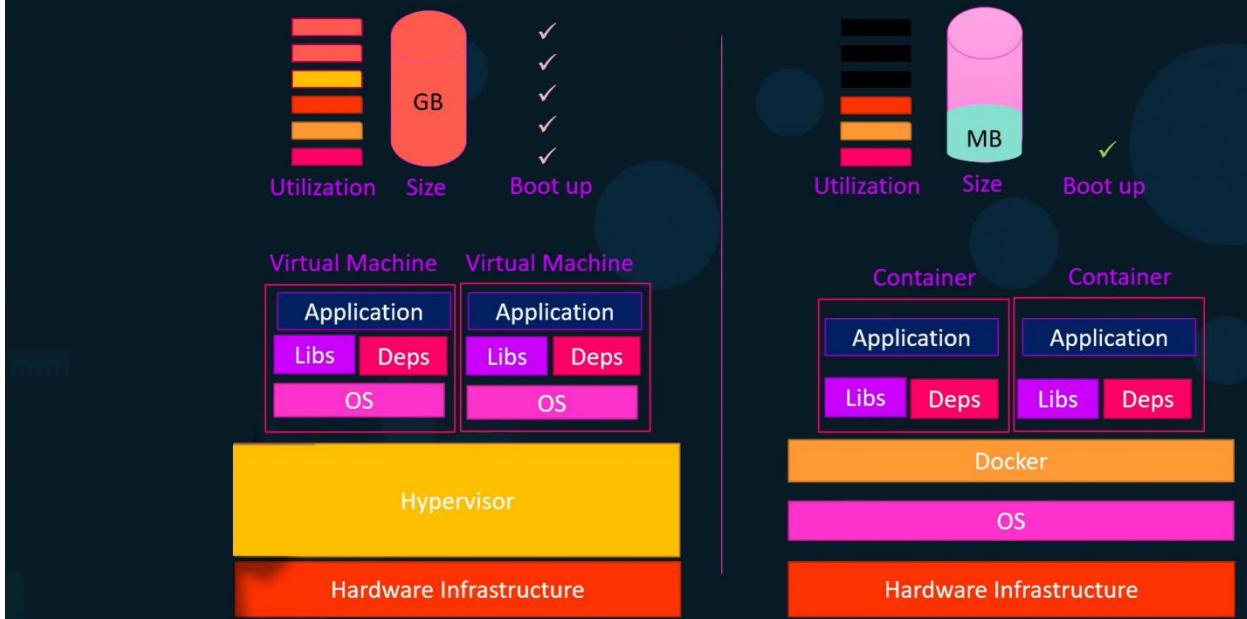
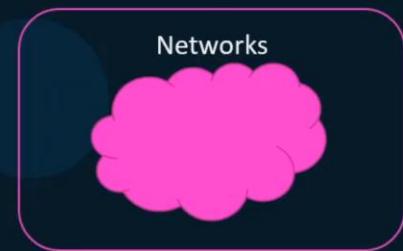


Docker

Containers vs Virtual Machines



Docker Objects



Registry



Docker service configuration

Check Service Status

```
▶ systemctl start docker
```

```
▶ systemctl status docker
```

```
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-10-21 04:21:01 UTC; 3 days ago
     Docs: https://docs.docker.com
 Main PID: 4197 (dockerd)
    Tasks: 13
   Memory: 129.7M
      CPU: 9min 6.980s
    CGroup: /system.slice/docker.service
           └─4197 /usr/bin/dockerd -H fd:// -H tcp://0.0.0.0 --containerd=/run/containerd/containerd.sock
```

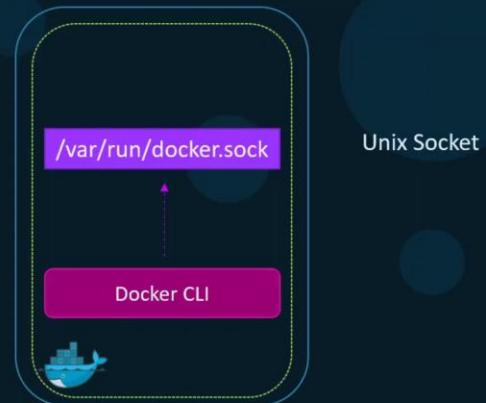
```
▶ systemctl stop docker
```

Start Manually

▶ dockerd

```
INFO[2020-10-24T08:20:40.372653463Z] Starting up
INFO[2020-10-24T08:20:40.375298351Z] parsed scheme: "unix"                                     module=grpc
INFO[2020-10-24T08:20:40.375510773Z] scheme "unix" not registered, fallback to default scheme   module=grpc
INFO[2020-10-24T08:20:40.375657667Z] ccResolverWrapper: sending update to cc: {[{unix:///run/containerd/containerd.sock
0 <nil>}]} <nil>} module=grpc
INFO[2020-10-24T08:20:40.375973480Z] ClientConn switching balancer to "pick_first" module=grpc
INFO[2020-10-24T08:20:40.377210185Z] parsed scheme: "unix"                                     module=grpc
INFO[2020-10-24T08:20:40.377384998Z] scheme "unix" not registered, fallback to default scheme module=grpc
INFO[2020-10-24T08:20:40.377491827Z] ccResolverWrapper: sending update to cc: {[{unix:///run/containerd/containerd.sock
0 <nil>}]} <nil>} module=grpc
INFO[2020-10-24T08:20:40.377762558Z] ClientConn switching balancer to "pick_first" module=grpc
INFO[2020-10-24T08:20:40.381198263Z] [graphdriver] using prior storage driver: overlay2
WARN[2020-10-24T08:20:40.572888603Z] Your kernel does not support swap memory limit
WARN[2020-10-24T08:20:40.573014192Z] Your kernel does not support cgroup rt period
WARN[2020-10-24T08:20:40.573404879Z] Your kernel does not support cgroup rt runtime
```

