

# Docker

By

Keshav Kummari

# Launch EC2 instance & Connect from Laptop using Cmdr/Putty

```
D:\Users\chikunman\Downloads\cmden
λ cd ../

D:\Users\chikunman\Downloads
D:\Users\chikunman\Downloads
λ pwd
D:\Users\chikunman\Downloads

D:\Users\chikunman\Downloads
λ ls -lnt docker-aws.pem
-rw-r--r-- 1 chikunman 1649089 1696 Jul 18 17:49 docker-aws.pem

D:\Users\chikunman\Downloads
λ chmod 400 docker-aws.pem

D:\Users\chikunman\Downloads
λ ls -lnt docker-aws.pem
-r--r--r-- 1 chikunman 1649089 1696 Jul 18 17:49 docker-aws.pem

D:\Users\chikunman\Downloads
λ ssh -i "docker-aws.pem" ec2-user@ec2-13-127-182-12.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-127-182-12.ap-south-1.compute.amazonaws.com (13.127.182.12)' can't be established.
ECDSA key fingerprint is SHA256:MWGX0Zn3d4yvt1t6GdgZ/1HvtpHf50iixMXids+5KwK.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-13-127-182-12.ap-south-1.compute.amazonaws.com,13.127.182.12' (ECDSA) to the list of known hosts.

cloud-aws-ec2-13-127-182-12.ap-south-1.compute.amazonaws.com
cloud-aws-ec2-13-127-182-12.ap-south-1.compute.amazonaws.com

1. Connect to your instance using its Public DNS:

ec2-13-127-182-12.ap-south-1.compute.amazonaws.com

Example:

ssh -i "docker-aws.pem" ec2-user@ec2-13-127-182-12.ap-south-1.compute.amazonaws.com
```

## Create a Docker Repository

---

```
[dockerrepo]
name=Docker Repository
baseurl=https://yum.dockerproject.org/repo/main/centos/7/
enabled=1
gpgcheck=1
gpgkey=https://yum.dockerproject.org/gpg
```

# Repo

```
$ cat docker.repo
```

```
[dockerrepo]
```

```
name=Docker Repository
```

```
baseurl=https://yum.dockerproject.org/repo/main/centos/7/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://yum.dockerproject.org/gpg
```

## Download, Install & Configure Docker

```
# yum update
```

```
# yum install -y docker-engine
```

```
# systemctl enable docker
```

```
# systemctl start docker
```

```
# systemctl status docker
```

```
# docker -version
```

```
1.12.1
```

```
# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
------------	-----	----------	---------	------

## Docker Configuration

