

```

jegan@tektutor:~/devops-dec-2023/Day1      x   jegan@tektutor:~/devops-dec-2023      x   jegan@tektutor:/tmp/mysql      x
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| tektutor |
+-----+
5 rows in set (0.01 sec)

mysql> USE tektutor;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_tektutor |
+-----+
| training |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM training;
+-----+-----+
| id | name          | duration |
+-----+-----+
| 1  | Kubernetes    | 5 Days   |
| 2  | Microservices using Spring Boot Framework | 5 Days   |
+-----+-----+
2 rows in set (0.00 sec)

mysql> █

```

Day 2 - Docker

Processor Packaging

- Two of types of Packing SCM
 - Single Chip Module
 - a single IC can host only one Processor MCM
 - Multiple Chip Module
 - a single IC can host multiple Processors
 - in a single CPU Socket, if we install a Processor (MCM) - this means 1 CPU Socket could support multiple Processors

Server Grade Motherboards

- they support Multi-socket

Processor with multiple CPU Cores

- maximum CPU cores you have seen in a server grade processor
- 256 cpu cores per processor

What is Hypervisor ?

- Virtualization Technology
- helps us run multiple Operating System on the same laptop/desktop/workstation/server
- mutiple OS can be active at the same time on the same machine
- this is a Hardware + Software technology
- We need Processor that supports Virtualization
- AMD processor
 - Virtualization feature supported by AMD Processor is AMD-V

- Intel processor
 - Virtualization feature supported by Intel Processor is VT-X
- There are two types of hypervisors softwares
 - Type 1
 - Type 2
- Type 1 Hypervisor
 - used in servers/workstations
 - this doesn't require an Operating System to be installed to create Virtual Machines
 - this is also called as Bare Metal Hypervisor
 - Examples
 - VMWare vSphere/vCenter
- Type 2 Hypervisor
 - used in laptops/desktops/workstations
 - this requires an OS to install the Hypervisor
 - Examples
 - VMWare
 - Fusion (Mac OS-X) - requires license
 - Workstation (Windows & Linux) - requires license
 - Microsoft Hyper-V (Windows) - comes with server grade windows
 - Oracle VirtualBox (Linux, Windows & Mac) - Free
 - KVM (Linux) - open source & Free
- this type of virtualization is considered heavy weight as each OS that we install in a Virtual Machine requires dedicated hardware resources
 - we need to allocate CPU Cores to each VM
 - we need to allocate RAM to each VM
 - we need to allocate Storage to each VM
 - Virtual machine also gets virtual network card and virtual graphics cards, etc.,
- each Virtual Machine represents one fully functional Operating System, these OS that runs inside the Virtual Machine is referred as Guest OS
- In case of Type 2 Virtualization software, the OS on which the virtualization software is installed is referred as Host OS

Advantages

- to host 1000 OS as VMs, how many minimal physical servers are required
- Technically, 1 Physical server can host 1000+ Virtual machines
-

How many physical servers are required to support 1000 OS without using Virtualization software/technology?

- 1000 Physical servers
- We need a data-center(server room) that can occupy 1000 servers
 - Real estate cost involved (rent/lease)
 - Servers tend to create lot of noise (sound proofing cost involved)
 - Servers tend to create lot of heat (Air Conditioning that runs 24x7 - cost involved)
 - Servers are power hungry, they will consume lot of electricity

- Electricity Bill
- Assume that the server motherboard has 4 CPU Sockets
- If we install MCM based Processor, i.e each IC supporting let's say 4 processors
- 16 Processors and each Processor supporting 256 CPU Cores
- total cores = $16 \times 256 = 4096$ Physical Cores
- total virtual cores = $4096 \times 2 = 8192$ virtual cores
- Virtual machines use virtual cores
 - Hyperthreading - each physical cpu core supports 2 to 4 virtual cores

Hypervisor vs Docker

High Level Architecture of Hypervisor

Linux Kernel Features that enable Container Technology

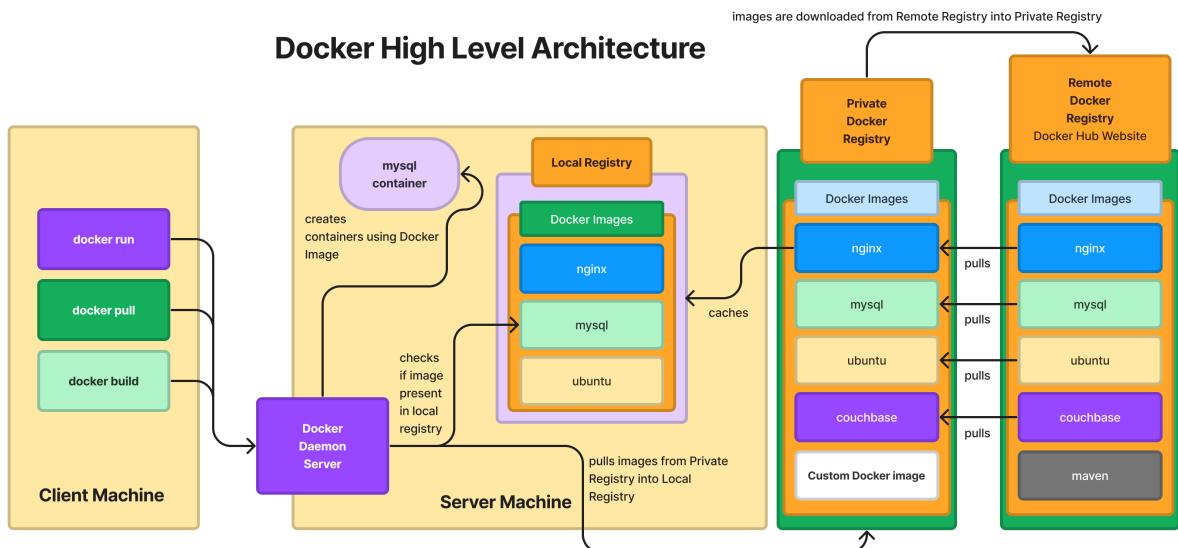
- Namespace
 - this helps in isolating one container from other containers
- Control Groups (CGroups)
 - this helps in applying some resource quota restrictions for containers

Docker Overview

- light-weight application virtualization technology
- each container represents one running application
- containers don't run OS
- containers don't get their own dedicated hardware resources
- containers don't have their own OS Kernel
- containers are just application process that runs in separate namespace
- containers will never be able to replace Operating System or Virtual Machines
- containers are not competing technology to Virtualization, they are complementing technology.
Meaning, they can be used in combination.
- each container represents one application
- each container runs in a separate namespace
- each container has its own virtual network stack
- hence, every container gets one or more IP Addresses
- containers have its own file system
- containers are created using Container Images
- Container Images come with pre-loaded/pre-installed applications

- Whichever software are there in the Container Image, are available in a ready to use fashion on the containers

High Level Architecture of Docker



What is Container Engine?

- high-level software which is also end-user friendly
- Container Engines depends on container Runtime softwares to manage containers
- as this is user-friendly, end-users like us tend to use this software instead of Container Runtime
- Example:
 - Docker is a Container Engine which internally depends on Containerd that in turn depends on runC Container Runtime
 - Podman is a Container Engine which internally depends on CRI-O Container Runtime

What is a Container Runtime?

- is a low-level software that knows how to manage containers
- manage containers
 - create containers in background/foreground
 - list containers
 - delete containers
 - start/stop/restart/kill/abort containers
- it is not so user-friendly software, hence normally no end-users use this software directly
- Examples
 - runC
 - CRI-O
 - rkt
 - LXC

Docker Alternatives

- Podman
- Containerd

Installing Docker in Windows

<https://docs.docker.com/desktop/install/windows-install/>

Installing Docker in Ubuntu

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

Installing Docker in CentOS

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

Installing Docker in Mac

<https://docs.docker.com/desktop/install/mac-install/>

What is Docker Image?

- Docker Image is similar to window iso DVD image we download from Microsoft website
- We can burn a DVD with windows iso, with that Windows Image DVD we can install Windows on any number of Laptops/PC
- Docker Image is similar to ISO images
- Docker Images container some pre-installed softwares
- With Docker Image we can create as many containers as we need
- a blueprint of a Container

What is Docker Container?

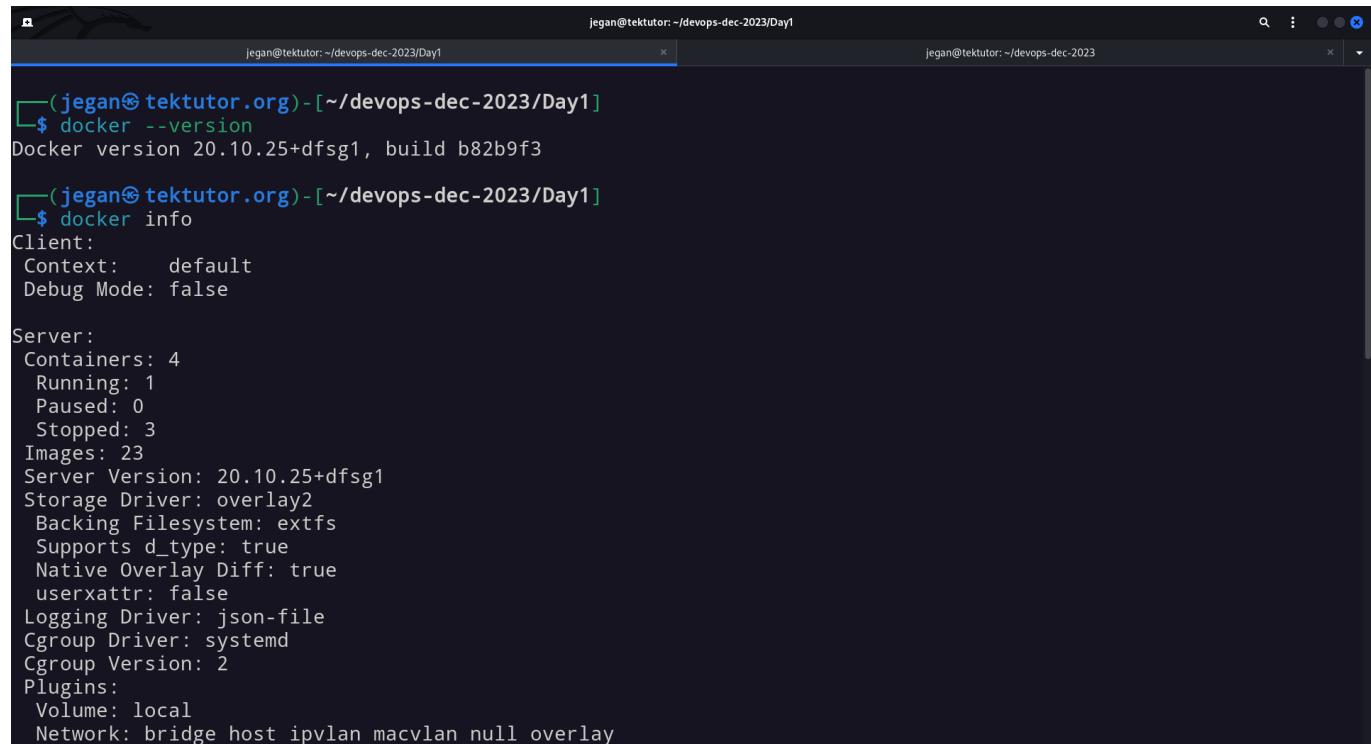
- Container is a running instance of a Docker Image
- When we create a container using a particular Docker Image, that containers gets an unique name and hostname
- The container will be in any one of following state
 - created
 - running
 - exited
- The container gets an Private IP address
- The container will have all the softwares installed on the Docker Image

