40

DAYS

Container

- A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

- Containers and virtual machines have similar resource isolation and allocation benefits, but function differently because containers virtualize the operating system instead of hardware. Containers are more portable and efficient.

- Containers take up less space than VMs , can handle more applications and require fewer VMs and Operating systems.

Containers and Virtual Machines Together

- Containers and VMs used together provide a great deal of flexibility in deploying and managing app .

- Virtual machines and containers differ in several ways, but the primary difference is that containers provide a way to virtualize an OS so that multiple workloads can run on a single OS instance. With VMs, the hardware is being virtualized to run multiple OS instances.

- VMs are a better choice for running apps that require all of the operating system’s resources and functionality when you need to run multiple applications on servers, or have a wide variety of operating systems to manage.

- Containers are a better choice when your biggest priority is maximizing the number of applications running on a minimal number of servers.

Benefits of Containers

- Reduced IT management resources
- Reduced size of snapshots
- Quicker spinning up apps
- Reduced & simplified security updates
- Less code to transfer, migrate, upload workloads

Types of Containers

Linux Containers (LXC) — The original Linux container technology is Linux Containers

Docker — Is technology that allows development teams to build, manage, and secure apps anywhere.

Docker is popular because of the possibilities it opens for software delivery and deployment. Many common problems and inefficiencies are resolved with containers.
reasons for Docker's popularity are:

1.Ease of use
2. Faster scaling of systems : Containers allow much more work to be done by far less computing hardware.
3. Better software delivery : Software delivery using containers can also be more efficient. Containers are portable.
4. Flexibility :Operating containerized applications is more flexible and resilient than that of non-containerized applications.
5. Software-defined networking .
6.The rise of microservices architecture.
7. Isolation: Dependencies or settings within a container will not affect any installations or configurations on your computer, or on any other containers that may be running.
8. Security

when we need to use Docker !

App isolation: If you want to run multiple applications on one server, keeping the components of each application in separate containers will prevent problems with dependency management.

Developer teams: If you have developers working with different setups, Docker provides a convenient way to have local development environments that closely match the production environment, without needing to ssh into a remote box.

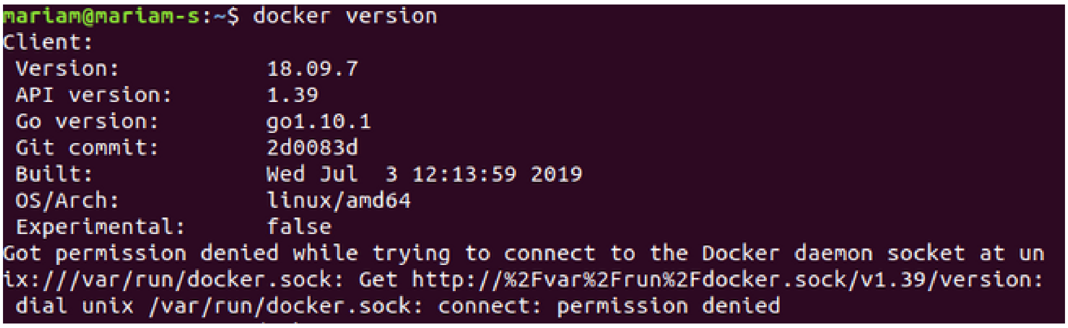
★ Docker image is a file, comprised of multiple layers, used to execute code in a Docker container.

Install Docker On

$ sudo ap-get install docker-ce

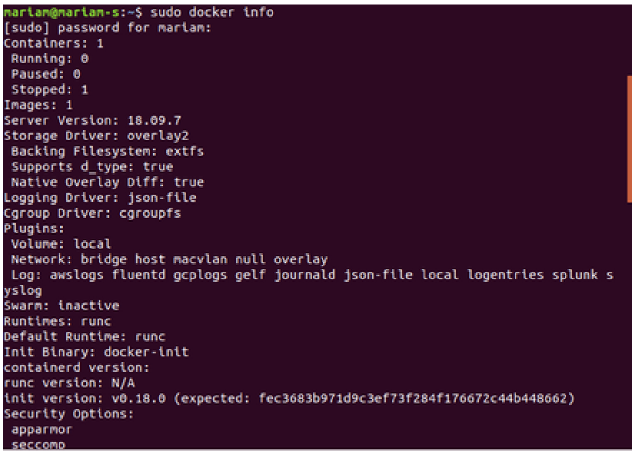
After downlod docker :