```
# cd ~
```

**Go to docker path:**

```
# cd /var/run
```

**Check the docker sockets:**

```
# ls -al dock*
```

**Check the docker group info :**

```
# cat /etc/group | grep docker
```

**Add the user in docker group:**

```
# usermod -a -G docker <user_name>
```


**Switch the user:**

```
# su - <user_name>
```

# The Docker Hub

https://hub.docker.com/?next=https%3A%2F%2Fhub.docker.com%2F

books Puppet 5.0 reference Beginner's guide to G5 Downloads VS25 Jenkins SaaScompute splunk Top 50 Python Projects https://www.kitfox.com/centos-Python-pip LIT yarnProxy Run a command

 docker hub

Explore Help Sign In

## Docker Hub


Dev test pipeline automation, 100,000+ free apps, public and private registries


New to Docker?  
Create your free Docker ID to get started.

☐ I agree to Docker's [Terms of Service](#).

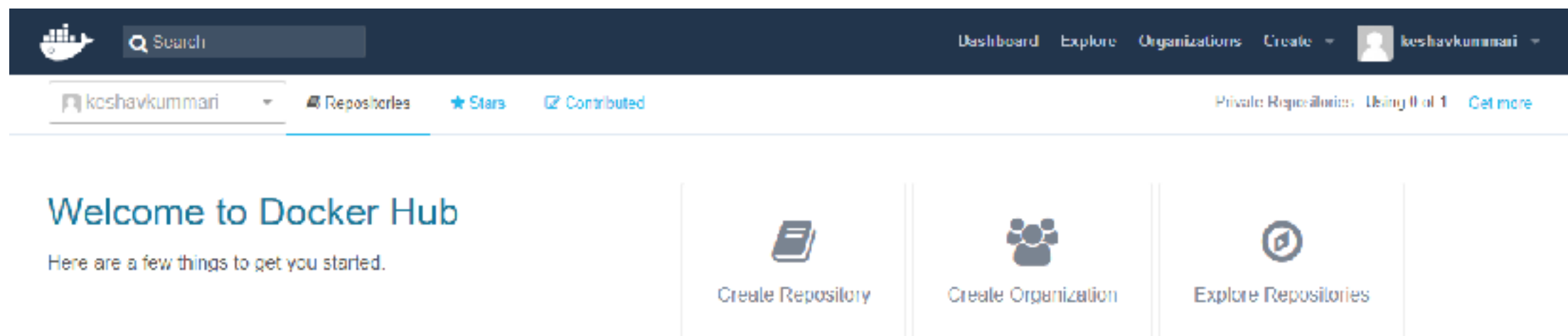
☐ I agree to Docker's [Privacy Policy](#) and [Data Processing Terms](#).

☐ I would like to receive email updates from Docker, including its various services and products.

☒ I'm not a robot 

Join us in San Francisco  
  
register now

# Login to Docker Online Repo





## Base Images

### Check the local images:

```
# docker images
```

### Search for Repository:

```
# docker search <any_name>
```

Example : # docker search centos or # docker search apache

*Note: Look at “STARS” and “OFFICIAL” info.*

### Pull a docker image:

```
# docker pull hello-world
```

```
# docker images
```

```
# docker run <image id>
```

```
# docker pull centos:centos6
```

```
# docker pull centos
```

### Search nginx :

```
# docker search nginx
```

### Pull Nginx :

```
# docker pull nginx:latest
```

```
# docker images
```

```
# docker inspect nginx <image id>
```

```
# docker insepct nginx
```

## Running Containers

# docker images

# docker run hello-world

### **Pull the images from Docker Hub:**

# docker pull docker/whalesay

# docker images

### **Run a container with specific message:**

# docker run dokcer/whalesay cowsay hello

# docker run dokcer/whalesay cowsay IAMDOCKER

### **List the list of running containers:**

# docker ps

```
# docker images
```

```
# docker inspect docker/whalesay
```

### List of running dcontainers:

```
# docker ps -a
```

```
# docker inspect happy_borg
```

### Run a container and login into it:

```
# docker run -it centos:latest /bin/bash
```

### Now, install Apache Webserver on this container:

```
# yum update -y
```

```
# yum install httpd* -y
```

```
# systemctl start httpd.service
```

## Do ssh to container:

```
# ssh user@ipaddress
```

## Run a container in the background:

```
# docker run -d centos:latest /bin/bash
```

```
# docker images
```

```
# docker ps -a
```

## Run a nginx in background:

```
# docker run -d nginx:latest
```

```
# docker ps
```

```
# docker inspect <docker_name/docker id>
```

```
# ping 192.168.0.11
```

## Check Webserver which is running in a container:

```
# elinks http://192.168.0.11
```

## To stop docker :

```
# docker stop <docker_name/image id>
```

## Run a container with custom name:

```
# docker run -d -name=MyWeb1 nginx:latest
```

```
# docker images
```

```
# docker inspect MyWeb1
```

```
# docker ps -a
```

# Container LifeCycle

# docker images

**Run a container in background:**

# docker run -d - - name=LifeCycle1 nginx:latest

# docker ps -a

To check the docker container ipaddress:

# ifconfig

Attach the container:

# docker attach LifeCycle1

# docker start LifeCycle1

# docker exec /bin/bash LifeCycle1

# docker exec -it LifeCycle1 /bin/bash

# ps -aux | grep nginx

# exit

# docker stop LifeCycle1

# docker ps

## Image and Container Management