Docker Commands

Lab - Finding the docker version

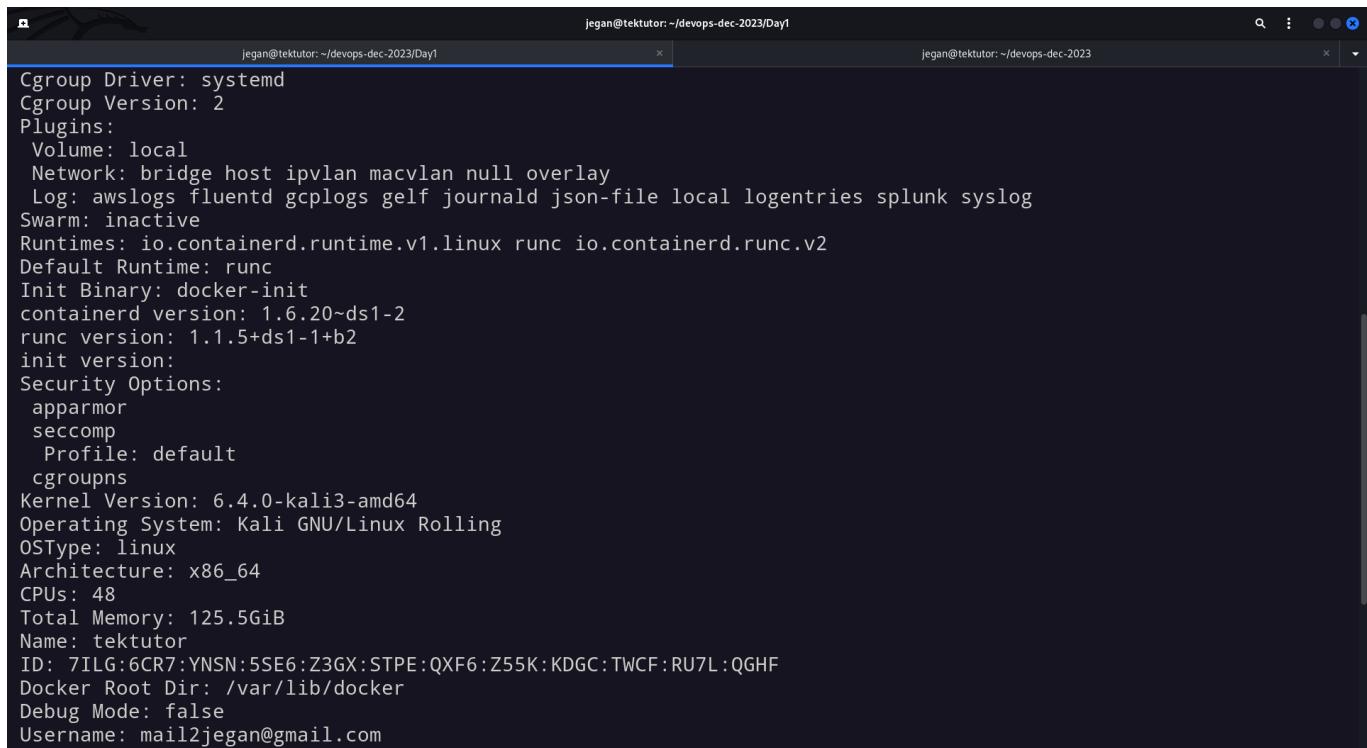
```
docker --version  
docker info
```

Expected output



The screenshot shows a terminal window with two tabs. The left tab is titled 'jegan@tektutor: ~/devops-dec-2023/Day1' and the right tab is titled 'jegan@tektutor: ~/devops-dec-2023'. The terminal displays the following command and its output:

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker --version  
Docker version 20.10.25+dfsg1, build b82b9f3  
  
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker info  
Client:  
  Context:    default  
  Debug Mode: false  
  
Server:  
  Containers: 4  
    Running: 1  
    Paused: 0  
    Stopped: 3  
  Images: 23  
  Server Version: 20.10.25+dfsg1  
  Storage Driver: overlay2  
    Backing Filesystem: extfs  
    Supports d_type: true  
    Native Overlay Diff: true  
    userxattr: false  
  Logging Driver: json-file  
  Cgroup Driver: systemd  
  Cgroup Version: 2  
  Plugins:  
    Volume: local  
    Network: bridge host ipvlan macvlan null overlay
```



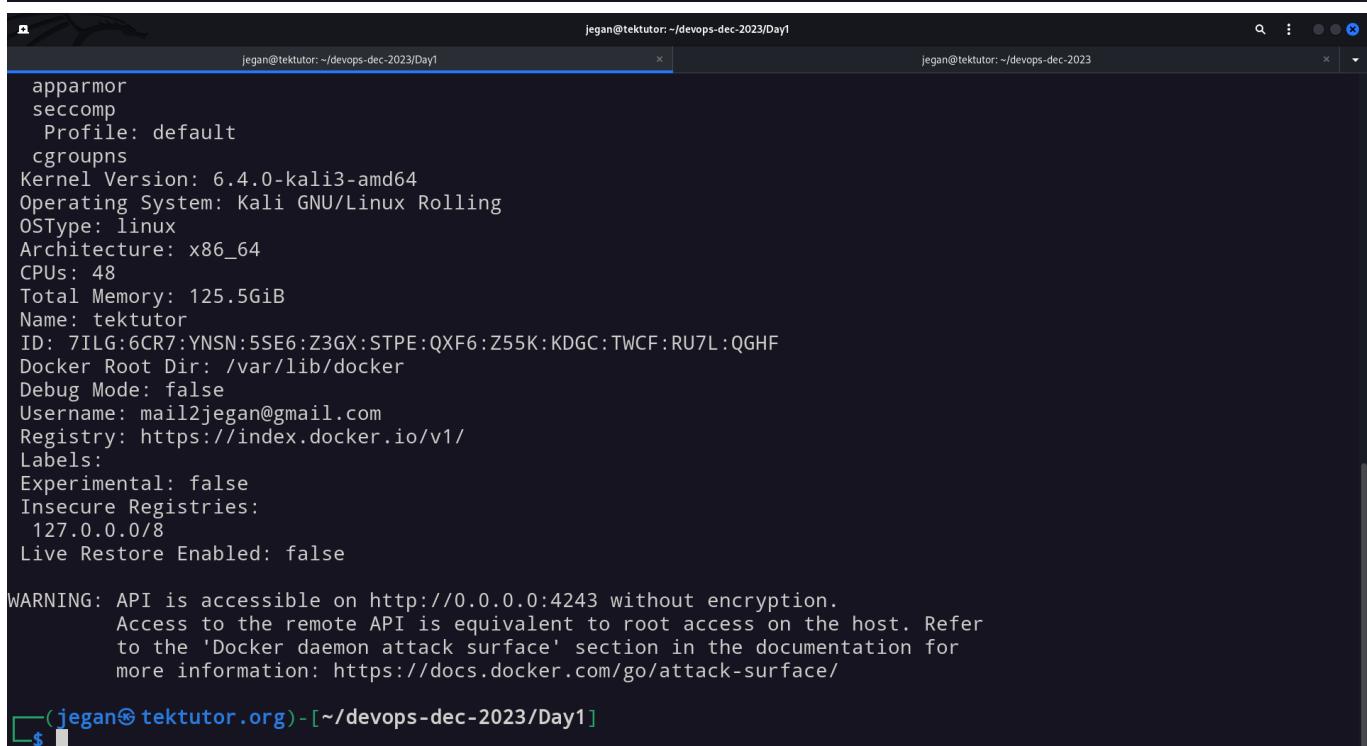
```
jegan@tektutor:~/devops-dec-2023/Day1
jegan@tektutor:~/devops-dec-2023

Cgroup Driver: systemd
Cgroup Version: 2
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
Swarm: inactive
Runtimes: io.containerd.runtime.v1.linux runc io.containerd.runc.v2
Default Runtime: runc
Init Binary: docker-init
containerd version: 1.6.20~ds1-2
runc version: 1.1.5+ds1-1+b2
init version:
Security Options:
  apparmor
  seccomp
    Profile: default
  cgroupsns
Kernel Version: 6.4.0-kali3-amd64
Operating System: Kali GNU/Linux Rolling
OSType: linux
Architecture: x86_64
CPUs: 48
Total Memory: 125.5GiB
Name: tektutor
ID: 7ILG:6CR7:YNSN:5SE6:Z3GX:STPE:QXF6:Z55K:KDGC:TWCF:RU7L:QGHF
Docker Root Dir: /var/lib/docker
Debug Mode: false
Username: mail2jegan@gmail.com

jegan@tektutor:~/devops-dec-2023/Day1
jegan@tektutor:~/devops-dec-2023

apparmor
seccomp
  Profile: default
cgroupsns
Kernel Version: 6.4.0-kali3-amd64
Operating System: Kali GNU/Linux Rolling
OSType: linux
Architecture: x86_64
CPUs: 48
Total Memory: 125.5GiB
Name: tektutor
ID: 7ILG:6CR7:YNSN:5SE6:Z3GX:STPE:QXF6:Z55K:KDGC:TWCF:RU7L:QGHF
Docker Root Dir: /var/lib/docker
Debug Mode: false
Username: mail2jegan@gmail.com
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Live Restore Enabled: false

WARNING: API is accessible on http://0.0.0.0:4243 without encryption.
Access to the remote API is equivalent to root access on the host. Refer
to the 'Docker daemon attack surface' section in the documentation for
more information: https://docs.docker.com/go/attack-surface/
```



```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

The default location of Docker Local Registry is /var/lib/docker

Lab - Listing docker images in your docker local registry

```
docker images
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023
```

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
tektutor/hello-microservice    v2      fd0985e6a33e  6 days ago   408MB
tektutor/hello-microservice    v1      3a8b73745255  6 days ago   408MB
tektutor/openshift-maven      latest   11f8b18e433e  7 days ago   408MB
bitnami/postgresql           latest   54c74bafb64f  9 days ago   274MB
releases-docker.jfrog.io/jfrog/artifactory-oss  latest   d7bff3713f0d  2 weeks ago  2.19GB
tektutor/maven               latest   41537e04c0f5  2 months ago  408MB
registry.access.redhat.com/ubi8/openjdk-11        latest   c71cc6972823  3 months ago  391MB
nginx                      latest   eea7b3dcba7e  3 months ago  187MB
ubuntu                     22.04   c6b84b685f35  3 months ago  77.8MB
mysql                      latest   99afc808f15b  3 months ago  577MB
hello-world                 latest   9c7a54a9a43c  7 months ago  13.3kB
centos                     7.9.2009  eeb6ee3f44bd  2 years ago   204MB
centos                     centos7.9.2009  eeb6ee3f44bd  2 years ago   204MB
ubuntu                     16.04   b6f507652425  2 years ago   135MB
maven                      3.6.3-jdk-11  e23b595c92ad  2 years ago   658MB
google/pause                latest   f9d5de079539  9 years ago   240kB
```

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Lab - Deleting a Docker Image from your local docker registry

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

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023
```

