite packages
```

```
apt-get update && apt-get install -y apt-transport-https  
sudo apt update  
sudo apt -y install curl apt-transport-https
```

```
#To add K8S repository
```

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -  
echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
```

```
#To install Docker
```

```
apt-get install docker.io -y  
sudo apt update
```

```
#To install tools like kubeadm,kubelet,kubectl and Kubernetes CNI
```

```
apt-get install -y kubelet kubeadm kubectl kubernetes-cni
```

```
#To access K8S you may get an error, to resolve that error use this CMD's
```

```
cat > /etc/docker/daemon.json <<EOF
```

```
{  
  "exec-opts": ["native.cgroupdriver=systemd"],  
  "log-driver": "json-file",  
  "log-opt": {
```

```

    "max-size": "100m"
},
"storage-driver": "overlay2",
"storage-opts": [
    "overlay2.override_kernel_check=true"
]
}
EOF

mkdir -p /etc/systemd/system/docker.service.d
systemctl daemon-reload
systemctl restart docker

```

From <<https://github.com/chandrahas20/Meterials/blob/main/kuebeadm-kubernetes-cluster-AWS>>

Sample History File :

```

root@ip-172-17-0-171:/home/ubuntu# history

5 apt-get update && apt-get install -y apt-transport-https
6 sudo apt update
7 sudo apt -y install curl apt-transport-https
8 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
9 echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list
10 apt-get install docker.io -y
11 sudo apt update
12 apt-get install -y kubelet kubeadm kubectl kubernetes-cni
13 cat > /etc/docker/daemon.json <<EOF
{
    "exec-opts": ["native.cgroupdriver=systemd"],
    "log-driver": "json-file",
    "log-opts": {
        "max-size": "100m"
    },
    "storage-driver": "overlay2",
    "storage-opts": [
        "overlay2.override_kernel_check=true"
    ]
}
EOF

14 mkdir -p /etc/systemd/system/docker.service.d
15 systemctl daemon-reload
16 systemctl restart docker
17 clear

```

```

18 kubeadm init
19 clear
20 history
21 clear
22 mkdir -p $HOME/.kube
23 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
24 sudo chown $(id -u):$(id -g) $HOME/.kube/config
25 mkdir -p $HOME/.kube
26 sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
27 sudo chown $(id -u):$(id -g) $HOME/.kube/config
28 kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=\$\(kubectl version | base64 | tr -d '\n'\)"
29 clear
-----
```

NOTE : For Master Server, we need to execute all the above Commands

For Node Server, we need to only execute until 16 line .. Sample :

Sample History commands for Node :

```

-----  

4 apt-get update && apt-get install -y apt-transport-https  

5 sudo apt update  

6 sudo apt -y install curl apt-transport-https  

7 curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -  

8 echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee  

/etc/apt/sources.list.d/kubernetes.list  

9 apt-get install docker.io -y  

10 sudo apt update  

11 apt-get install -y kubelet kubeadm kubectl kubernetes-cni  

12 cat > /etc/docker/daemon.json <<EOF  

{  

  "exec-opts": ["native.cgroupdriver=systemd"],  

  "log-driver": "json-file",  

  "log-opt": {  

    "max-size": "100m"  

  },  

  "storage-driver": "overlay2",  

  "storage-opt": [  

    "overlay2.override_kernel_check=true"  

  ]  

}  

EOF  

13 mkdir -p /etc/systemd/system/docker.service.d  

14 systemctl daemon-reload
```

```
15 systemctl restart docker
24 kubeadm join 172.17.0.180:6443 --token kwqx67.2ynasoyl420nu6oa --discovery-token-ca-cert-
hash sha256:a3731a9cd9774839c2e53230cfb6dd77ec4a136958e956c9f4fdb16f3e53cf1f
25 hostname
26 history
-----

```

Do the Same Commands in Other non-Master Servers also.

To join Node to Master :

Ex : kubeadm join 172.17.0.180:6443 --token kwqx67.2ynasoyl420nu6oa --discovery-token-ca-cert-hash
sha256:a3731a9cd9774839c2e53230cfb6dd77ec4a136958e956c9f4fdb16f3e53cf1f

```
root@ip-172-17-0-171:/home/ubuntu# kubeadm join 172.17.0.180:6443 --token kwqx67.2ynasoyl420nu6oa --discovery-t
oken-ca-cert-hash sha256:a3731a9cd9774839c2e53230cfb6dd77ec4a136958e956c9f4fdb16f3e53cf1f
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
W0417 10:36:43.541805 16950 utils.go:69] The recommended value for "resolvConf" in "KubeletConfiguration" is:
/run/systemd/resolve/resolv.conf; the provided value is: /run/systemd/resolve/resolv.conf
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-flags.env"
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

Go to Master Server & type "Kubectl get nodes" (Check if the Node is joined to the Master .)

```
root@ip-172-17-0-180:/home/ubuntu# kubectl get nodes
NAME           STATUS   ROLES      AGE     VERSION
ip-172-17-0-171   Ready    <none>    72s    v1.23.5
ip-172-17-0-180   Ready    control-plane,master  14m    v1.23.5
root@ip-172-17-0-180:/home/ubuntu# █
```

Refer Link :

<https://usermanual.wiki/Pdf/Painless20Docker20Basic20Edition20A20Practical20Guide20to20Master20Docker20and20its20Ecosystem20Based20on20Real20World20Examples.708067721/html#pfbd>
<https://github.com/quickbooks2018/aws/blob/master/docker-cloud-watch-logging-driver/docker-run-command.sh>

<https://www.youtube.com/c/AWSLinuxWindows/featured>