```
# docker images # docker ps
```

```
# docker ps -a
```

### **Delete a Container:**

```
# docker rmi centos:centos6
```

```
# docker rmi hello-world
```

```
# docker ps -a
```

```
# docker rmi -f hello-world
```

```
# docker rm <container_id>
```

```
# docker rmi hello-world:latest
```

```
# docker images
```

```
# docker ps -a
```

```
# docker ps
```



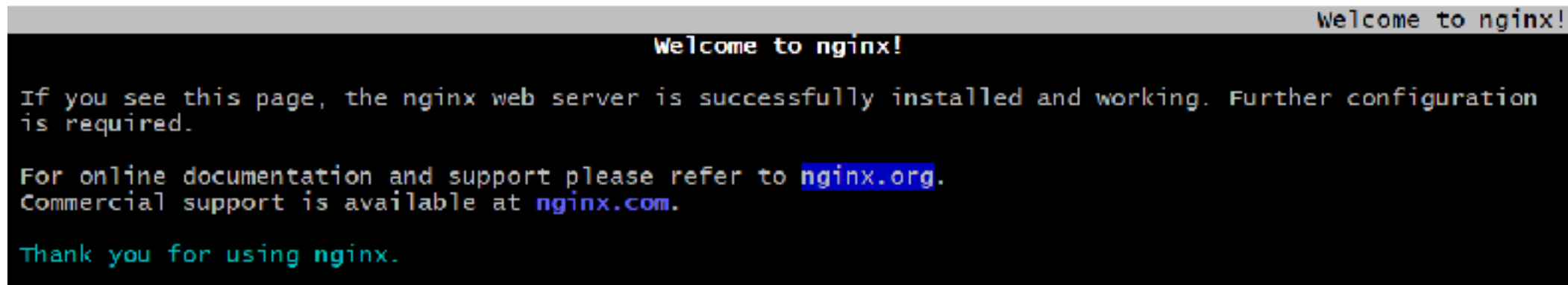
## Redirection – Ports and Volumes

```
# docker run -d nginx:latest
```

```
# docker ps
```

```
# docker inspect priceless_blackwell | grep IPAddr
```

```
# elinks http://172.17.0.3
```

A screenshot of a terminal window with a dark background. The title bar at the top is light gray and contains the text "Welcome to nginx!". The terminal content shows the same "Welcome to nginx!" message in a light blue font. Below it, a paragraph in a light green font states: "If you see this page, the nginx web server is successfully installed and working. Further configuration is required." This is followed by two lines of light green text: "For online documentation and support please refer to [nginx.org](http://nginx.org)." and "Commercial support is available at [nginx.com](http://nginx.com)." The final line of the screenshot is "Thank you for using nginx." in a light green font.

```
Welcome to nginx!  
  
Welcome to nginx!  
  
If you see this page, the nginx web server is successfully installed and working. Further configuration  
is required.  
  
For online documentation and support please refer to nginx.org.  
Commercial support is available at nginx.com.  
  
Thank you for using nginx.
```

*Nginx WebServer is Up in the Container*

## Stop the Docker Container using name:

```
# docker stop priceless_Blackwell
```

## Remove all your previous Containers:

```
# docker rm `docker ps -a -q`
```

Download, & Run Nginx in

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

```
# docker inspect WebServer2 | grep IPAddr
```

```
# elinks http://localhost:8080
```

```
# elinks http://172.17.0.2
```

```
# docker run -d -p 8080:80 --name=dev.example.com -v /mnt/  
data nginx:latest
```

# Docker Nginx WebServer Example

```
[user@keshav-kummari-abc5ff761 ~]$ docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
61605b64b56e   nginx:latest  "nginx -g 'daemon ...'"  36 seconds ago  Up 36 seconds  0.0.
0.0:8080->80/tcp   dev.example.com
[user@keshav-kummari-abc5ff761 ~]$ docker port dev.example.com
80/tcp -> 0.0.0.0:8080
[user@keshav-kummari-abc5ff761 ~]$ docker inspect dev.example.com | grep IPAddr
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.2",
    "IPAddress": "172.17.0.2",
[user@keshav-kummari-abc5ff761 ~]$ elinks http://172.17.0.2
```

Welcome to nginx!

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to [nginx.org](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

Thank you for using nginx.

Do you really want to exit ELinks?

[ Yes ]

[ No ]

## Route the Ports and Launch a local Container