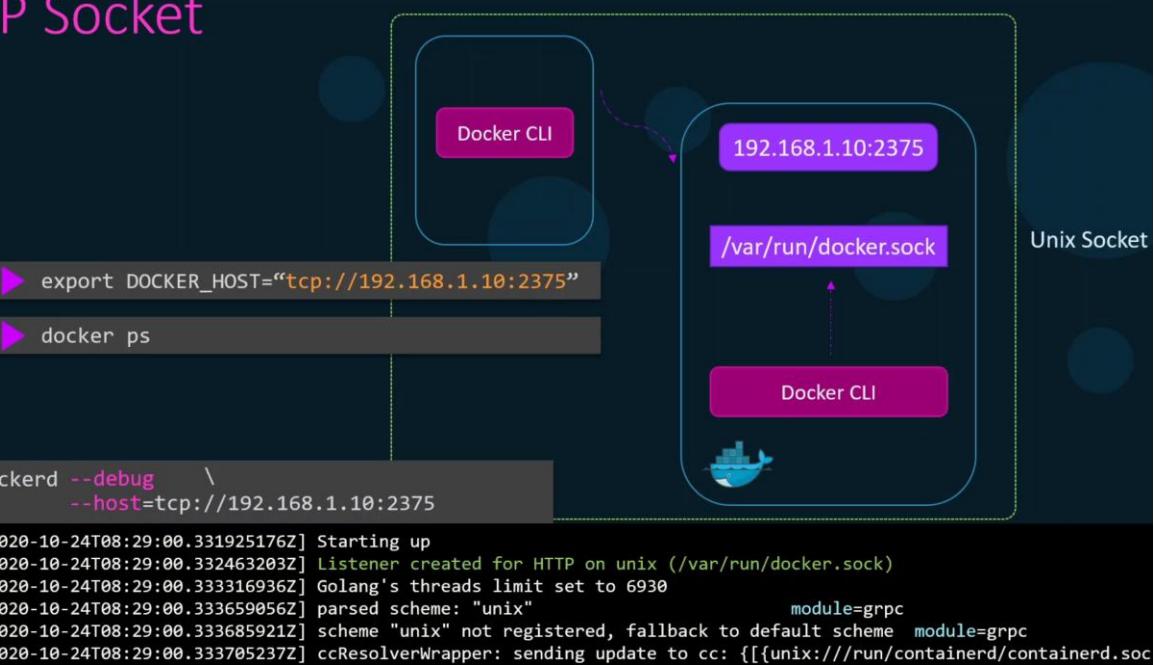
Unix Socket



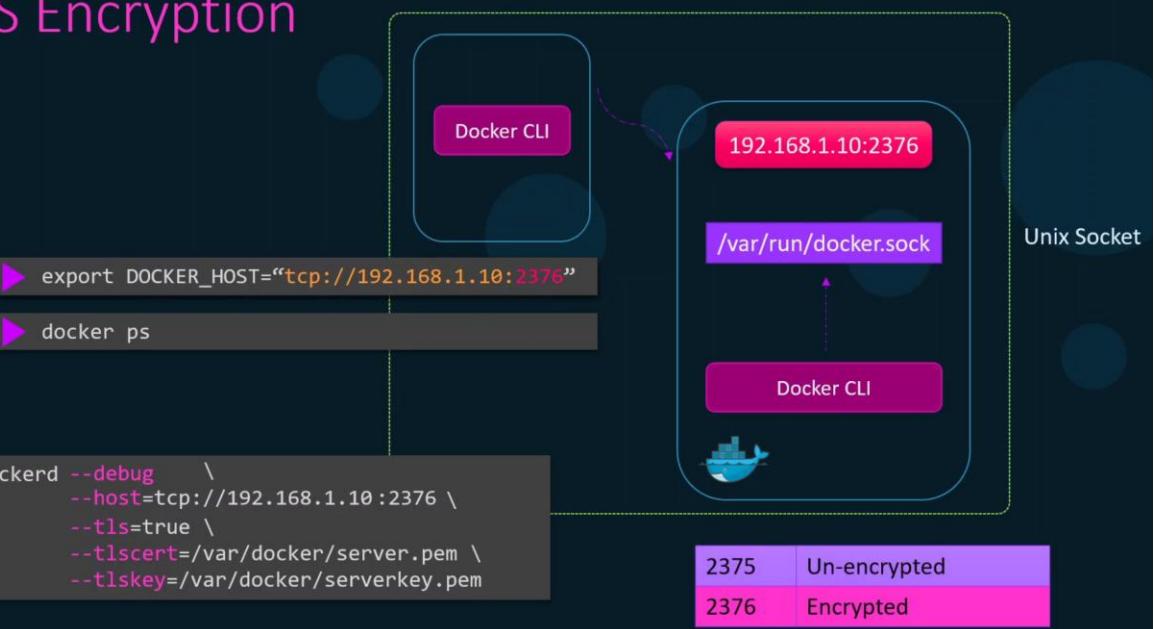
▶ dockerd --debug

```
INFO[2020-10-24T08:29:00.331925176Z] Starting up
DEBU[2020-10-24T08:29:00.332463203Z] Listener created for HTTP on unix (/var/run/docker.sock)
DEBU[2020-10-24T08:29:00.333316936Z] Golang's threads limit set to 6930
```

