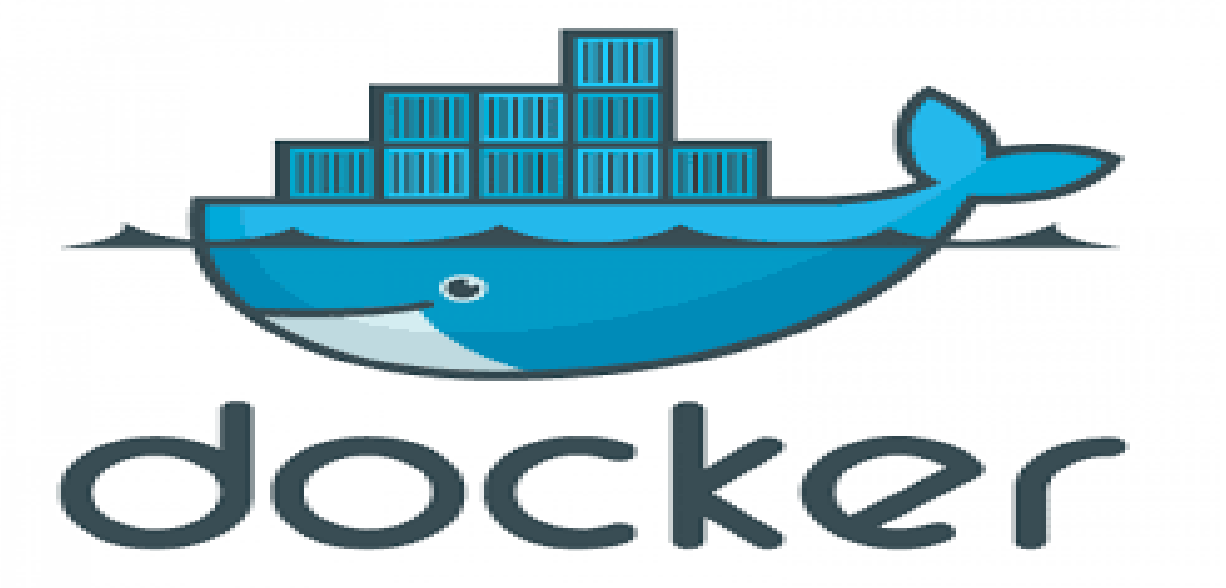
Day 16 Task: Docker for DevOps Engineers.



Docker

Docker is a platform and technology for building, shipping, and running distributed applications, whether on laptops, datacentre VMs, or the cloud. It allows developers to package their applications and dependencies into a portable container, which can then be run on any system with the Docker runtime installed. This makes it easy to deploy and run applications consistently across different environments, and helps to ensure that the application will run correctly regardless of the underlying infrastructure. Docker also provides tools for managing and orchestrating multiple containers, making it a popular choice for building and deploying microservices-based applications.

Docker also provides a tool called Docker Compose, which allows developers to define and run multi-container applications. It uses a YAML file to configure the application's services and networks, and it can spin up and tear down entire environments with a single command.

Follow the below steps to setup docker

Steps 1 : Create a Ec2 instance in AWS.

Steps 2 : Install Docker in instance.

Commands:

- sudo yum update -y

- sudo amazon-linux-extras install docker
- sudo service docker start
- sudo systemctl enable docker
- sudo usermod -a -G docker ec2-user
- Log out and log back in again to pick up the new docker group permissions. You can accomplish this by closing your current SSH terminal window and reconnecting to your instance in a new one. Your new SSH session will have the appropriate docker group permissions.
- docker info

Steps 3 : Create a Docker image

Commands :

- docker pull <image_name>
- docker run -d --name <container-name> -p <port mapping> <image_name:tag>

Tasks

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

```
ubuntu@ip-172-31-90-202:~$ docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:aa0cc8055b82dc2509bed2e19b275c8f463506616377219d9642221ab53cf9fe
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

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

```
ubuntu@ip-172-31-90-202:~$ docker run -d --name jekins-server-dpcker -p 8080:808
0 jenkins/jenkins:latest
7f282ca71d2b839070aa53e90c3aa6c03a6388ed2e8c58ad74df2381591440db
ubuntu@ip-172-31-90-202:~$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
  STATUS          PORTS                                                 NAMES
7f282ca71d2b   jenkins/jenkins:latest   "/usr/bin/tini -- /u…"   11 seconds ago
  Up 10 seconds   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp   jekins-
server-dpcker
```