```
[user@keshav-kummari-abc5ff761 www]$ pwd
/home/user/www
[user@keshav-kummari-abc5ff761 www]$ ls -lrt
total 4
-rw-rw-r--. 1 user user 147 Jul 11 12:57 index.html
[user@keshav-kummari-abc5ff761 www]$ cat index.html
<html>
<head>
<title>Docker Nginx WebServer</title>
</head>
<body bgcolor="olive">

<h1> Welcome to Development Environment </h1>

</body>
</html>
```

```
[user@keshav-kummari-abc5ff761 www]$ docker run -d -p 8080:80 --name=tst.example.com -v /home/user/www:/usr/nginx/html nginx:latest
790c1fe194de511dc7141eb172d6d9064196cc854f15d5469ee99f6a0a06fa2a
[user@keshav-kummari-abc5ff761 www]$ |
```

## Launch a Container and mount local Directory with Nginx

```
[user@keshav-kummari-abc5ff761 www]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                latest             3c5a05123222       4 days ago         109MB
centos               latest             49f7960eb7e4       5 weeks ago        200MB
[user@keshav-kummari-abc5ff761 www]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
790c1fe194de       nginx:latest       "nginx -g 'daemon ..."  51 seconds ago     Up 51 seconds      0.0.0.0:8080->80/tcp
[user@keshav-kummari-abc5ff761 www]$ docker port tst.example.com
80/tcp -> 0.0.0.0:8080
[user@keshav-kummari-abc5ff761 www]$
[user@keshav-kummari-abc5ff761 www]$ docker inspect tst.example.com | grep IPAddr
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.2",
    "IPAddress": "172.17.0.2",
```

```
[user@keshav-kummari-abc5ff761 www]$ elinks http://localhost:8080
welcome to nginx!

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration
is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.
```

## The Dockerfile

**Check currently running Docker containers:**

```
# docker ps
```

**All containers info:**

```
# docker ps -a
```

**Remove all existing Docker containers:**

```
# docker rm `docker ps -a -q`
```

**Check the list of images are available :**

```
# docker images
```

## Let's Create a Docker File

```
# vi Dockerfile
```

```
FROM debian:stable
```

```
MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>
```

```
RUN apt-get update
```

```
RUN apt-get upgrade
```

```
:wq!
```

## Run a Ubuntu Container using Dockerfile

# docker build -t keshavkummari/myapache .

```
[user@keshav-kummari-abc5ff761 ~]$ pwd
/home/user
[user@keshav-kummari-abc5ff761 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx                latest             3c5a05123222       4 days ago        109MB
centos               latest             49f7960eb7e4       5 weeks ago       200MB
[user@keshav-kummari-abc5ff761 ~]$ vi Dockerfile
[user@keshav-kummari-abc5ff761 ~]$
[user@keshav-kummari-abc5ff761 ~]$ ls -lrt Dockerfile
-rw-rw-r--. 1 user user 116 Jul 11 13:24 Dockerfile
[user@keshav-kummari-abc5ff761 ~]$ cat Dockerfile
FROM debian:stable

MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>

RUN apt-get update
RUN apt-get upgrade
[user@keshav-kummari-abc5ff761 ~]$ docker build -t keshavkummari/myapache .
Sending build context to Docker daemon  58.19MB
Step 1/4 : FROM debian:stable
stable: Pulling from library/debian
31c6765cabf1: Pull complete
Digest: sha256:6aedee3ef827be950fc9002d5eb767f0d2766612734e476542772531ee26a661
Status: Downloaded newer image for debian:stable
```



## Output