TCP Socket



TLS Encryption



Daemon Configuration File

```
▶ dockerd
```

```
/etc/docker/daemon.json
{
  "debug": true,
  "hosts": ["tcp://192.168.1.10:2376"]
  "tls": true,
  "tlscert": "/var/docker/server.pem",
  "tlskey": "/var/docker/serverkey.pem"
}
```

```
▶ dockerd --debug=false
```

```
unable to configure the Docker daemon with file /etc/docker/daemon.json: the following directives are specified both as a flag and in the configuration file: debug: (from flag: false, from file: true)
```

```
▶ systemctl start docker
```

Docker Objects

```
▶ docker
```

```
<docker-object>
```

```
<sub-command>
```

```
[options]
```

```
<Arguments/Commands>
```

```
▶ docker image ls
```



Images

```
▶ docker network ls
```



```
▶ docker container ls
```



Containers

```
▶ docker volume ls
```



Volumes

Docker Engine Command



Container Create - Create a new container

```
▶ docker container create httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
8ec398bc0356: Pull complete
354e6904d655: Pull complete
36412f6b2f6e: Pull complete
Digest:
sha256:769018135ba22d3a7a2b91cb89b8de711562cdf51ad6621b2b9b13e95f3798de
Status: Downloaded newer image for httpd:latest
36a391532e10d45f772f2c9430c2cc38dad4b441aa7a1c44d459f6fa3d78c6b6
```

```
▶ ls /var/lib/docker/
builder  containers  network          plugins   swarm    trust
buildkit  image       overlay2        runtimes  tmp      volumes
```

```
▶ ls -lrt /var/lib/docker/containers/
36a391532e10d45f772f2c9430c2cc38dad4b441aa7a1c44d459f6fa3d78c6b6
```

```
▶ ls -lrt /var/lib/docker/containers/36a391532e10*
Checkpoint      hostconfig.json  config.v2.json
```

Container ls - List the details for container

```
▶ docker container ls
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS          PORTS
36a391532e10        httpd      "httpd-foreground" 2 minutes ago   Created
▶ docker container ls -a
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS          PORTS
36a391532e10        httpd      "httpd-foreground" 2 minutes ago   Created
▶ docker container ls -l
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS          PORTS
36a391532e10        httpd      "httpd-foreground" 2 minutes ago   Created
▶ docker container ls -q
36a391532e10
```

Container Start - Start a container

```
▶ docker container start 36a391532e10
36a391532e10
▶ docker container ls
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS          PORTS
36a391532e10        httpd      "httpd-foreground" 6 minutes ago   Up 1 minutes   80/tcp
                                         NAMES
                                         charming_wile
```

Container Run – Create and Start a container

```
▶ docker container create httpd
+                               ▶ docker container start 36a391532e10
▶ docker container run httpd
Unable to find image 'httpd:latest' locally
latest: Pulling from library/httpd
```

```
▶ docker container run ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
2746a4a261c9: Pull complete
4c1d20cdee96: Pull complete
0d3160e1d0de: Pull complete
c8e37668deea: Pull complete
Digest: sha256:250cc6f3f3ffc5cdcaa9d8f4946ac79821aafb4d3afc93928f0de9336eba21aa4
Status: Downloaded newer image for ubuntu:latest

▶ docker container ls
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS           PORTS     NAMES
d969ecdb44ea      ubuntu      "/bin/bash"   2 minutes ago   Exited (0) 2 minutes ago

▶ docker container ls -a
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS           PORTS     NAMES
d969ecdb44ea      ubuntu      "/bin/bash"   2 minutes ago   Exited (0) 2 minutes ago
```

```
▶ docker container run ubuntu
▶ docker container ls -a
CONTAINER ID        IMAGE       COMMAND      CREATED          STATUS           PORTS     NAMES
d969ecdb44ea      ubuntu      "/bin/bash"   2 minutes ago   Exited (0) 2 minutes ago
```

Container Run – With Options

```
▶ docker container run -it ubuntu
root@6caba272c8f5:/#
root@6caba272c8f5:/# hostname
6caba272c8f5
root@6caba272c8f5:/#
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6caba272c8f5	ubuntu	"/bin/bash"	About a minute ago	Up About a minute		quizzical_aust

▶ docker container run -it ubuntu
▶ docker container run ubuntu -it

Container Run – exiting a running process