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED       SIZE
tektutor/hello-microservice    v2      fd0985e6a33e  6 days ago   408MB
tektutor/hello-microservice    v1      3a8b73745255  6 days ago   408MB
tektutor/openshift-maven      latest   11f8b18e433e  7 days ago   408MB
bitnami/postgresql           latest   54c74bafb64f  9 days ago   274MB
releases-docker.jfrog.io/jfrog/artifactory-oss  latest   d7bff3713f0d  2 weeks ago  2.19GB
tektutor/maven               latest   41537e04c0f5  2 months ago  408MB
registry.access.redhat.com/ubi8/openjdk-11        latest   c71cc6972823  3 months ago  391MB
nginx                      latest   eea7b3dcba7e  3 months ago  187MB
ubuntu                     22.04   c6b84b685f35  3 months ago  77.8MB
mysql                      latest   99afc808f15b  3 months ago  577MB
hello-world                 latest   9c7a54a9a43c  7 months ago  13.3kB
centos                     7.9.2009  eeb6ee3f44bd  2 years ago   204MB
centos                     centos7.9.2009  eeb6ee3f44bd  2 years ago   204MB
ubuntu                     16.04   b6f507652425  2 years ago   135MB
maven                      3.6.3-jdk-11  e23b595c92ad  2 years ago   658MB
google/pause                latest   f9d5de079539  9 years ago   240kB
```

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker rmi hello-world:latest
Untagged: hello-world:latest
Untagged: hello-world@sha256:c79d06dfdf3d3eb04caf0dc2bacab0992ebc243e083cab208bac4dd7759e0
Deleted: sha256:9c7a54a9a43cca047013b82af109fe963fde787f63f9e016fd3384500c2823d
Deleted: sha256:01bb4fce3eb1b56b05adf99504dafd31907a5aadac736e36b27595c8b92f07f1
```

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Lab - Downloading docker image from Docker Remote Registry to Docker Local Registry

```
docker images
docker pull hello-world:latest
docker images
```

Expected output

The screenshot shows a terminal window with two tabs. The left tab shows the command `docker pull hello-world:latest` being run, followed by its output: "latest: Pulling from library/hello-world", "719385e32844: Pull complete", "Digest: sha256:c79d06dfdf3d3eb04cafd0dc2bacab0992ebc243e083cabe208bac4dd7759e0", "Status: Downloaded newer image for hello-world:latest", and "docker.io/library/hello-world:latest". The right tab shows the command `docker images` being run, displaying a table of images with columns: REPOSITORY, TAG, IMAGE ID, CREATED, and SIZE. The `hello-world` image is highlighted in green, showing it was created 7 months ago and is 13.3kB in size. Other images listed include tektutor/hello-microservice (v2, v1), tektutor/openshift-maven, bitnami/postgresql, releases-docker.jfrog.io/jfrog/artifactory-oss, tektutor/maven, registry.access.redhat.com/ubi8/openjdk-11, nginx, ubuntu (22.04), mysql, centos, maven, google/pause.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
tektutor/hello-microservice	v2	fd0985e6a33e	6 days ago	408MB
tektutor/hello-microservice	v1	3a8b73745255	6 days ago	408MB
tektutor/openshift-maven	latest	11f8b18e433e	7 days ago	408MB
bitnami/postgresql	latest	54c74bafb64f	9 days ago	274MB
releases-docker.jfrog.io/jfrog/artifactory-oss	latest	d7bff3713f0d	2 weeks ago	2.19GB
tektutor/maven	latest	41537e04c0f5	2 months ago	408MB
registry.access.redhat.com/ubi8/openjdk-11	latest	c71cc6972823	3 months ago	391MB
nginx	latest	eea7b3dcba7e	3 months ago	187MB
ubuntu	22.04	c6b84b685f35	3 months ago	77.8MB
mysql	latest	99afc808f15b	3 months ago	577MB
hello-world	latest	9c7a54a9a43c	7 months ago	13.3kB
centos	7.9.2009	eeb6ee3f44bd	2 years ago	204MB
centos	centos7.9.2009	eeb6ee3f44bd	2 years ago	204MB
ubuntu	16.04	b6f507652425	2 years ago	135MB
maven	3.6.3-jdk-11	e23b595c92ad	2 years ago	658MB
google/pause	latest	f9d5de079539	9 years ago	240kB

Lab - Creating a container in background

```
docker run -dit --name ubuntu1 --hostname ubuntu1 ubuntu:16.04 /bin/bash
```

The above command will create a new container and start running the container

Things to note

d - stands for deattached/daemon i.e runs the container in the background
it- interactive terminal
name - ubuntu1 is the name of the container we have assigned
hostname - ubuntu1 is the hostname we have assigned for the container
ubuntu:16.04 - is the docker image we downloaded from Docker Hub Remote Registry
16.04 - is the tag/version of ubuntu we downloaded from Docker Hub Remote Registry
/bin/bash - is the shell we launched inside the container

Each time we execute the docker run command, it would create a new container and starts that, hence we must provide a unique container name and hostname.

As long as the bash shell continues to run inside the container, the container will be running. If we exit the bash shell inside the container then it will exit container as well, leading to stopping the container.

Let us list the docker images to check if the above command has download the ubuntu:16.04 image from Docker Hub - Remote Registry to Local Docker Registry

```
docker images
```

Expected output

The screenshot shows a terminal window with two tabs. The left tab shows the command `docker run -dit --name ubuntu1 --hostname ubuntu1 ubuntu:16.04 /bin/bash` being run, which successfully pulls the image and creates a container named `ubuntu1`. The right tab shows the output of the `docker images` command, listing various images including the newly created `ubuntu:16.04` image.

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
tektutor/hello-microservice	v2	fd0985e6a33e	6 days ago	408MB
tektutor/hello-microservice	v1	3a8b73745255	6 days ago	408MB
tektutor/openshift-maven	latest	11f8b18e433e	7 days ago	408MB
bitnami/postgresql	latest	54c74bafb64f	9 days ago	274MB
releases-docker.jfrog.io/jfrog/artifactory-oss	latest	d7bff3713f0d	2 weeks ago	2.19GB
tektutor/maven	latest	41537e04c0f5	2 months ago	408MB
registry.access.redhat.com/ubi8/openjdk-11	latest	c71cc6972823	3 months ago	391MB
nginx	latest	eea7b3dcba7e	3 months ago	187MB
hello-world	latest	9c7a54a9a43c	7 months ago	13.3kB
centos	7.9.2009	eeb6ee3f44bd	2 years ago	204MB
centos	centos7.9.2009	eeb6ee3f44bd	2 years ago	204MB
ubuntu	16.04	b6f507652425	2 years ago	135MB
maven	3.6.3-jdk-11	e23b595c92ad	2 years ago	658MB
google/pause	latest	f9d5de079539	9 years ago	240kB

Finding if the `ubuntu1` container that we created is running

```
docker ps
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker images
REPOSITORY           TAG      IMAGE ID      CREATED     SIZE
tektutor/hello-microservice    v2      fd0985e6a33e  6 days ago  408MB
tektutor/hello-microservice    v1      3a8b73745255  6 days ago  408MB
tektutor/openshift-maven      latest   11f8b18e433e  7 days ago  408MB
bitnami/postgresql           latest   54c74ba64f   9 days ago  274MB
releases-docker.jfrog.io/jfrog/artifactory-oss  latest   d7bff3713f0d  2 weeks ago  2.19GB
tektutor/maven               latest   41537e04c0f5  2 months ago 408MB
registry.access.redhat.com/ubi8/openjdk-11       latest   c71cc6972823  3 months ago 391MB
nginx                      latest   eea7b3dcba7e  3 months ago 187MB
hello-world                 latest   9c7a54a9a43c  7 months ago 13.3kB
centos                      7.9.2009  eeb6ee3f44bd  2 years ago  204MB
centos                      centos7.9.2009  eeb6ee3f44bd  2 years ago  204MB
ubuntu                       16.04    b6f507652425  2 years ago  135MB
maven                        3.6.3-jdk-11  e23b595c92ad  2 years ago  658MB
google/pause                 latest   f9d5de079539  9 years ago  240kB

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED     NAMES
6f8bcb3044f0   ubuntu:16.04   "/bin/bash"              32 seconds ago  ubuntu1
Up 31 seconds
88afffc63c0d9  releases-docker.jfrog.io/jfrog/artifactory-oss:latest  "/entrypoint-artifac..."  22 hours ago
Up 22 hours     0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp   jfrog

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Lab - Stopping a running container

```
docker ps
docker stop ubuntu1
docker ps
docker ps -a
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED     NAMES
6f8bcb3044f0   ubuntu:16.04   "/bin/bash"              32 seconds ago  ubuntu1
Up 31 seconds
88afffc63c0d9  releases-docker.jfrog.io/jfrog/artifactory-oss:latest  "/entrypoint-artifac..."  22 hours ago
Up 22 hours     0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp   jfrog

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker stop ubuntu1
ubuntu1

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED     NAMES
88afffc63c0d9  releases-docker.jfrog.io/jfrog/artifactory-oss:latest  "/entrypoint-artifac..."  23 hours ago   U
Up 23 hours     0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp   jfrog

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED     NAMES
6f8bcb3044f0   ubuntu:16.04   "/bin/bash"              23 minutes ago  ubuntu1
Exited (0) 16 seconds ago
88afffc63c0d9  releases-docker.jfrog.io/jfrog/artifactory-oss:latest  "/entrypoint-artifac..."  23 hours ago
Up 23 hours     0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp   jfrog
a8ee6f47355a   bitnami/postgresql:latest   "/opt/bitnami/script..."  6 days ago
```

Lab - Start an exited container

```
docker ps -a  
docker start ubuntu1  
docker ps
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1  
jegan@tektutor: ~/devops-dec-2023  
88afffc63c0d9 releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 23 hours ago U  
p 23 hours 0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog  
└─(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker ps -a  
CONTAINER ID IMAGE COMMAND CREATED NAMES  
STATUS PORTS  
6f8bcb3044f0 ubuntu:16.04 "/bin/bash" 23 minutes ago ubuntu1  
Exited (0) 16 seconds ago  
88afffc63c0d9 releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 23 hours ago  
Up 23 hours 0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog  
a8ee6f47355a bitnami/postgresql:latest "/opt/bitnami/script..." 6 days ago pg  
Exited (0) 4 days ago  
└─(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker start ubuntu1  
ubuntu1  
└─(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker ps  
CONTAINER ID IMAGE COMMAND CREATED NAMES  
STATUS PORTS  
6f8bcb3044f0 ubuntu:16.04 "/bin/bash" 27 minutes ago ubuntu1  
Up 1 second  
88afffc63c0d9 releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 23 hours ago  
Up 23 hours 0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog  
└─(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$
```

Lab - Restarting a running container