```
---> 95c509d2f403
Step 2/4 : MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>
---> Running in 3fd43fd67c20
---> bd63d56147bb
Removing intermediate container 3fd43fd67c20
Step 3/4 : RUN apt-get update
---> Running in 1d166b210d9d
Get:1 http://security.debian.org/debian-security stable/updates InRelease [94.3 kB]
Get:2 http://security.debian.org/debian-security stable/updates/main amd64 Packages [369 kB]
Ign:3 http://cdn-fastly.deb.debian.org/debian stable InRelease
Get:4 http://cdn-fastly.deb.debian.org/debian stable-updates InRelease [91.0 kB]
Get:5 http://cdn-fastly.deb.debian.org/debian stable Release [118 kB]
Get:6 http://cdn-fastly.deb.debian.org/debian stable-updates/main amd64 Packages [11.0 kB]
Get:7 http://cdn-fastly.deb.debian.org/debian stable Release.gpg [2434 B]
Get:8 http://cdn-fastly.deb.debian.org/debian stable/main amd64 Packages [7122 kB]
Fetched 7807 kB in 3s (2281 kB/s)
Reading package lists...
---> 7def33f9e994
Removing intermediate container 1d166b210d9d
Step 4/4 : RUN apt-get upgrade
---> Running in 6c8ced0cf3f1
Reading package lists...
Building dependency tree...
Reading state information...
Calculating upgrade...
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
---> b30d1ba96b96
Removing intermediate container 6c8ced0cf3f1
Successfully built b30d1ba96b96
Successfully tagged keshavkummari/myapache:latest
[user@keshav-kummari-abc5ff761 ~]$ |
```

## Install a Package using Dockerfile

```
[user@keshav-kummari-abc5ff761 ~]$ pwd
/home/user
[user@keshav-kummari-abc5ff761 ~]$ docker build -t keshavkummari/myapache .
Sending build context to Docker daemon 58.19MB
Step 1/5 : FROM debian:stable
--> 95c509d2f403
Step 2/5 : MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>
--> Using cache
--> bd63d56147bb
Step 3/5 : RUN apt-get update
--> Using cache
--> 7def33f9e994
Step 4/5 : RUN apt-get upgrade
--> Using cache
--> b30d1ba96b96
Step 5/5 : RUN apt-get install telnet
--> Running in 9668abd0d250
Reading package lists...
Building dependency tree...
Reading state information...
The following additional packages will be installed:
  netbase
The following NEW packages will be installed:
  netbase telnet
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 91.1 kB of archives.
After this operation, 206 kB of additional disk space will be used.
Do you want to continue? [Y/n] Abort.
The command '/bin/sh -c apt-get install telnet' returned a non-zero code: 1
[user@keshav-kummari-abc5ff761 ~]$
```

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

```
[user@keshav-kummari-abc5ff761 ~]$ cat Dockerfile  
FROM debian:stable
```

```
MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>
```

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

```
ENV MYVALUE my-value
```

## Configure Environment Variables using Dockerfile

```
[user@keshav-kummari-abc5ff761 ~]$ cat Dockerfile
FROM debian:stable

MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>

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

ENV MYVALUE my-value
[user@keshav-kummari-abc5ff761 ~]$ docker build -t keshavkummari/myapache .
Sending build context to Docker daemon 58.19MB
Step 1/4 : FROM debian:stable
--> 95c509d2f403
Step 2/4 : MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>
--> Using cache
--> bd63d56147bb
Step 3/4 : RUN apt-get update && apt-get upgrade -y && apt-get install -y apache2 telnet elinks openssh-server vim
--> Using cache
--> a98ced4479a5
Step 4/4 : ENV MYVALUE my-value
--> Running in d8cc16a2fcc3
--> d313a6e11f68
Removing intermediate container d8cc16a2fcc3
Successfully built d313a6e11f68
Successfully tagged keshavkummari/myapache:latest
[user@keshav-kummari-abc5ff761 ~]$
```

## Cross check the Environment Variables by login into container

```
Successfully tagged keshavkummari/myapache:latest
[user@keshav-kummari-abc5ff761 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
keshavkummari/myapache  latest            d313a6e11f68       About a minute ago  303MB
<none>              <none>           b30d1ba96b96       26 minutes ago     117MB
nginx                latest            3c5a05123222       4 days ago         109MB
debian               stable            95c509d2f403       2 weeks ago        101MB
centos                latest            49f7960eb7e4       5 weeks ago        200MB
[user@keshav-kummari-abc5ff761 ~]$ docker run -it keshavkummari/myapache:latest /bin/bash
root@55563520525f:/# echo $MYVALUE
my-value
root@55563520525f:/#
```

## Clean Up!!