```
▶ docker container run -it ubuntu
root@6caba272c8f5:/#
root@6caba272c8f5:/# hostname
6caba272c8f5
root@6caba272c8f5:/# exit
exit
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
6caba272c8f5	ubuntu	"/bin/bash"	8 minutes ago	Exited (0) 37 seconds ago		quizzical_aust

Container Run – Container Name

```
▶ docker container ls -l
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
6caba272c8f5      ubuntu              "/bin/bash"         8 minutes ago     Exited (0) 37 seconds ago   quizzical_austin

▶ docker container run -itd --name=webapp ubuntu
59aa5eacd88c42970754cd6005ce315944a2efcd32288df998b29267ae54c152

▶ docker container ls -l
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
59aa5eacd88c      ubuntu              "/bin/bash"         20 seconds ago    Up 19 seconds

▶ docker container rename webapp custom-webapp

▶ docker container ls -l
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
59aa5eacd88c      ubuntu              "/bin/bash"         About a minute ago  Up About a minute  custom-webapp
```

Container Run – Detached Mode

```
▶ docker container run httpd
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.3. Set the 'ServerName' directive globally to suppress this message
[Thu Sep 17 15:39:31.138134 2020] [mpm_event:notice] [pid 1:tid 139893041316992] AH00489: Apache/2.4.46 (Unix) configured -- resuming normal operations
[Thu Sep 17 15:39:31.138584 2020] [core:notice] [pid 1:tid 139893041316992] AH00094: Command line: 'httpd -D FOREGROUND'

▶ docker container run -d httpd
11cbd7fe7e65a9da453e159ed0fe163592dcc8a7845abc91b8305c78f50ac70

▶ docker container attach 11cb
```

Interacting with a Running Container

Container Run – Escape Sequence

```
▶ docker container run -it ubuntu
root@6caba272c8f5:/# exit
exit

▶ docker container ls -l
CONTAINER ID        IMAGE       COMMAND      CREATED     STATUS      PORTS     NAMES
6caba272c8f5        ubuntu      "/bin/bash"   8 minutes ago   Exited (0) 37 seconds ago               quizzical_austin

▶ docker container run -it ubuntu
root@b71f15d33b60:/# [PRESS CTRL+p+q]

▶ docker container ls -l
CONTAINER ID        IMAGE       COMMAND      CREATED     STATUS      PORTS     NAMES
b71f15d33b60        ubuntu      "/bin/bash"   3 minutes ago   Up 3 minutes                  magical_babbage
```

Container Exec – Executing Commands

```
▶ docker container exec b71f15d33b60 hostname
b71f15d33b60

▶ docker container exec -it b71f15d33b60 /bin/bash
root@b71f15d33b60:/#
root@b71f15d33b60:/# ps -ef
UID      PID  PPID  C STIME TTY          TIME CMD
root      1      0  0 12:53 pts/0    00:00:00 /bin/bash
root      86      1  0 13:10 pts/0    00:00:00 ps -ef
root@b71f15d33b60:/# tty
/dev/pts/0
root@b71f15d33b60:/# exit
exit

▶ docker container attach b71f15d33b60
root@b71f15d33b60:/#
```

Inspecting a Container

Container Inspect

```
▶ docker container inspect webapp
[{"Id": "59aa5eacd88c42970754cd6005ce315944a2efcd32288df998b29267ae54c152", "Created": "2020-01-14T13:23:01.225868339Z", "Path": "/bin/bash", "Args": [], "State": { "Status": "running", "Running": true, "Paused": false, "Restarting": false, "StartTimestamp": "2020-01-14T13:23:01.225868339Z", "LastTransitionTimestamp": "2020-01-14T13:23:01.225868339Z", "Ready": true, "OOMKilled": false, "PIDs": 1, "ExitCode": 0, "Error": "", "Version": "1.0", "IP": "172.17.0.5", "IPPrefixLen": 16, "IPv6Gateway": "", "GlobalIPv6Address": "", "GlobalIPv6PrefixLen": 0, "MacAddress": "02:42:ac:11:00:05", "DriverOpts": null }, "NetworkSettings": { "Bridge": "bridge", "Networks": { "bridge": { "IP": "172.17.0.5", "CIDR": "172.17.0.0/16", "Mac": "02:42:ac:11:00:05", "Gw": "172.17.0.1", "Ports": {} } } } }
```

Container Stats

```
▶ docker container stats
CONTAINER ID        NAME          CPU %     MEM USAGE / LIMIT      MEM %     NET I/O            BLOCK I/O
59aa5eacd88c        webapp        50.00%    400KiB / 989.4MiB    0.04%    656B / 0B           0B /
a00b5535783d        epic_leavitt  0.00%    404KiB / 989.4MiB    0.04%    656B / 0B           0B /
616f80b0f026        elegant_cohen  0.00%    404KiB / 989.4MiB    0.04%    656B / 0B           0B /
36a391532e10        charming_wiles 0.01%    8.363MiB / 989.4MiB  0.85%    656B / 0B           0B /
```

Container Top

```
▶ docker container top webapp
UID          PID      PPID      C      STIME      TTY      TIME      CMD
root        17001    16985      0      13:23      ?      00:00:00  stress
```