```
docker ps  
docker restart ubuntu1  
docker ps
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker start ubuntu1
ubuntu1

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker ps
CONTAINER ID IMAGE STATUS PORTS COMMAND NAMES CREATED
6f8bcb3044f0 ubuntu:16.04 Up 1 second releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 27 minutes ago
88afffc63c0d9 jfrog 0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog Up 23 hours

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker restart ubuntu1
ubuntu1

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker ps
CONTAINER ID IMAGE STATUS PORTS COMMAND NAMES CREATED
6f8bcb3044f0 ubuntu:16.04 Up 3 seconds releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 29 minutes ago
88afffc63c0d9 jfrog 0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog Up 23 hours

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]$
```

Lab - Deleting a running container

```
docker ps
docker rm ubuntu1
docker stop ubuntu1
docker rm ubuntu1
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker ps
CONTAINER ID IMAGE STATUS PORTS COMMAND NAMES CREATED
6f8bcb3044f0 ubuntu:16.04 Up 3 seconds releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 29 minutes ago
88afffc63c0d9 jfrog 0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog Up 23 hours

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker rm ubuntu1
Error response from daemon: You cannot remove a running container 6f8bcb3044f078e69eff50bc5ba4acecb1acaf606793672c8506315b898f3ce. Stop the container before attempting removal or force remove

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker stop ubuntu1
ubuntu1

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker rm ubuntu1
ubuntu1

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]-$ docker ps -a
CONTAINER ID IMAGE STATUS PORTS COMMAND NAMES CREATED S
TATUS
88afffc63c0d9 releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 23 hours ago U
p 23 hours 0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog
a8ee6f47355a bitnami/postgresql:latest "/opt/bitnami/script... pg" 6 days ago E
xited (0) 4 days ago

[jegan@tektutor.org]-(~/devops-dec-2023/Day1]
[jegan@tektutor.org]$
```

Lab - Forcibly deleting a running container

```
docker ps
docker run -dit --name ubuntu1 --hostname ubuntu1 ubuntu:16.04 /bin/bash
docker ps
docker rm -f ubuntu1
docker ps
```

Expected output

The screenshot shows a terminal window with two tabs. The active tab is titled 'jegan@tektutor: ~/devops-dec-2023/Day1'. The session history is as follows:

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -dit --name ubuntu1 --hostname ubuntu1 ubuntu:16.04 /bin/bash
4876ac7e29160f3cd76ce4df1322bb3b48ed32b1dd8e65435374719dea12b6ce

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID IMAGE COMMAND NAMES STATUS PORTS CREATED
4876ac7e2916 ubuntu:16.04 "/bin/bash" ubuntu1 Up 2 seconds
88afffc63c0d9 releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." Up 23 hours
0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker rm -f ubuntu1
ubuntu1

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID IMAGE COMMAND NAMES STATUS PORTS CREATED
88afffc63c0d9 releases-docker.jfrog.io/jfrog/artifactory-oss:latest "/entrypoint-artifac..." 23 hours ago U
p 23 hours
0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp jfrog
a8ee6f47355a bitnami/postgresql:latest "/opt/bitnami/script..." 6 days ago E
xited (0) 4 days ago
pg

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Lab - Delete multiple containers forcibly with one command

```
docker ps -a
docker rm -f pg jfrog c2
docker ps -a
```

Expected output

The screenshot shows a terminal window with two tabs. The left tab is titled 'jegan@tektutor: ~/devops-dec-2023/Day1' and the right tab is 'jegan@tektutor: ~/devops-dec-2023'. The left tab contains the following command history:

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS     NAMES
374e8274f067   ubuntu:16.04   "bash"        12 minutes ago
Up 12 minutes
88affc63c0d9   releases-docker.jfrog.io/jfrog/artifactory-oss:latest   "/entrypoint-artifac..."  23 hours ago
Up 23 hours
0.0.0.0:8081-8082->8081-8082/tcp, :::8081-8082->8081-8082/tcp   jfrog
a8ee6f47355a   bitnami/postgresql:latest   "/opt/bitnami/script..."  6 days ago
Exited (0) 4 days ago

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker rm -f pg jfrog c2
pg
jfrog
c2

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID   IMAGE          COMMAND       CREATED      STATUS      PORTS     NAMES

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Lab - Deleting multiple containers forcibly without using their names

Let's first create 3 containers

```
docker run -dit --name ubuntu1 --hostname ubuntu1 ubuntu:16.04 /bin/bash
docker run -dit --name ubuntu2 --hostname ubuntu2 ubuntu:16.04 /bin/bash
docker run -dit --name ubuntu3 --hostname ubuntu3 ubuntu:16.04 /bin/bash
```

Let's list and check them if they are running

```
docker ps
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023

$ docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -dit --name ubuntu1 --hostname ubuntu1 ubuntu:16.04 /bin/bash
2e1bf2262d85de2492f4016c366c504720ee4179660deb7d49a53e11c888261d

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -dit --name ubuntu2 --hostname ubuntu2 ubuntu:16.04 /bin/bash
17acd1b5fae9e5bd50c269472d7e26fc3c8be7c1fbf9bbebc72a8d4eef5fe76
^[[A

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -dit --name ubuntu3 --hostname ubuntu3 ubuntu:16.04 /bin/bash
161dca812822cf37ebaa1c31c977453375010d65302fb3c22aa577ecd6510f01

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
161dca812822  ubuntu:16.04  "/bin/bash"  3 seconds ago  Up 1 second
17acd1b5fae9  ubuntu:16.04  "/bin/bash"  8 seconds ago  Up 7 seconds
2e1bf2262d85  ubuntu:16.04  "/bin/bash"  14 seconds ago Up 13 seconds

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -q
161dca812822
17acd1b5fae9
2e1bf2262d85
```

Let's list only the container ids of currently running containers

```
docker ps -q
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -dit --name ubuntu3 --hostname ubuntu3 ubuntu:16.04 /bin/bash
161dca812822cf37ebaa1c31c977453375010d65302fb3c22aa577ecd6510f01

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
161dca812822  ubuntu:16.04  "/bin/bash"  3 seconds ago  Up 1 second
17acd1b5fae9  ubuntu:16.04  "/bin/bash"  8 seconds ago  Up 7 seconds
2e1bf2262d85  ubuntu:16.04  "/bin/bash"  14 seconds ago Up 13 seconds

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -q
161dca812822
17acd1b5fae9
2e1bf2262d85

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker rm -f $(docker ps -q)
161dca812822
17acd1b5fae9
2e1bf2262d85

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
```

Let's now combine the above command as a sub-command to delete all running containers

```
docker rm -f $(docker ps -q)
```

Expected output

```
jegan@tektutor:~/devops-dec-2023/Day1
$ docker run -dit --name ubuntu3 --hostname ubuntu3 ubuntu:16.04 /bin/bash
161dca812822cf37ebaa1c31c977453375010d65302fb3c22aa577ecd6510f01

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS     PORTS      NAMES
161dca812822  ubuntu:16.04  "/bin/bash"  3 seconds ago  Up 1 second          ubuntu3
17acd1b5fae9  ubuntu:16.04  "/bin/bash"  8 seconds ago  Up 7 seconds          ubuntu2
2e1bf2262d85  ubuntu:16.04  "/bin/bash"  14 seconds ago Up 13 seconds         ubuntu1

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -q
161dca812822
17acd1b5fae9
2e1bf2262d85

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker rm -f $(docker ps -q)
161dca812822
17acd1b5fae9
2e1bf2262d85

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED    STATUS      PORTS      NAMES
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Lab - Creating a mysql db server container

```
docker images
docker run -d --name mysql --hostname mysql -e MYSQL_ROOT_PASSWORD=root@123
mysql:latest
docker ps
```

Expected output

```
jegan@tektutor:~/devops-dec-2023/Day1
jegan@tektutor:~/devops-dec-2023
google/pause          latest        f9d5de079539  9 years ago  240kB

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -d --name mysql --hostname mysql -e MYSQL_ROOT_PASSWORD=root@123 mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
8e0176adc18c: Pull complete
2d2c52718f65: Pull complete
d88d03ce139b: Pull complete
4a7d7f11aa1e: Pull complete
ce5949193e4c: Pull complete
f7f024dfb329: Pull complete
5fc3c840facc: Pull complete
509068e49488: Pull complete
cbc847bab598: Pull complete
942bef62a146: Pull complete
Digest: sha256:1773f3c7aa9522f0014d0ad2bbdaf597ea3b1643c64c8ccc2123c64af8b82b1
Status: Downloaded newer image for mysql:latest
1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aafa

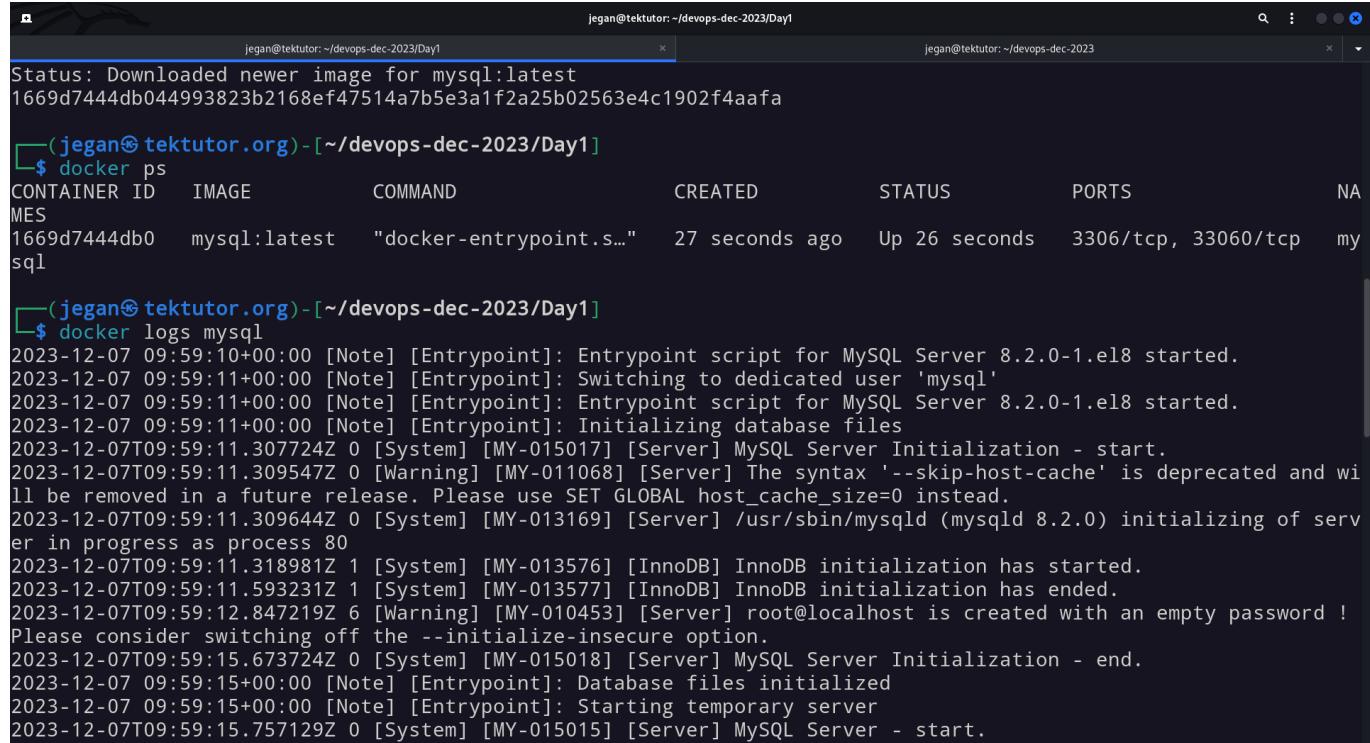
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND      CREATED    STATUS      PORTS      NAMES
MES           mysql:latest "docker-entrypoint.s..."  27 seconds ago  Up 26 seconds  3306/tcp, 33060/tcp  my
sql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Checking the mysql db server logs