```
ubuntu@ip-172-31-90-202:~$ docker run -d --name jekins-server-dpcker -p 8080:808
0 jenkins/jenkins:latest
7f282ca71d2b839070aa53e90c3aa6c03a6388ed2e8c58ad74df2381591440db
ubuntu@ip-172-31-90-202:~$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED
  STATUS          PORTS                                                 NAMES
7f282ca71d2b   jenkins/jenkins:latest   "/usr/bin/tini -- /u…"   11 seconds ago
  Up 10 seconds   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp   jekins-
server-dpcker
ubuntu@ip-172-31-90-202:~$ docker inspect 7f282ca71d2b
[
    {
        "Id": "7f282ca71d2b839070aa53e90c3aa6c03a6388ed2e8c58ad74df2381591440db"
,
        "Created": "2023-01-21T08:56:25.034148774Z",
        "Path": "/usr/bin/tini",
        "Args": [
```

```
7befa292da30be54188",
                    "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
                }
            }
        }
    }
]
ubuntu@ip-172-31-90-202:~$
```

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

```
ubuntu@ip-172-31-90-202:~$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED         STATUS         PORTS                                                 NAMES
7f282ca71d2b   jenkins/jenkins:latest   "/usr/bin/tini -- /u…"   7 minutes ago   Up 7 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp   jekins-server-dpcker
ubuntu@ip-172-31-90-202:~$ docker stats 7f282ca71d2d
Error response from daemon: No such container: 7f282ca71d2d
ubuntu@ip-172-31-90-202:~$ docker stats 7f282ca71d2b
CONTAINER ID   NAME                  CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O       PIDS
7f282ca71d2b   jekins-server-dpcker   4.07%     290.5MiB / 966.2MiB   30.07%    5.56MB / 29kB   168MB / 128MB   38
CONTAINER ID   NAME                  CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O       PIDS
7f282ca71d2b   jekins-server-dpcker   0.39%     288.5MiB / 966.2MiB   29.86%    5.56MB / 29kB   168MB / 128MB   38
CONTAINER ID   NAME                  CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O       PIDS
```

```
ubuntu@ip-172-31-90-202:~$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS
7f282ca71d2b   jenkins/jenkins:latest   "/usr/bin/tini -- /u…"   4 minutes ago    Up 4 minutes
ubuntu@ip-172-31-90-202:~$ docker port 7f282ca71d2b
8080/tcp -> 0.0.0.0:8080
8080/tcp -> :::8080
```

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

```
CONTAINER ID   NAME                   CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O       PIDS
7f282ca71d2b   jekins-server-dpcker   0.29%     288.5MiB / 966.2MiB   29.86%    5.56MB / 29kB   168MB / 128MB   38
```

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

```
ubuntu@ip-172-31-90-202:~$ docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED         STATUS         PORTS                                                  NA
7f282ca71d2b   jenkins/jenkins:latest   "/usr/bin/tini -- /u…"   11 minutes ago   Up 11 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 50000/tcp   je
s-server-dpcker
ubuntu@ip-172-31-90-202:~$ docker top 7f282ca71d2b
UID                 PID                 PPID                C                   STIME               TTY                 TIME                CMD
ubuntu              1577                1556                0                   08:56               ?                   00:00:00            /usr/bin/tini --
usr/local/bin/jenkins.sh
ubuntu              1608                1577                5                   08:56               ?                   00:00:41            java -Duser.home
var/jenkins_home -Djenkins.model.Jenkins.slaveAgentPort=50000 -Dhudson.lifecycle=hudson.lifecycle.ExitLifecycle -jar /usr/share/jenkins/jenkins.war
ubuntu@ip-172-31-90-202:~$
```

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

```
ubuntu@ip-172-31-90-202:~$ docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED             STATUS             PORTS
7f282ca71d2b   jenkins/jenkins:latest   "/usr/bin/tini -- /u…"   About an hour ago   Up About an hour   0.0.0.0:8080->8080/tcp, :::8080
-dpcker
ubuntu@ip-172-31-90-202:~$ docker save jenkins/jenkins > jenkins_jenkin.tar
ubuntu@ip-172-31-90-202:~$
```

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

```
ubuntu@ip-172-31-90-202:~$ docker save jenkins/jenkins > jenkins_jenkin.tar
ubuntu@ip-172-31-90-202:~$ docker load -i jenkins_jenkin.tar
Loaded image: jenkins/jenkins:latest
ubuntu@ip-172-31-90-202:~$
```