Container Logs

```
▶ docker container logs logtest
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.6. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, using 172.17.0.6. Set the 'ServerName' directive globally to suppress this message
[Tue Jan 14 13:38:15.699310 2020] [mpm_event:notice] [pid 1:tid 140610463122560] AH00489: Apache/2.4.41 (Unix) configured -- resuming normal operations
[Tue Jan 14 13:38:15.699520 2020] [core:notice] [pid 1:tid 140610463122560] AH00094: Command line: 'httpd -D FOREGROUND'
```

```
▶ docker container logs -f logtest
```

Docker System Events

```
▶ docker container start webapp
webapp

▶ docker system events --since 60m
2020-01-14T18:30:30.423389441Z network connect d349c5984e7eebab74db57b8529df40e11a140f98a6b5e3ee1807aaeafa0e684
(container=68649c8b359f89db7a3866ee0ebcc7261c0cb9697f3a624cd314c8f4f652f84b, name=bridge, type=bridge)
2020-01-14T18:30:30.721669156Z container start 68649c8b359f89db7a3866ee0ebcc7261c0cb9697f3a624cd314c8f4f652f84b (image=ubuntu, name=casethree)
2020-01-14T18:40:46.779320656Z network connect d349c5984e7eebab74db57b8529df40e11a140f98a6b5e3ee1807aaeafa0e684
(container=71c90a19b9876c9ce2eb9d035355a062fdaceed4a714b61ddf0612651d47d3e2, name=bridge, type=bridge)
2020-01-14T18:40:47.076482525Z container start 71c90a19b9876c9ce2eb9d035355a062fdaceed4a714b61ddf0612651d47d3e2 (image=ubuntu, name=webapp)
```

Stopping and Removing a Container

Linux Signals

```
▶ httpd
▶ docker container run --name web httpd

▶ kill -SIGSTOP $(pgrep httpd)
▶ docker container pause web
    freezer cgroup

▶ kill -SIGCONT $(pgrep httpd)
▶ docker container unpause web

▶ kill -SIGTERM $(pgrep httpd)
▶ docker container stop web

▶ kill -SIGKILL $(pgrep httpd)
▶ docker container kill --signal=9 web

▶ kill -9 $(pgrep httpd)
```

Removing a container

```
▶ docker container stop web
web

▶ ls -lrt /var/lib/docker/containers/
36a391532e10d45f772f2c9430c2cc38dad4b441aa7a1c44d459f6fa3d78c6b6

▶ docker container rm web
web

Error response from daemon: You cannot remove a running container
36c57f29b607460fc53dace758dac47afbf8cb698694d2fcfcbb0ab43a74f0d90. Stop the container
before attempting removal or force remove

▶ ls -lrt /var/lib/docker/containers/
```

Remove All Container

```
▶ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
59aa5eacd88c        ubuntu              "/bin/bash"         23 minutes ago    Up 23 minutes
a00b5535783d        ubuntu              "/bin/bash"         25 minutes ago    Up 25 minutes
616f80b0f026        ubuntu              "/bin/bash"         31 minutes ago    Up 28 minutes
36a391532e10        httpd              "httpd-foreground" About an hour ago   Up About an hour   80/tcp
Names: kodekloudagain, epic_leavitt, elegant_cohen, charming_wiles
▶ docker container ls -q
59aa5eacd88c
a00b5535783d
616f80b0f026
36a391532e10
▶ docker container stop $(docker container ls -q)
59aa5eacd88c
a00b5535783d
616f80b0f026
36a391532e10
▶ docker container rm $(docker container ls -aq)
59aa5eacd88c
a00b5535783d
616f80b0f026
36a391532e10
```

Container Prune

```
▶ docker container ls -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
59aa5eacd88c        ubuntu              "/bin/bash"         23 minutes ago    Up 23 minutes
a00b5535783d        ubuntu              "/bin/bash"         25 minutes ago    Up 25 minutes
616f80b0f026        ubuntu              "/bin/bash"         31 minutes ago    Up 28 minutes
36a391532e10        httpd              "httpd-foreground" About an hour ago   Up About an hour   80/tcp
Names: kodekloudagain, epic_leavitt, elegant_cohen, charming_wiles
▶ docker container ls -q
59aa5eacd88c
a00b5535783d
616f80b0f026
36a391532e10
▶ docker container stop $(docker container ls -q)
59aa5eacd88c
a00b5535783d
616f80b0f026
36a391532e10
▶ docker container prune
WARNING! This will remove all stopped containers.
Are you sure you want to continue? [y/N] y
Deleted Containers:
59aa5eacd88c
a00b5535783d
616f80b0f026
36a391532e10
Total reclaimed space: 1223423
```

Remove Flag

```
▶ docker container run --rm ubuntu expr 4 + 5  
9
```

Docker edition

- Community edition
- Enterprise edition-some extra feature with price

Docker install in ubuntu

<https://docs.docker.com/engine/install/ubuntu/>

```
sudo apt-get update  
sudo apt-get install ca-certificates curl gnupg  
  
sudo install -m 0755 -d /etc/apt/keyrings  
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o  
/etc/apt/keyrings/docker.gpg  
sudo chmod a+r /etc/apt/keyrings/docker.gpg  
  
echo \  
"deb [arch="$(dpkg --print-architecture)" signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
"$(. /etc/os-release && echo "$VERSION_CODENAME")" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null  
  
sudo apt-get update  
  
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
#sudo systemctl status docker  
#sudo systemctl start docker  
##sudo systemctl enable docker
```