```
docker logs mysql
```

Expected output



The screenshot shows a terminal window with two tabs. The left tab shows the output of the command `docker ps`, which lists a single MySQL container named `mysql`. The right tab shows the output of the command `docker logs mysql`, which displays the MySQL initialization logs.

```
jegan@tektutor:~/devops-dec-2023/Day1
Status: Downloaded newer image for mysql:latest
1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aafa

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED      STATUS      PORTS     NAMES
1669d7444db0   mysql:latest "docker-entrypoint.s..."  27 seconds ago   Up 26 seconds   3306/tcp, 33060/tcp   mysql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker logs mysql
2023-12-07 09:59:10+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.2.0-1.el8 started.
2023-12-07 09:59:11+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2023-12-07 09:59:11+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.2.0-1.el8 started.
2023-12-07 09:59:11+00:00 [Note] [Entrypoint]: Initializing database files
2023-12-07T09:59:11.307724Z 0 [System] [MY-015017] [Server] MySQL Server Initialization - start.
2023-12-07T09:59:11.309547Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please use SET GLOBAL host_cache_size=0 instead.
2023-12-07T09:59:11.309644Z 0 [System] [MY-013169] [Server] /usr/sbin/mysqld (mysqld 8.2.0) initializing of server in progress as process 80
2023-12-07T09:59:11.318981Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2023-12-07T09:59:11.593231Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2023-12-07T09:59:12.847219Z 6 [Warning] [MY-010453] [Server] root@localhost is created with an empty password ! Please consider switching off the --initialize-insecure option.
2023-12-07T09:59:15.673724Z 0 [System] [MY-015018] [Server] MySQL Server Initialization - end.
2023-12-07 09:59:15+00:00 [Note] [Entrypoint]: Database files initialized
2023-12-07 09:59:15+00:00 [Note] [Entrypoint]: Starting temporary server
2023-12-07T09:59:15.757129Z 0 [System] [MY-015015] [Server] MySQL Server - start.
```

Lab - Finding details of mysql container

```
docker ps
docker inspect mysql
```

Expected output

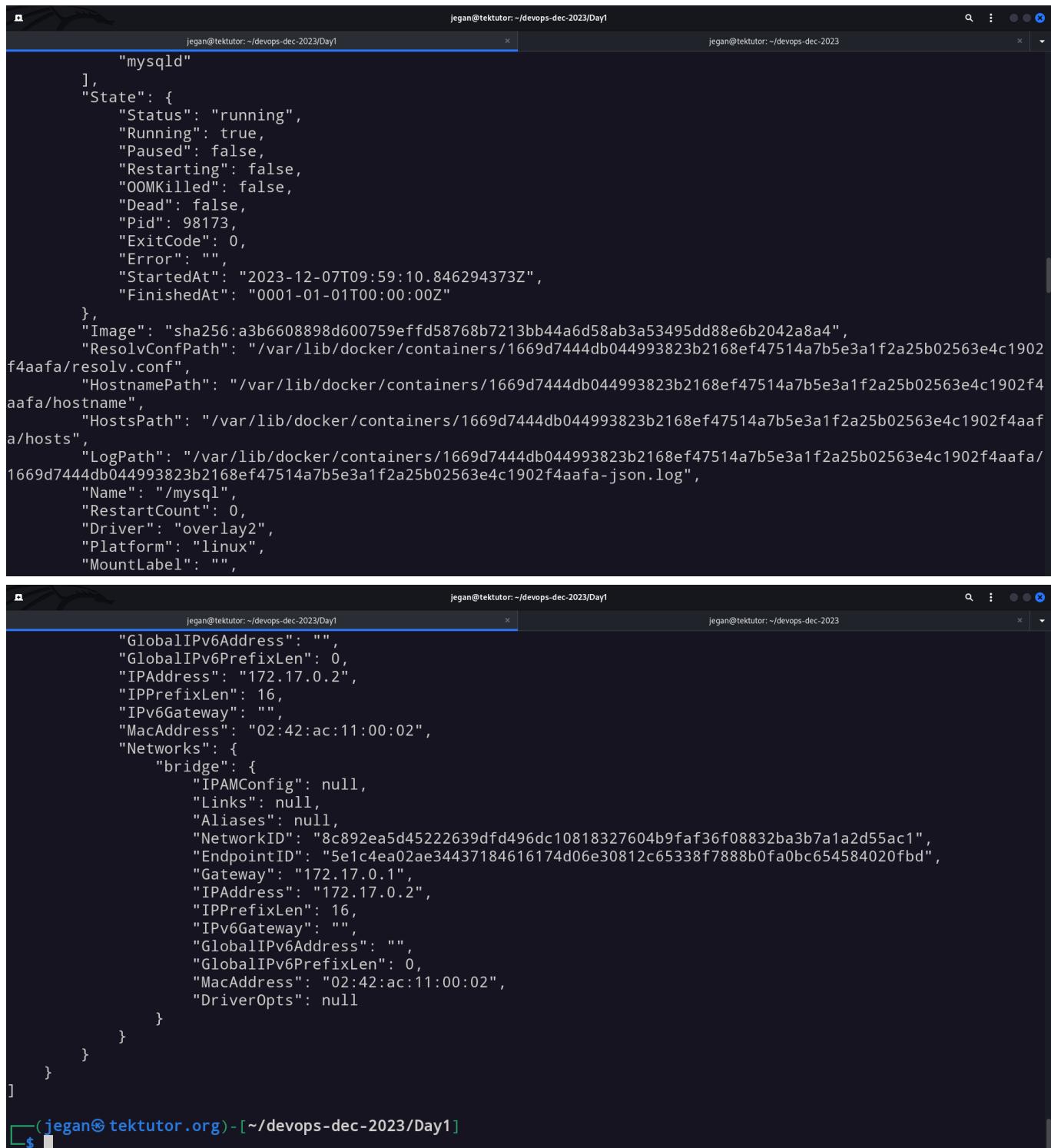
The screenshot shows a terminal window with two tabs. The left tab is titled 'jegan@tektutor: ~/devops-dec-2023/Day1' and the right tab is titled 'jegan@tektutor: ~/devops-dec-2023'. The left tab contains the following command and its output:

```
maven google/pause
      3.6.3-jdk-11    e23b595c92ad  2 years ago  658MB
      latest          f9d5de079539  9 years ago  240kB
```

The right tab shows the results of running 'docker ps' and 'docker inspect mysql' commands:

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND      CREATED     STATUS      PORTS
1669d7444db0   mysql:latest "docker-entrypoint.s..."  7 minutes ago   Up 7 minutes   3306/tcp, 33060/tcp   mysql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker inspect mysql
[{"Id": "1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aaaf", "Created": "2023-12-07T09:59:10.251208656Z", "Path": "docker-entrypoint.sh", "Args": ["mysqld"], "State": {"Status": "running", "Running": true, "Paused": false, "Restarting": false, "OOMKilled": false, "Dead": false, "Pid": 98173, "ExitCode": 0}}
```



```
jegan@tektutor:~/devops-dec-2023/Day1
jegan@tektutor:~/devops-dec-2023
```

```
"mysqld"
],
"State": {
    "Status": "running",
    "Running": true,
    "Paused": false,
    "Restarting": false,
    "OOMKilled": false,
    "Dead": false,
    "Pid": 98173,
    "ExitCode": 0,
    "Error": "",
    "StartedAt": "2023-12-07T09:59:10.846294373Z",
    "FinishedAt": "0001-01-01T00:00:00Z"
},
"Image": "sha256:a3b6608898d600759effd58768b7213bb44a6d58ab3a53495dd88e6b2042a8a4",
"ResolvConfPath": "/var/lib/docker/containers/1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aaafa/resolv.conf",
"HostnamePath": "/var/lib/docker/containers/1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aaafa/hostname",
"HostsPath": "/var/lib/docker/containers/1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aaafa/hosts",
"LogPath": "/var/lib/docker/containers/1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aaafa/1669d7444db044993823b2168ef47514a7b5e3a1f2a25b02563e4c1902f4aaafa-json.log",
"Name": "/mysql",
"RestartCount": 0,
"Driver": "overlay2",
"Platform": "linux",
"MountLabel": "",

"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:02",
"Networks": {
    "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID": "8c892ea5d45222639dfd496dc10818327604b9faf36f08832ba3b7a1a2d55ac1",
        "EndpointID": "5e1c4ea02ae34437184616174d06e30812c65338f7888b0fa0bc654584020fdb",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:11:00:02",
        "DriverOpts": null
    }
}
}
]
```

Lab - Finding the IP address of a running container

```
docker inspect mysql | grep IPA
docker inspect -f {{.NetworkSettings.IPAddress}} mysql
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023

"NetworkID": "8c892ea5d45222639df496dc10818327604b9faf36f08832ba3b7a1a2d55ac1",
"EndpointID": "5e1c4ea02ae34437184616174d06e30812c65338f7888b0fa0bc654584020fdb",
"Gateway": "172.17.0.1",
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"MacAddress": "02:42:ac:11:00:02",
"DriverOpts": null
}
}
}
]

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker inspect mysql | grep IPA
    "SecondaryIPAddresses": null,
    "IPAddress": "172.17.0.2",
    "IPAMConfig": null,
    "IPAddress": "172.17.0.2",

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker inspect -f ${.NetworkSettings.IPAddress} mysql
172.17.0.2

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Pinging the container IP from your RPS CentOS machine terminal

```
ping 172.17.0.2
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
1669d7444db0 mysql:latest "docker-entrypoint.s..." 18 minutes ago Up 18 minutes 3306/tcp, 33060/tcp mysql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker inspect -f ${.HostConfig.NetworkMode} mysql
default

