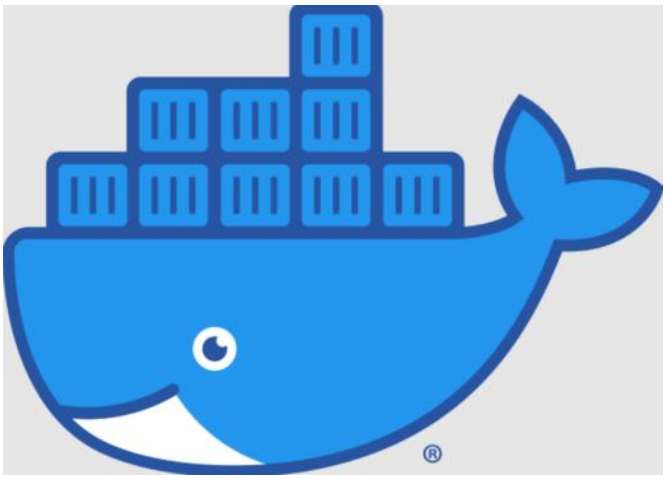
# Day 16: Docker for DevOps Engineers

**This is #90DaysofDevops challenge under the guidance of Shubham Londhe sir.**

Day 16 TASK

check this for task:

https://github.com/LondheShubham153/90DaysOfDevOps/blob/master/2023/day16/tasks.md



## Docker

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

## Tasks

Install Docker on your server first.

Follow below commands for the installation:

https://swapnasagarpradhan.medium.com/how-to-install-docker-on-amazon-linux-2-8e5161ac5464

Now we will perform some Docker Commands:

1. **Use the `docker run` command to start a new container and interact with it through the command line. [Hint: docker run hello-world]**

```
[root@ip-172-31-40-110 ec2-user]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
debian       latest    54e726b437fb   10 days ago   124MB
[root@ip-172-31-40-110 ec2-user]# docker run -d debian
fb42d8d6f8407ce1e41b2920f981f6644b552416e09d589e70bab8c5551abe79
[root@ip-172-31-40-110 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-40-110 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED         STATUS                     PORTS     NAMES
fb42d8d6f840   debian    "bash"    9 seconds ago   Exited (0) 8 seconds ago             confident_clarke
[root@ip-172-31-40-110 ec2-user]#
```

Here we first pulled an image from DockerHub by command.

```
#docker pull debian
```

Then we used docker run command to run a container in detached mode using the pulled image.

- To create a container with container name:
- command — docker run -d — name <container-name> <image-name>

```
[root@ip-172-31-40-110 ec2-user]# docker run -d --name cont1 debian
a24a97e017a7c1d494e3c400f6f39df19ec16f5e324dddc8905826e08eb6afb6
[root@ip-172-31-40-110 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-40-110 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED         STATUS                     PORTS     NAMES
a24a97e017a7   debian    "bash"    6 seconds ago   Exited (0) 5 seconds ago             cont1
fb42d8d6f840   debian    "bash"    3 minutes ago   Exited (0) 3 minutes ago             confident_clarke
[root@ip-172-31-40-110 ec2-user]#
```

- To create a container with port number:
- command — docker run -d -p 8080:80 <image-name>

```
[root@ip-172-31-40-110 ec2-user]# docker run -d -p 8080:80 debian
c2323e0039f6783651a1c7c4068fabc8ad730e244945c5ac47702c0410dae077
[root@ip-172-31-40-110 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED              STATUS                     PORTS     NAMES
c2323e0039f6   debian    "bash"    5 seconds ago        Exited (0) 3 seconds ago             exciting_hellman
a24a97e017a7   debian    "bash"    About a minute ago   Exited (0) 59 seconds ago            cont1
fb42d8d6f840   debian    "bash"    4 minutes ago        Exited (0) 4 minutes ago             confident_clarke
```