Run docker command without sudo

```
$ sudo groupadd docker  
$ sudo gpasswd -a $USER docker  
$ sudo usermod -aG docker rajiv  
$ sudo service docker restart  
$ su – rajiv
```

Make user without sudo password

```
$sudo vi /etc/sudoers  
add the following line  
%rajiv ALL=(ALL:ALL) NOPASSWD:ALL
```

Docker compose

Go to the vm server from gitbash

```
git clone https://github.com/dockersamples/example-voting-app.git
```

```
#cd example-voting-app.git  
#cd /vote  
#docker build . -t voting-app      =create voting image  
#docker images      =checking the image create or not  
#docker run -p 5000:80 voting-app
```

Docker Command

Run – start a container

```
▶ docker run nginx  
Unable to find image 'nginx:latest' locally  
latest: Pulling from library/nginx  
fc7181108d40: Already exists  
d2e987ca2267: Pull complete  
0b760b431b11: Pull complete  
Digest:  
sha256:96fb261b66270b900ea5a2c17a26abffabe95506e73c3a3c65869a6dbe83223a  
Status: Downloaded newer image for nginx:latest
```

ps – list containers

▶ docker ps							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	
796856ac413d	nginx	"nginx -g 'daemon off;'"	7 seconds ago	Up 6 seconds	80/tcp	silly_sammet	
▶ docker ps -a							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	
796856ac413d	nginx	"nginx -g 'daemon off;'"	7 seconds ago	Up 6 seconds		silly_sammet	
cff8ac918a2f	redis	"docker-entrypoint.s..."	6 seconds ago	Exited (0) 3 seconds ago		relaxed_aryabhata	

STOP – stop a container

▶ docker ps							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	
796856ac413d	nginx	"nginx -g 'daemon off;'"	7 seconds ago	Up 6 seconds	80/tcp	silly_sammet	
▶ docker stop silly_sammet							
	silly_sammet						
▶ docker ps -a							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	
796856ac413d	nginx	"nginx -g 'daemon off;'"	7 seconds ago	Exited (0) 3 seconds ago		silly_sammet	
cff8ac918a2f	redis	"docker-entrypoint.s..."	6 seconds ago	Exited (0) 3 seconds ago		relaxed_aryabhata	

Rm – Remove a container

▶ docker rm silly_sammet							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	
silly_sammet							
▶ docker ps -a							
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES	
cff8ac918a2f	redis	"docker-entrypoint.s..."	6 seconds ago	Exited (0) 3 seconds ago		relaxed_aryabhata	

images – List images

▶ docker images				
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx	latest	f68d6e55e065	4 days ago	109MB
redis	latest	4760dc956b2d	15 months ago	107MB
ubuntu	latest	f975c5035748	16 months ago	112MB
alpine	latest	3fd9065eaf02	18 months ago	4.14MB

rmi – Remove images

```
▶ docker rmi nginx
```

```
Untagged: nginx:latest
Untagged: nginx@sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Deleted: sha256:f68d6e55e06520f152403e6d96d0de5c9790a89b4fcfc99f4626f68146fa1dbdc
Deleted: sha256:1b0c768769e2bb66e74a205317ba531473781a78b77feef8ea6fd7be7f4044e1
Deleted: sha256:34138fb60020a180e512485fb96fd42e286fb0d86cf1fa2506b11ff6b945b03f
Deleted: sha256:cf5b3c6798f77b1f78bf4e297b27cfa5b6caa982f04caeb5de7d13c255fd7ale
```

Pull – download an image

```
▶ docker run nginx
```

```
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Already exists
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

```
▶ docker pull nginx
```

```
Using default tag: latest
latest: Pulling from library/nginx
fc7181108d40: Pull complete
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest:
sha256:96fb261b66270b900ea5a2c17a26abbfabe95506e73c3a3c65869a6dbe83223a
Status: Downloaded newer image for nginx:latest
```

```
▶ docker run ubuntu
```

```
▶ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
--------------	-------	---------	---------	--------	-------

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
45aacca36850	ubuntu	"/bin/bash"	43 seconds ago	Exited (0) 41 seconds ago	



```
▶ docker run ubuntu
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
45aacca36850	ubuntu	"/bin/bash"	43 seconds ago	Exited (0) 41 seconds ago	

```
▶ docker ps
```

```
▶ docker ps -a
```

Append a command

```
▶ docker run ubuntu
```

```
▶ docker run ubuntu sleep 5
```



Exec – execute a command

```
▶ docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	NAMES
538d037f94a7	ubuntu	"sleep 100"	6 seconds ago	Up 4 seconds	distracted_mcclintock

```
▶ docker exec distracted_mcclintock cat /etc/hosts
```

```
127.0.0.1      localhost
::1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.18.0.2      538d037f94a7
```

Run – attach and detach

```
▶ docker run      rajiv/simple-webapp
This is a sample web application that displays a colored background.
* Serving Flask app "app" (lazy loading)
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)
```

```
▶ docker run -d      rajiv/simple-webapp
a043d40f85fefa414254e4775f9336ea59e19e5cf597af5c554e0a35a1631118
```

```
▶ docker attach a043d
```