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker inspect -f ${.NetworkSettings.IPAddress} mysql
172.17.0.2

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ ping 172.17.0.2
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.122 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.055 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.053 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.053 ms
64 bytes from 172.17.0.2: icmp_seq=6 ttl=64 time=0.053 ms
^C
--- 172.17.0.2 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5097ms
rtt min/avg/max/mdev = 0.053/0.065/0.122/0.025 ms

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

Lab - Finding more details about a Docker image

```
/
```

```
docker image inspect mysql:latest
```

Expected output

```
jegan@tektutor:~/devops-dec-2023/Day1
$ docker image inspect mysql:latest
[{"Id": "sha256:a3b6608898d600759effd58768b7213bb44a6d58ab3a53495dd88e6b2042a8a4", "RepoTags": ["mysql:latest"], "RepoDigests": ["mysql@sha256:1773f3c7aa9522f0014d0ad2bbdaf597ea3b1643c64c8ccc2123c64af8b82b1"], "Parent": "", "Comment": "buildkit.dockerfile.v0", "Created": "2023-10-24T16:24:49Z", "Container": "", "ContainerConfig": {"Hostname": "", "Domainname": "", "User": "", "AttachStdin": false, "AttachStdout": false, "AttachStderr": false, "Tty": false, "OpenStdin": false, "StdinOnce": false, "Env": null, "Cmd": null, "Image": ""}, "f7b04/diff": {"WorkDir": "/var/lib/docker/overlay2/6cab4cc455766e233f1b99c30d8d899688faf703f3b015973c8365f1ec2f7b04/work"}, {"Name": "overlay2"}, "RootFS": {"Type": "layers", "Layers": ["sha256:d6fe63e8be63d078aeef9739f2c7ea101e6cc1a3f998d179af63a10e7f0f959d", "sha256:69ef53c77128b4110395e0a6088180a9b721ad4e657519af3833b582be1025e3", "sha256:69a630646f651d14ff89d41075d81b1277f918cb11a4f7fe90176bc19be360e2", "sha256:9458eec006d024a88859da375e58cba647138e67c0f28929a2e55f57bd7cc059", "sha256:01cbaaa2e3b944b563eb5bbbb9c4594ea9ec4bf2c61d4004223df75c6396d3e4", "sha256:f159770d104df09d0d2f7c6ba8556cafe3a8e9de56f799c9d5d570bbb8abdb53", "sha256:235367ebc71a6828430323a7c33e0c462c6c1fd335fa917625de7145b36ff5af", "sha256:867dc881a6cfbc6311a5d77214854881768bf1c34c48fab6a0fa0c82b0c08d50", "sha256:1af6e691b2c1e8acb910e66b74cc5e6acb7a995213d796b01ad3dd5587cf9633", "sha256:eff93312e2490ea70bca3cdb18a0e653da5432ee287a1f4b8e379f378345465f"]}, "Metadata": {"LastTagTime": "0001-01-01T00:00:00Z"}}
```

Lab - Connecting to mysql server using mysql client inside the container shell

Getting inside the mysql container shell

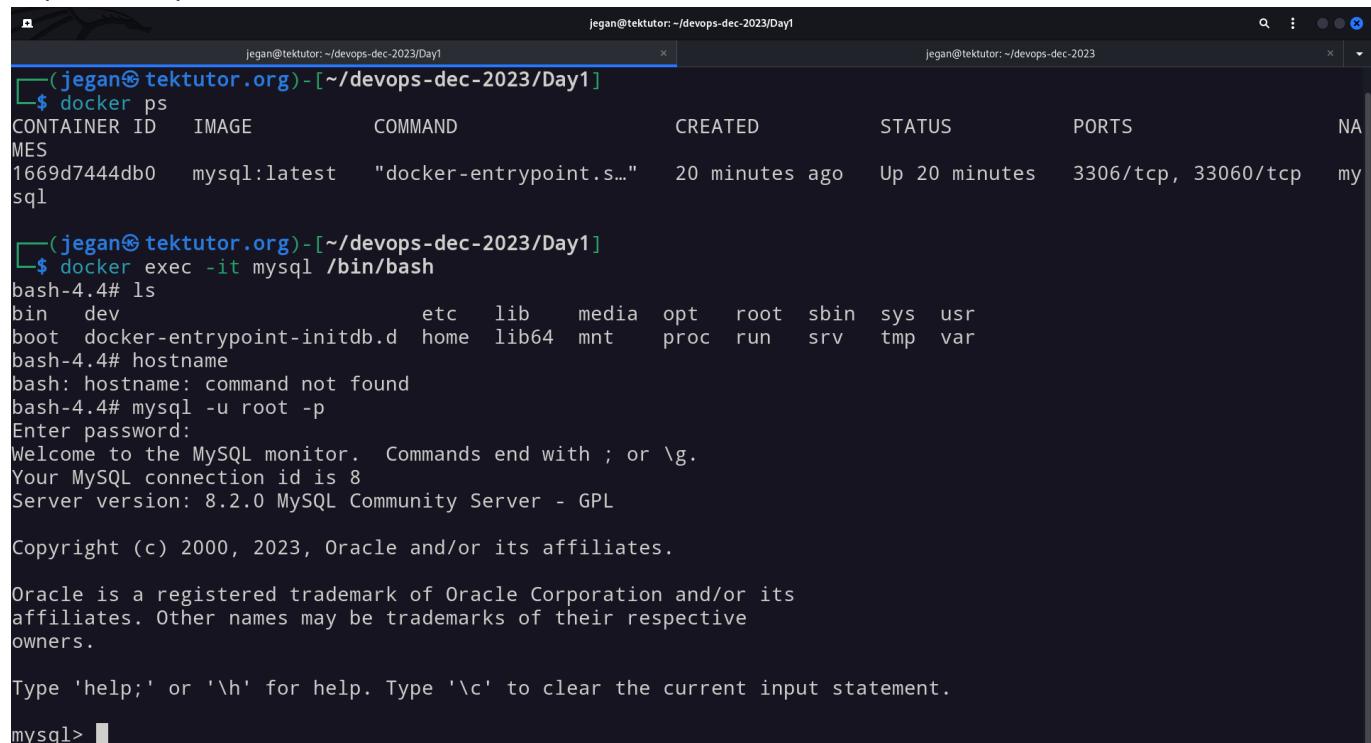
```
docker exec -it mysql /bin/bash
ls
```

Connecting to mysql server, using the mysql client that comes with the mysql container

```
mysql -u root -p
```

When it prompts for password, type 'root@123' without quotes.

Expected output



The screenshot shows a terminal window with two tabs. The active tab is titled '(jegan@tektutor.org)-[~/devops-dec-2023/Day1]'. It contains the following MySQL session:

```
jegan@tektutor:~/devops-dec-2023/Day1
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED        STATUS        PORTS
MES
1669d7444db0   mysql:latest "docker-entrypoint.s..."  20 minutes ago   Up 20 minutes   3306/tcp, 33060/tcp   my
sql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker exec -it mysql /bin/bash
bash-4.4# ls
bin  dev          etc  lib    media  opt   root  sbin  sys  usr
boot docker-entrypoint-initdb.d  home  lib64  mnt    proc  run   srv  tmp  var
bash-4.4# hostname
bash: hostname: command not found
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> █
```

Creating a database in the mysql db server and switching to the database

```
SHOW DATABASES;
CREATE DATABASE tektutor;
USE tektutor;
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)

mysql> CREATE DATABASE tektutor;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| tektutor |
+-----+
5 rows in set (0.00 sec)

mysql> USE tektutor;
```

Let's create a table and insert some records as shown below

```
SHOW TABLES;
CREATE TABLE training (id INT NOT NULL, name VARCHAR(200) NOT NULL,
duration VARCHAR(200) NOT NULL, PRIMARY KEY(id) );
INSERT INTO training VALUES ( 1, "DevOps", "5 Days" );
INSERT INTO training VALUES ( 2, "Kubernetes", "5 Days" );
SELECT * FROM training;
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023
mysql> CREATE DATABASE tektutor;
Query OK, 1 row affected (0.01 sec)

mysql> USE tektutor;
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql> CREATE TABLE training (id INT NOT NULL, name VARCHAR(200) NOT NULL, duration VARCHAR(200) NOT NULL, PRIMARY KEY(id) );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO training VALUES ( 1, "DevOps", "5 Days" );
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO training VALUES ( 2, "Kubernetes", "5 Days" );
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM training;
+---+-----+-----+
| id | name      | duration |
+---+-----+-----+
| 1 | DevOps    | 5 Days   |
| 2 | Kubernetes | 5 Days   |
+---+-----+-----+
2 rows in set (0.00 sec)

mysql> ■
```

Let's disconnect my mysql server, exit and delete the container

```
exit
exit
docker ps
docker rm -f mysql
docker ps -a
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023/Day1
| id | name      | duration |
+----+-----+-----+
| 1 | DevOps    | 5 Days   |
| 2 | Kubernetes | 5 Days   |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye
bash-4.4# exit
exit

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID        IMAGE           COMMAND       CREATED          STATUS          PORTS          NAMES
8bf7b5249a92        mysql:latest    "docker-entrypoint.s..."   4 minutes ago   Up 3 minutes   3306/tcp, 33060/tcp   mysql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker rm -f mysql
mysql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID        IMAGE           COMMAND       CREATED          STATUS          PORTS          NAMES

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```

At this point, we lost all data stored within the mysql container. The reason being, we used the container storage, which is a very bad idea. As containers are temporary resources, we shouldn't store data inside the container storage, instead we should use an external storage. We will learn the best practice in our next lab exercise.

Lab - Using host directory as an external volume within mysql container to persist data permanently

Let's create a mysql directory inside /tmp directory

```
mkdir -p /tmp/mysql
ls -lha /tmp/mysql
```

Let's see if any containers are running

```
docker ps -a
```

Let's create the mysql container using /tmp/mysql folder as the external storage within container

```
docker run -d --name mysql --hostname mysql -v /tmp/mysql:/var/lib/mysql -e  
MYSQL_ROOT_PASSWORD=root@123 mysql:latest  
docker ps
```

Expected output

The screenshot shows a terminal window with two tabs. The left tab shows the command history for creating the MySQL container:

```
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ mkdir -p /tmp/mysql  
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ ls -lha /tmp/mysql  
total 44K  
drwxr-xr-x  2 jegan jegan 4.0K Dec  7 16:36 .  
drwxrwxrwt 26 root  root  36K Dec  7 16:36 ..  
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker ps -a  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES  
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker run -d --name mysql --hostname mysql -v /tmp/mysql:/var/lib/mysql -e MYSQL_ROOT_PASSWORD=root@123 mysql:latest  
c03e715f2af8c0e04348235f097468c0817209c5228783a28f6a89532249268  
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$ docker ps  
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES  
c03e715f2af8c0e04348235f097468c0817209c5228783a28f6a89532249268  
(jegan@tektutor.org)-[~/devops-dec-2023/Day1]  
$
```

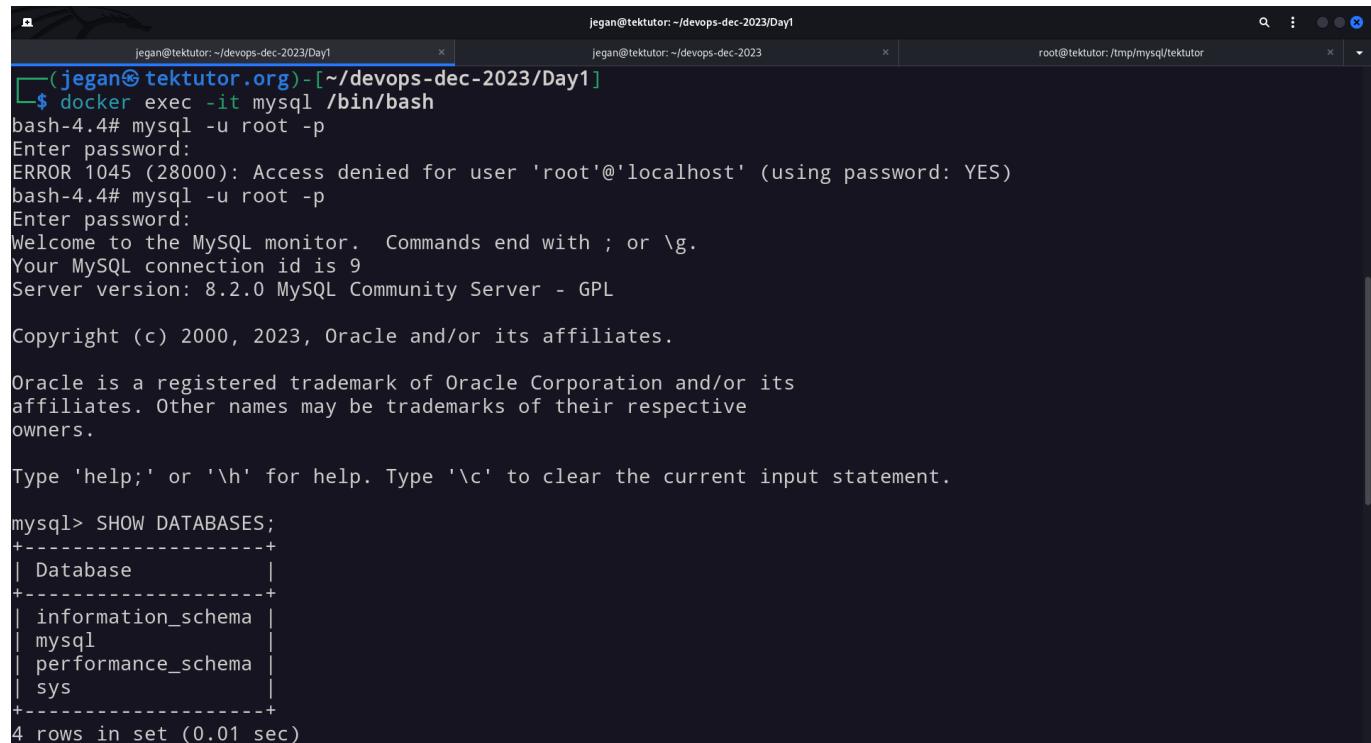
The right tab shows the MySQL container running with port mappings:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAME
c03e715f2af8c0e04348235f097468c0817209c5228783a28f6a89532249268	mysql:latest	"docker-entrypoint.s..."	5 seconds ago	Up 4 seconds	3306/tcp, 33060/tcp	mysql

Let's connect to mysql server using the mysql client that comes with the mysql container

```
docker exec -it mysql /bin/bash  
mysql -u root -p  
SHOW DATABASES;  
CREATE DATABASE tektutor;  
USE tektutor;  
SHOW TABLES;  
CREATE TABLE training (id INT NOT NULL, name VARCHAR(200), duration  
VARCHAR(200), PRIMARY KEY(id) );  
INSERT INTO training VALUES ( 1, "Kubernetes", "5 Days" );  
INSERT INTO training VALUES ( 2, "Microservices using Spring Boot  
Framework", "5 Days" );  
exit  
exit
```

Expected output



The screenshot shows a terminal window with three tabs. The active tab is titled '(jegan@tektutor.org)-[~/devops-dec-2023/Day1]'. It contains the following MySQL session:

```
jegan@tektutor: ~/devops-dec-2023/Day1
$ docker exec -it mysql /bin/bash
bash-4.4# mysql -u root -p
Enter password:
ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| sys            |
+-----+
4 rows in set (0.01 sec)
```

The screenshot shows three terminal windows side-by-side. The left window (jegan@tektutor: ~/devops-dec-2023/Day1) displays MySQL commands to create a database and tables, and insert data. The middle window (jegan@tektutor: ~/devops-dec-2023) shows the database being used and the creation of a table named 'training'. The right window (root@tektutor: /tmp/mysql/tektutor) is a MySQL root shell where the 'training' table is selected and its contents are displayed.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.01 sec)

mysql> CREATE DATABASE tektutor;
Query OK, 1 row affected (0.01 sec)

mysql> SHOW TABLES;
ERROR 1046 (3D000): No database selected
mysql> USE tektutor;
Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql> CREATE TABLE training (id INT NOT NULL, name VARCHAR(200), duration VARCHAR(200), PRIMARY KEY(id) );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO training VALUES ( 1, "Kubernetes", "5 Days" );
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO training VALUES ( 2, "Microservices using Spring Boot Framework", "5 Days" );

Database changed
mysql> SHOW TABLES;
Empty set (0.00 sec)

mysql> CREATE TABLE training (id INT NOT NULL, name VARCHAR(200), duration VARCHAR(200), PRIMARY KEY(id) );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO training VALUES ( 1, "Kubernetes", "5 Days" );
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO training VALUES ( 2, "Microservices using Spring Boot Framework", "5 Days" );
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM training;
+----+-----+-----+
| id | name           | duration |
+----+-----+-----+
| 1  | Kubernetes     | 5 Days   |
| 2  | Microservices using Spring Boot Framework | 5 Days   |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye
bash-4.4# exit
exit
```

Let's delete the mysql container forcibly

```
docker rm -f mysql
docker ps -a
```

Expected output

```
jegan@tektutor: ~/devops-dec-2023/Day1
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO training VALUES ( 2, "Microservices using Spring Boot Framework", "5 Days" );
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM training;
+----+-----+-----+
| id | name           | duration |
+----+-----+-----+
| 1  | Kubernetes     | 5 Days   |
| 2  | Microservices using Spring Boot Framework | 5 Days   |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye
bash-4.4# exit
exit

[jegan@tektutor.org] - [~/devops-dec-2023/Day1]
$ docker rm -f mysql
mysql

[jegan@tektutor.org] - [~/devops-dec-2023/Day1]
$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
[jegan@tektutor.org] - [~/devops-dec-2023/Day1]
$
```

If you check the local machine /tmp/mysql folder, it would look as shown below

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023
root@tektutor: /tmp/mysql/tektutor

[jegan@tektutor.org] - [/tmp/mysql]
$ ls
auto.cnf      ca.pem          ib_buffer_pool    mysql          public_key.pem  undo_001
binlog.000001  client-cert.pem ibdata1          mysql.ibd    server-cert.pem undo_002
binlog.000002  client-key.pem  ibtmp1          mysql.sock   server-key.pem
binlog.index   '#ib_16384_0 dblwr'  '#innodb_redo' performance_schema sys
ca-key.pem    '#ib_16384_1 dblwr'  '#innodb_temp'  private_key.pem tektutor

[jegan@tektutor.org] - [/tmp/mysql]
$ cd tektutor
cd: permission denied: tektutor

[jegan@tektutor.org] - [/tmp/mysql]
$ sudo su -
[sudo] password for jegan:
[jegan@tektutor.org] - [~]
# cd /tmp/mysql/tektutor

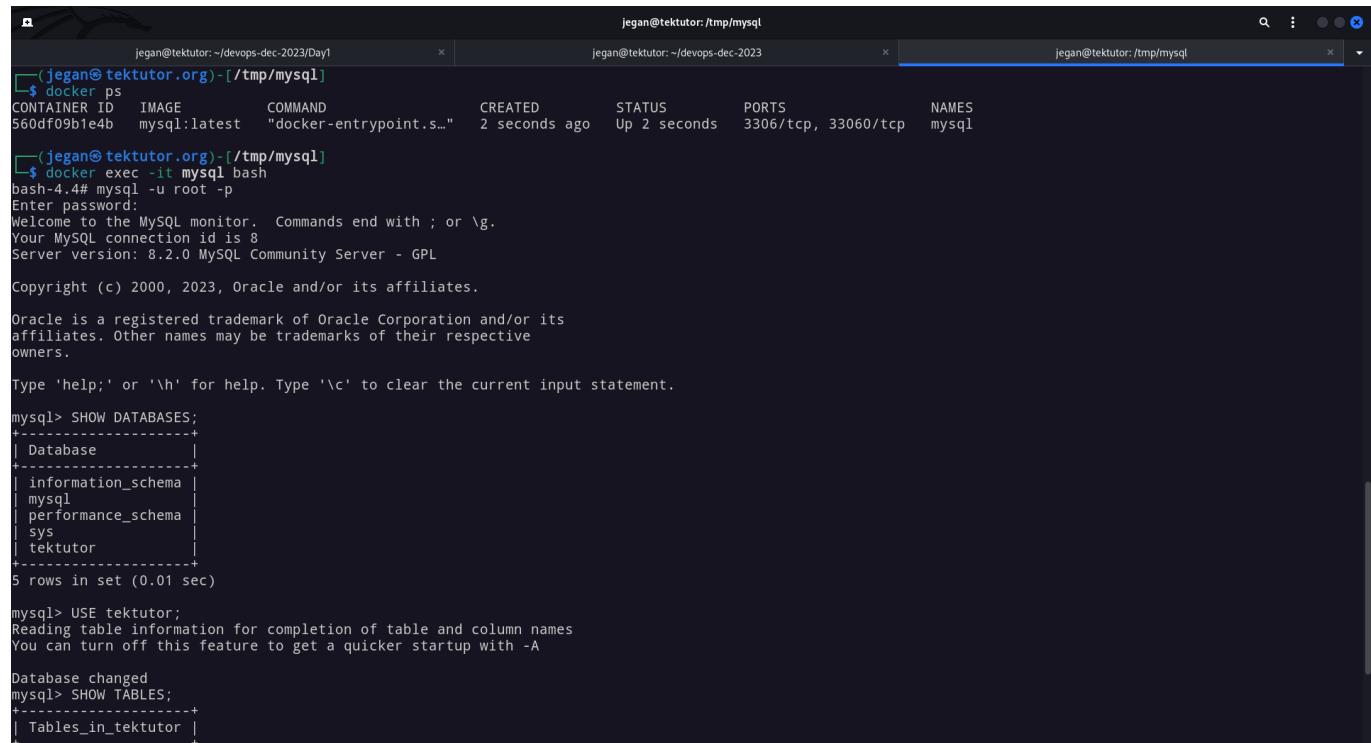
[jegan@tektutor.org] - [/tmp/mysql/tektutor]
# ls
[jegan@tektutor.org] - [/tmp/mysql/tektutor]
# ls -lha
total 8.0K
drwxr-x-- 2 999 systemd-journal 4.0K Dec  7 16:48 .
drwxr-xr-x 8 999 jegan       4.0K Dec  7 16:48 ..