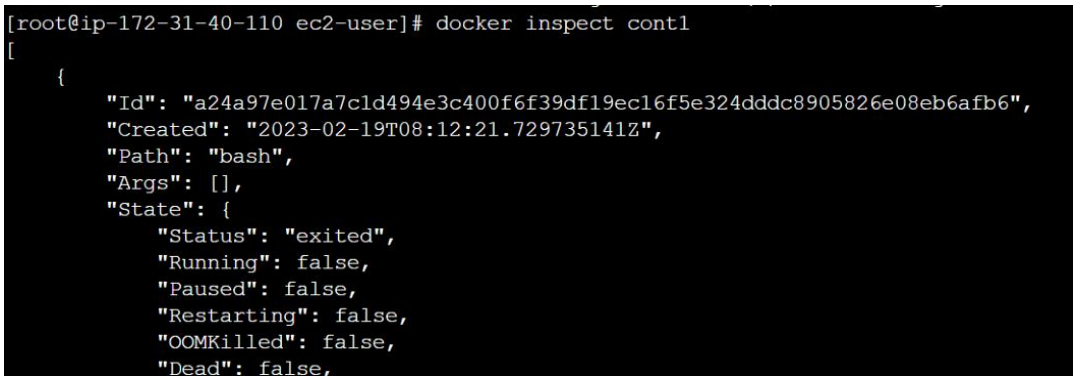
- Create a container with name and port number:

```
[root@ip-172-31-44-34 ec2-user]# docker run -d --name new1 -p 8081:80 nginx:alpine
f8494b24f897e0c613cf76ecba14e65f71db50c35c2920e5cedf8f8056db5e90
[root@ip-172-31-44-34 ec2-user]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED              STATUS              PORTS                                     NAMES
f8494b24f897   nginx:alpine   "/docker-entrypoint.…"   4 seconds ago        Up 3 seconds        0.0.0.0:8081->80/tcp, :::8081->80/tcp     new1
747670d495b9   nginx:alpine   "/docker-entrypoint.…"   About a minute ago   Up About a minute   80/tcp                                    cont101
```

- **2. Use the `docker inspect` command to view detailed information about a container or image.**

`docker inspect` is a Docker command used to obtain detailed information about a Docker container or image.

```
docker inspect <container name>
```




**3. Use the `docker port` command to list the port mappings for a container.**

`docker port` is a Docker command that displays the public-facing port(s) of a running container.

```
docker port <container name>
```


- **Use the `docker stats` command to view resource usage statistics for one or more containers.**

`docker stats` is a Docker command that displays real-time usage statistics for running containers.

```
docker stats [OPTIONS] [CONTAINER...]
```

Options:

- — all , -a: Show all containers (default shows just running)
- — format: Pretty print images using a Go template
- — no-stream: Disable streaming stats and only pull the first result
- — no-trunc: Do not truncate output

```
CONTAINER ID   NAME      CPU %     MEM USAGE / LIMIT     MEM %     NET I/O       BLOCK I/O     PIDS
f8494b24f897   new1      0.00%     1.973MiB / 964.8MiB   0.20%     850B / 0B     0B / 17.4kB   2
747670d495b9   cont101   0.00%     2.02MiB / 964.8MiB    0.21%     1.07kB / 0B   0B / 20.5kB   2
55b0ffaf5179   cont1     0.00%     0B / 0B               0.00%     0B / 0B       0B / 0B       0
```

- **Use the `docker top` command to view the processes running inside a container.**

`docker top` is a Docker command that shows the processes running inside a container.

The output of the `docker top` command includes the process ID, user, CPU usage, memory usage, and command that is being executed by each process.

```
docker top <container name>
```

```
[root@ip-172-31-44-34 ec2-user]# docker top new1
UID                 PID                 PPID                C                   STIME               TTY                 TIME                CMD
root                3728                3705                0                   14:56               ?                   00:00:00            nginx
: master process nginx -g daemon off;
101                 3778                3728                0                   14:56               ?                   00:00:00            nginx
: worker process
[root@ip-172-31-44-34 ec2-user]#
```

- **Use the `docker save` command to save an image to a tar archive.**

`docker save` is a Docker command that saves one or more Docker images to a tar archive.

```
docker save [OPTIONS] IMAGE [IMAGE…]
```

```
[root@ip-172-31-44-34 ec2-user]# docker save  nginx:alpine --output nginx-backup.tar
[root@ip-172-31-44-34 ec2-user]# ls
nginx-backup.tar
[root@ip-172-31-44-34 ec2-user]#
```

- **Use the `docker load` command to load an image from a tar archive.**

`docker load` is a Docker command that loads one or more Docker images from a tar archive.

```
docker load [OPTIONS] < myimages.tar
```

```
[root@ip-172-31-44-34 ec2-user]# docker load --input nginx-backup.tar
Loaded image: nginx:alpine
[root@ip-172-31-44-34 ec2-user]#
```

**Please, feel free to drop any questions in the comments below. I would be happy to answer them.**

**If this post was helpful, please do follow and click the clap**

**_Thank you for reading**

**_Rajani**