Run – tag

```
docker run redis
Using default tag: latest
latest: Pulling from library/redis
f5d23c7fed46: Pull complete
Status: Downloaded newer image for redis:latest

1:C 31 Jul 2019 09:02:32.624 # o000o000o000o Redis is starting o000o000o000o
1:C 31 Jul 2019 09:02:32.624 # Redis version=5.0.5, bits=64, commit=00000000, modified=0, pid=1, just started
1:M 31 Jul 2019 09:02:32.626 # Server initialized
```

```
docker run redis:4.0      TAG
Unable to find image 'redis:4.0' locally
4.0: Pulling from library/redis
e44f086c03a2: Pull complete
Status: Downloaded newer image for redis:4.0

1:C 31 Jul 09:02:56.527 # o000o000o000o Redis is starting o000o000o000o
1:C 31 Jul 09:02:56.527 # Redis version=4.0.14, bits=64, commit=00000000, modified=0, pid=1, just started
1:M 31 Jul 09:02:56.530 # Server initialized
```

RUN - STDIN

```
~/prompt-application$ ./app.sh  
Welcome! Please enter your name: Rajiv
```

```
Hello and Welcome Rajiv
```

```
docker run      rajiv/simple-prompt-docker
```

```
Hello and Welcome !
```

```
docker run -i      rajiv/simple-prompt-docker
```

```
Rajiv
```

```
Hello and Welcome Rajiv
```

```
docker run -it      rajiv/simple-prompt-docker
```

```
Welcome! Please enter your name: Rajiv
```

```
Hello and Welcome Rajiv
```

Run – PORT mapping

```
docker run      rajiv/webapp
* Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
```

http://172.17.0.2:5000

Internal IP

```
docker run -p 80:5000      rajiv/simple-webapp
```

```
docker run -p 8000:5000    rajiv/simple-webapp
```

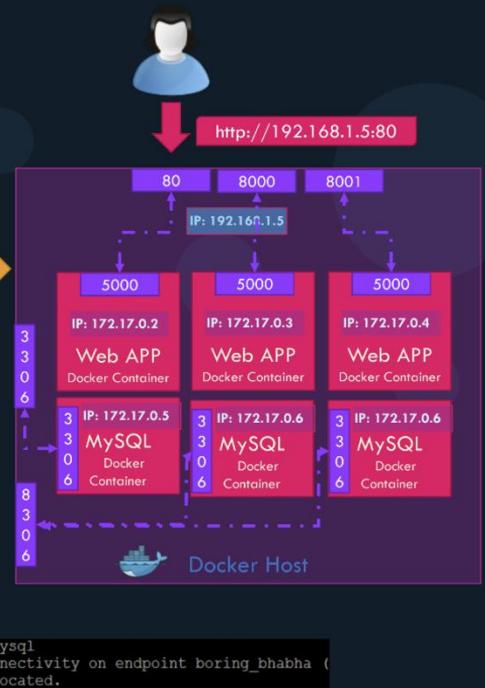
```
docker run -p 8001:5000    rajiv/simple-webapp
```

```
docker run -p 3306:3306    mysql
```

```
docker run -p 8306:3306    mysql
```

```
docker run -p 8306:3306    mysql
```

```
root@osboxes:~# docker run -p 8306:3306 -e MYSQL_ROOT_PASSWORD=pass mysql
docker: Error response from daemon: driver failed programming external connectivity on endpoint boring_bhabha (5079d342b7e8ee1c71d46): Bind for 0.0.0.0:8306 failed: port is already allocated.
```

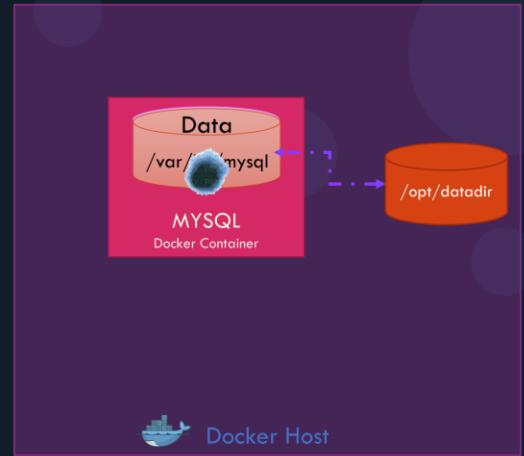


RUN – Volume mapping

```
docker run mysql
```

```
docker stop mysql
docker rm mysql
```

```
docker run -v /opt/datadir:/var/lib/mysql mysql
```



Inspect Container

```
▶ docker inspect blissful_hopper
```

```
[  
  {  
    "Id": "35505f7810d17291261a43391d4b6c0846594d415ce4f4d0a6ffbf9cc5109048",  
    "Name": "/blissful_hopper",  
    "Path": "python",  
    "Args": [  
      "app.py"  
    ],  
    "State": {  
      "Status": "running",  
      "Running": true,  
    },  
    "Mounts": [],  
    "Config": {  
      "Entrypoint": [  
        "python",  
        "app.py"  
      ],  
      "NetworkSettings": {...}  
    }  
  }  
]
```

Container Logs

```
▶ docker logs blissful_hopper
```

This is a sample web application that displays a colored background.
A color can be specified in two ways.