[jegan@tektutor.org] - [/tmp/mysql/tektutor]
```

Let's create a new mysql container mounting the same /tmp/mysql local folder as shown below

```
docker ps -a
docker run -d --name mysql --hostname mysql -v /tmp/mysql:/var/lib/mysql -e
MYSQL_ROOT_PASSWORD=root@123 mysql:latest
docker ps
docker exec -it mysql bash
SHOW DATABASES;
USE tektutor;
SHOW TABLES;
SELECT * FROM training;
```

Expected output



The screenshot shows a terminal window with three tabs. The active tab is titled '(jegan@tektutor.org)-[/tmp/mysql]'. The terminal session starts with:

```
jegan@tektutor: ~/devops-dec-2023/Day1
[jegan@tektutor.org]-[/tmp/mysql]
$ docker ps
CONTAINER ID   IMAGE      COMMAND           CREATED          STATUS    PORTS     NAMES
560df09b1e4b   mysql:latest "docker-entrypoint.s..." 2 seconds ago   Up 2 seconds   3306/tcp, 33060/tcp   mysql

(jegan@tektutor.org)-[/tmp/mysql]
$ docker exec -it mysql bash
bash-4.4# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.2.0 MySQL Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

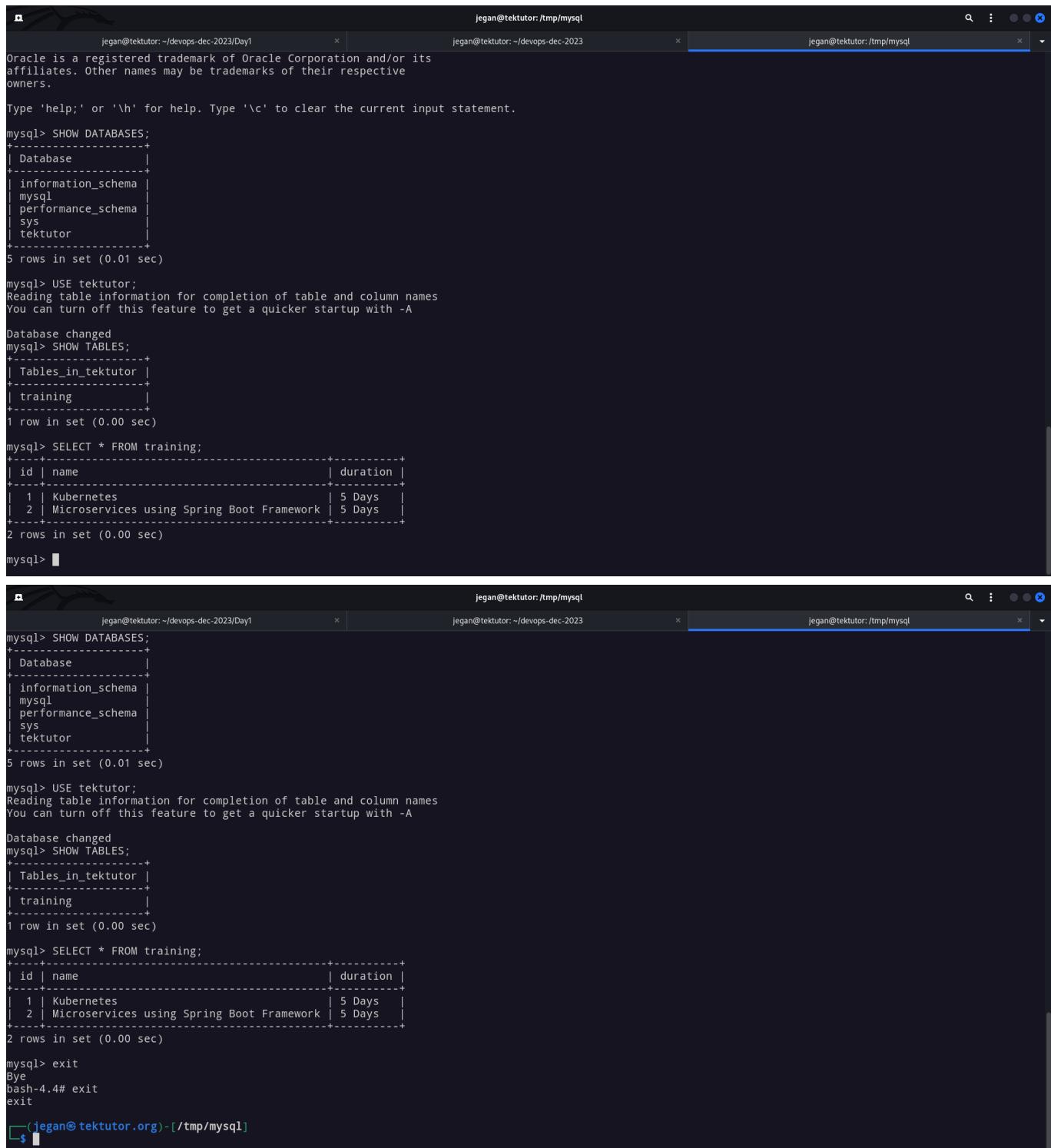
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| tektutor |
+-----+
5 rows in set (0.01 sec)

mysql> USE tektutor;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_tektutor |
+-----+
```



The screenshot shows three terminal windows side-by-side, all titled 'jegan@tektutor: /tmp/mysql'. The first window contains MySQL commands to show databases, use the 'tektutor' database, and select from a 'training' table. The second window is identical to the first. The third window shows the MySQL prompt again, followed by an 'exit' command and the bash prompt 'bash-4.4# exit'.

```
jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023
jegan@tektutor: /tmp/mysql

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| tektutor |
+-----+
5 rows in set (0.01 sec)

mysql> USE tektutor;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_tektutor |
+-----+
| training |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM training;
+----+-----+-----+
| id | name | duration |
+----+-----+-----+
| 1 | Kubernetes | 5 Days |
| 2 | Microservices using Spring Boot Framework | 5 Days |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> 

jegan@tektutor: ~/devops-dec-2023/Day1
jegan@tektutor: ~/devops-dec-2023
jegan@tektutor: /tmp/mysql

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| tektutor |
+-----+
5 rows in set (0.01 sec)

mysql> USE tektutor;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> SHOW TABLES;
+-----+
| Tables_in_tektutor |
+-----+
| training |
+-----+
1 row in set (0.00 sec)

mysql> SELECT * FROM training;
+----+-----+-----+
| id | name | duration |
+----+-----+-----+
| 1 | Kubernetes | 5 Days |
| 2 | Microservices using Spring Boot Framework | 5 Days |
+----+-----+-----+
2 rows in set (0.00 sec)

mysql> exit
Bye
bash-4.4# exit
exit
$ (jegan@tektutor.org) - [/tmp/mysql]
```

Lab - Creating nginx web server as a container

```
docker ps -a
docker run -d --name web1 --hostname web1 nginx:latest
docker ps
```

Expected output

```
jegan@tektutor.org-[~/devops-dec-2023/Day1]
$ docker run -d --name web1 --hostname web1 nginx:latest
87a51566aec0df256720401cebe7d9e35c83e756e729fade70201d1cc3c1bdb

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID   IMAGE      COMMAND       CREATED      STATUS      PORTS
MES
87a51566aec0   nginx:latest "/docker-entrypoint..."  2 seconds ago Up 1 second  80/tcp
b1
560df09b1e4b   mysql:latest "docker-entrypoint.s..."  20 minutes ago Up 20 minutes  3306/tcp, 33060/tcp
sql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker inspect -f {{.NetworkSettings.IPAddress}} web1
172.17.0.3

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ curl http://172.17.0.3:80
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
```

Lab - Creating nginx web server with port forward to expose the container service to outside world

```
docker rm -f web1
docker run -d --name web1 --hostname web1 -p 8001:80 nginx:latest
docker ps
docker inspect web1 | grep IPA
ifconfig ens192
curl http://localhost:8001
```

Expected output

The screenshot shows two terminal sessions. The top session is a standard terminal window titled '(jegan@tektutor.org)-[~/devops-dec-2023/Day1]'. It displays the command \$ docker rm -f web1 followed by the output 'web1'. Below this, it shows the command \$ docker run -d --name web1 --hostname web1 -p 8001:80 nginx:latest, with its output: 27fd7ec373223d2934b56457eb12c2e747b82a571eafbb7169c7226554926325. Then it runs \$ docker ps to list containers, showing two: web1 (nginx:latest) created 2 seconds ago and Up 1 second, and mysql (mysql:latest) created 45 minutes ago and Up 45 minutes. Finally, it runs \$ curl http://localhost:8001, displaying the nginx welcome page.

The bottom session is a MySQL terminal window titled 'mysql' and '(jegan@tektutor.org)-[~/devops-dec-2023/Day1]'. It shows the command \$ curl http://localhost:8001, followed by the full HTML response from the nginx server, which includes the welcome message, documentation links, and a note about commercial support.

```

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker rm -f web1
web1

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -d --name web1 --hostname web1 -p 8001:80 nginx:latest
27fd7ec373223d2934b56457eb12c2e747b82a571eafbb7169c7226554926325

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID        IMAGE       COMMAND             CREATED          STATUS           PORTS
27fd7ec37322        nginx:latest "/docker-entrypoint..."   2 seconds ago   Up  1 second    0.0.0.0:8001->80/tcp, ::1:8001->80/tcp
560df09b1e4b        mysql:latest "docker-entrypoint.s..."  45 minutes ago  Up  45 minutes   3306/tcp, 33060/tcp

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ curl http://localhost:8001
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>
<p><em>Thank you for using nginx.</em></p>
</body>
</html>

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ curl http://localhost:8001

```

Lab - Creating a container in the interactive/foreground mode

```

docker ps
docker run -it --name c1 --hostname c1 ubuntu:16.04 /bin/bash
docker ps

```

Expected output

```
jegan@tektutor.org:[~/devops-dec-2023/Day1]
$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
27fd7ec37322 nginx:latest "/docker-entrypoint..." About a minute ago Up About a minute 0.0.0.0:8001->80
/tcp, :::8001->80/tcp web1
560df09b1e4b mysql:latest "docker-entrypoint.s..." 46 minutes ago Up 46 minutes 3306/tcp, 33060/
tcp mysql

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$ docker run -it --name c1 --hostname c1 ubuntu:16.04 /bin/bash
root@c1:/# hostname
c1
root@c1:/# hostname -i
172.17.0.4
root@c1:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@c1:/# exit
exit

(jegan@tektutor.org)-[~/devops-dec-2023/Day1]
$
```