```
[user@keshav-kummari-abc5ff761 ~]$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
keshavkummari/myapache  latest             d313a6e11f68       3 minutes ago      303MB
<none>               <none>             b30d1ba96b96       27 minutes ago     117MB
nginx                latest             3c5a05123222       4 days ago         109MB
debian               stable             95c509d2f403       2 weeks ago        101MB
centos                latest             49f7960eb7e4       5 weeks ago        200MB
[user@keshav-kummari-abc5ff761 ~]$
[user@keshav-kummari-abc5ff761 ~]$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
[user@keshav-kummari-abc5ff761 ~]$
[user@keshav-kummari-abc5ff761 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
55563520525f       keshavkummari/myapache:latest  "/bin/bash"        About a minute ago  Exited (0)
34 seconds ago    friendly_sammet
9668abd0d250       b30d1ba96b96       "/bin/sh -c 'apt-g..."  12 minutes ago     Exited (1)
12 minutes ago    festive_cray
db24793a4161       b30d1ba96b96       "/bin/sh -c 'apt-g..."  23 minutes ago     Exited (100
) 23 minutes ago  musing_mcnulty
[user@keshav-kummari-abc5ff761 ~]$ docker rm `docker ps -a -q`
55563520525f
9668abd0d250
db24793a4161
[user@keshav-kummari-abc5ff761 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS
NAMES
[user@keshav-kummari-abc5ff761 ~]$
```

```
[user@keshav-kummari-abc5ff761 ~]$ pwd
/home/user
[user@keshav-kummari-abc5ff761 ~]$ vim Dockerfile
[user@keshav-kummari-abc5ff761 ~]$ cat Dockerfile
FROM debian:stable

MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>

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

ENV MYVALUE my-value

EXPOSE 80
EXPOSE 22

[user@keshav-kummari-abc5ff761 ~]$
```

## Enable Ports using Dockerfile

```
[user@keshav-kummari-abc5ff761 ~]$ docker build -t keshavkummari/myapache .
Sending build context to Docker daemon 58.19MB
Step 1/6 : FROM debian:stable
---> 95c509d2f403
Step 2/6 : MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>
---> Using cache
---> bd63d56147bb
Step 3/6 : RUN apt-get update && apt-get upgrade -y && apt-get install -y apache2 telnet elinks openssh-server vim
---> Using cache
---> a98ced4479a5
Step 4/6 : ENV MYVALUE my-value
---> Using cache
---> d313a6e11f68
Step 5/6 : EXPOSE 80
---> Running in 1dc7f97a232b
---> 342454f3878c
Removing intermediate container 1dc7f97a232b
Step 6/6 : EXPOSE 22
---> Running in 454ee2a30639
---> 76f9c20694bf
Removing intermediate container 454ee2a30639
Successfully built 76f9c20694bf
Successfully tagged keshavkummari/myapache:latest
[user@keshav-kummari-abc5ff761 ~]$ |
```



## Start Apache WebServer using Dockerfile

```
[user@keshav-kummari-abc5ff761 ~]$  
[user@keshav-kummari-abc5ff761 ~]$ pwd  
/home/user  
[user@keshav-kummari-abc5ff761 ~]$  
[user@keshav-kummari-abc5ff761 ~]$ ls -lrt Dockerfile  
-rw-rw-r--. 1 user user 272 Jul 11 14:09 Dockerfile  
[user@keshav-kummari-abc5ff761 ~]$  
[user@keshav-kummari-abc5ff761 ~]$ cat Dockerfile  
FROM debian:stable  
  
MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>  
  
RUN apt-get update && apt-get upgrade -y && apt-get install -y apache2 telnet elinks openssh-server vim  
  
ENV MYVALUE my-value  
  
EXPOSE 80  
EXPOSE 22  
  
CMD ["/usr/sbin/apache2ctl","-D","FOREGROUND"]  
[user@keshav-kummari-abc5ff761 ~]$
```

## Start a Apache Service using Dockerfile

```
[user@keshav-kummari-abc5ff761 ~]$ docker build -t keshavkummari/myapache .
Sending build context to Docker daemon 58.19MB
Step 1/7 : FROM debian:stable
--> 95c509d2f403
Step 2/7 : MAINTAINER keshavkummari <keshavkummari@cloudcicd.org>
--> Using cache
--> bd63d56147bb
Step 3/7 : RUN apt-get update && apt-get upgrade -y && apt-get install -y apache2 telnet elinks openssh-server vim
--> Using cache
--> a98ced4479a5
Step 4/7 : ENV MYVALUE my-value
--> Using cache
--> d313a6e11f68
Step 5/7 : EXPOSE 80
--> Using cache
--> 342454f3878c
Step 6/7 : EXPOSE 22
--> Using cache
--> 76f9c20694bf
Step 7/7 : CMD /usr/sbin/apach2ctl -D FOREGROUND
--> Running in eda86eb8ce4f
--> 3f0ebbf12229
Removing intermediate container eda86eb8ce4f
Successfully built 3f0ebbf12229
Successfully tagged keshavkummari/myapache:latest
[user@keshav-kummari-abc5ff761 ~]$ |
```

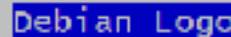
## Now, Apache2 WebServer is Up!

```
[user@keshav-kummari-abc5ff761 ~]$ docker run -d keshavkummari/myapache:latest
15ea6e84d5fb5b22531cdd7445a0da488bd35aacaab16b4790813b4556ed6965
[user@keshav-kummari-abc5ff761 ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
15ea6e84d5fb	keshavkummari/myapache:latest	"/usr/sbin/apache2..."	15 seconds ago	Up 14 seconds

```
[user@keshav-kummari-abc5ff761 ~]$ docker inspect focused_fermat | grep IPAddr
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.2",
    "IPAddress": "172.17.0.2",
[user@keshav-kummari-abc5ff761 ~]$ elinks http://172.17.0.2
```

Apache2 Debian Default Page: It works (1/2)

 Apache2 Debian Default Page

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Debian systems. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.

# Execute Docker Container in Background

```
[user@keshav-kummari-abc5ff761 ~]$ docker exec -it focused_fermat /bin/bash
root@15ea6e84d5fb:/# ps -aux | grep http
root      75  0.0  0.0  11104   704 ?        S+   14:17   0:00 grep http
root@15ea6e84d5fb:/#
root@15ea6e84d5fb:/# ps -aux | grep apache
root      1  0.0  0.0   4280   688 ?        Ss   14:10   0:00 /bin/sh /usr/sbin/apache2ctl -D FOREGROUND
root     13  0.0  0.1  75604  3304 ?        S    14:10   0:00 /usr/sbin/apache2 -D FOREGROUND
www-data 14  0.0  0.2 364764  4308 ?        Sl   14:10   0:00 /usr/sbin/apache2 -D FOREGROUND
www-data 15  0.0  0.2 430364  4872 ?        Sl   14:10   0:00 /usr/sbin/apache2 -D FOREGROUND
root     77  0.0  0.0  11104   704 ?        S+   14:18   0:00 grep apache
root@15ea6e84d5fb:/# echo $MYVALUE
my-value
root@15ea6e84d5fb:/# exit
exit
[user@keshav-kummari-abc5ff761 ~]$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
15ea6e84d5fb	keshavkummari/myapache:latest	"/usr/sbin/apache2..."	8 minutes ago	Up 8 minutes
	22/tcp, 80/tcp	focused_fermat		

```
[user@keshav-kummari-abc5ff761 ~]$
```