1. As a command line argument with --color as the argument. Accepts one of red,green,blue,blue2,pink,darkblue
2. As an Environment variable APP_COLOR. Accepts one of red,green,blue,blue2,pink,darkblue
3. If none of the above then a random color is picked from the above list.
Note: Command line argument precedes over environment variable.

No command line argument or environment variable. Picking a Random Color =blue
* Serving Flask app "app" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:8080/ (Press CTRL+C to quit)

Environment Variables

app.py

```
import os
from flask import Flask

app = Flask(__name__)

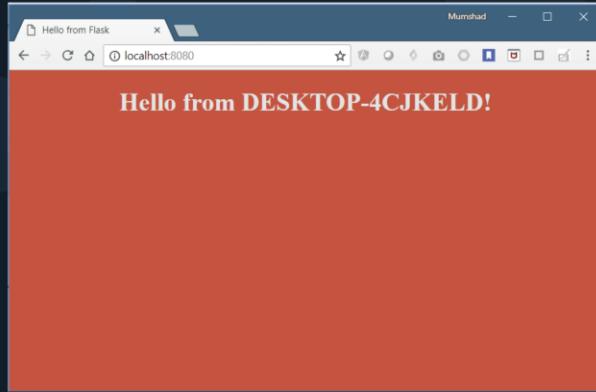
...
...

color = "red"

@app.route("/")
def main():
    print(color)
    return render_template('hello.html', color=color)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="8080")
```

▶ python app.py



Environment Variables

app.py

```
import os
from flask import Flask

app = Flask(__name__)

...
...

color = "red"

@app.route("/")
def main():
    print(color)
    return render_template('hello.html', color=color)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="8080")
```

Environment Variables

```
app.py
```

```
import os
from flask import Flask

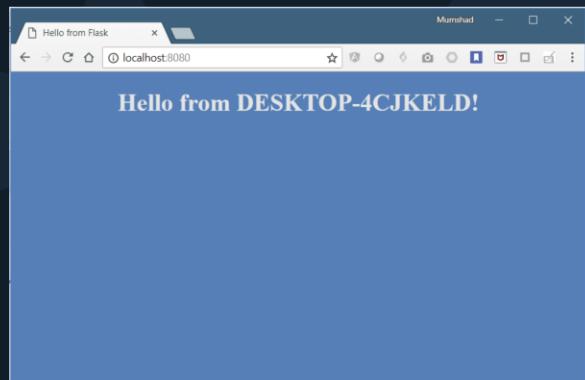
app = Flask(__name__)

...
...

color = os.environ.get('APP_COLOR')

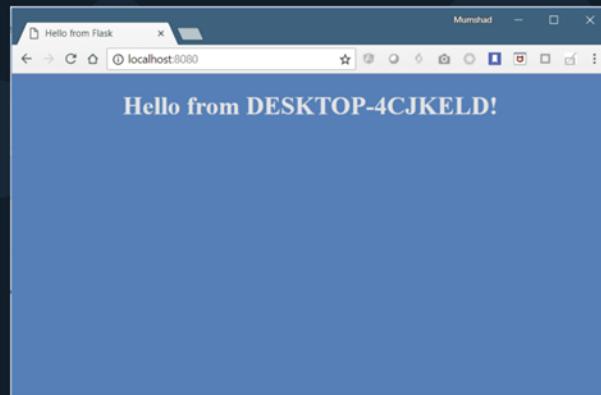
@app.route("/")
def main():
    print(color)
    return render_template('hello.html', color=color)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port="8080")
```



```
▶ export APP_COLOR=blue; python app.py
```

ENV Variables in Docker



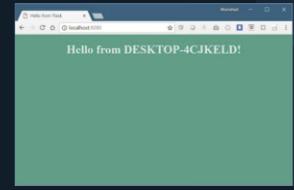
```
▶ docker run -e APP_COLOR=blue
```

ENV Variables in Docker

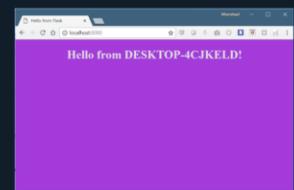
```
▶ docker run -e APP_COLOR=blue simple-webapp-color
```



```
▶ docker run -e APP_COLOR=green simple-webapp-color
```



```
▶ docker run -e APP_COLOR=pink simple-webapp-color
```



Inspect Environment Variable

```
▶ docker inspect blissful_hopper
```

```
[  
  {  
    "Id": "35505f7810d17291261a43391d4b6c0846594d415ce4f4d0a6ffbf9cc5109048",  
    "State": {  
      "Status": "running",  
      "Running": true,  
    },  
  
    "Mounts": [],  
    "Config": {  
      "Env": [  
        "APP_COLOR=blue",  
        "LANG=C.UTF-8",  
        "GPG_KEY=0D96DF4D4110E5C43FBFB17F2D347EA6AA65421D",  
        "PYTHON_VERSION=3.6.6",  
        "PYTHON_PIP_VERSION=18.1"  
      ],  
      "Entrypoint": [  
        "python",  
        "app.py"  
      ],  
    }  
  }  
]
```