To check the version



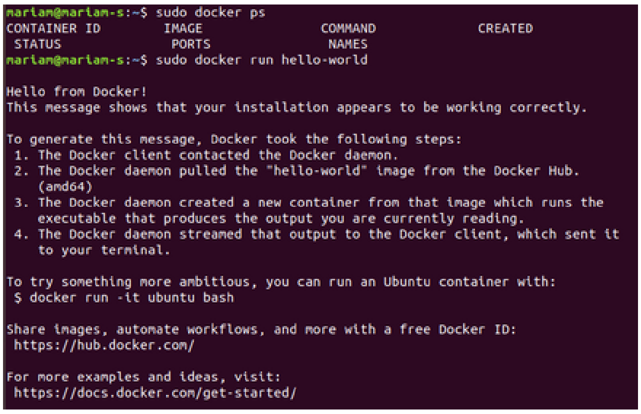
```
mariam@mariam-s:~$ docker version
Client:
 Version:           18.09.7
 API version:       1.39
 Go version:        go1.10.1
 Git commit:        2d0083d
 Built:             Wed Jul  3 12:13:59 2019
 OS/Arch:           linux/amd64
 Experimental:      false
Got permission denied while trying to connect to the Docker daemon socket at un
ix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.39/version:
 dial unix /var/run/docker.sock: connect: permission denied
```

To check the information about docker:

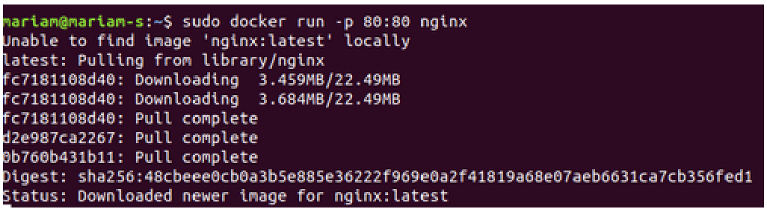


```
mariam@mariam-s:~$ sudo docker info
[sudo] password for mariam:
Containers: 1
 Running: 0
 Paused: 0
 Stopped: 1
Images: 1
Server Version: 18.09.7
Storage Driver: overlay2
 Backing Filesystem: extfs
 Supports d_type: true
 Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
 Volume: local
 Network: bridge host macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk s
yslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version:
runc version: N/A
init version: v0.18.0 (expected: fec3683b971d9c3ef73f284f176672c44b448662)
Security Options:
 apparmor
 seccomp
```

To run the first container :



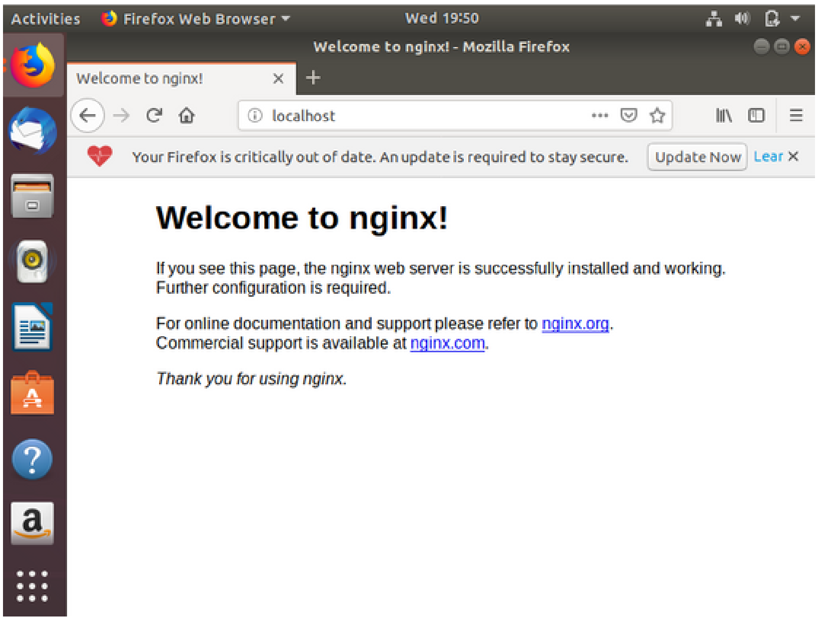
```
mariam@mariam-s:~$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
 STATUS              PORTS               NAMES
mariam@mariam-s:~$ sudo docker run hello-world

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/
```



```
mariam@mariam-s:~$ sudo docker run -p 80:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
fc7181108d40: Downloading  3.459MB/22.49MB
fc7181108d40: Downloading  3.684MB/22.49MB
fc7181108d40: Pull complete
d2e987ca2267: Pull complete
0b760b431b11: Pull complete
Digest: sha256:48cbeee0cb0a3b5e885e36222f969e0a2f41819a68e07aeb6631ca7cb356fed1
Status: Downloaded newer image for nginx:latest
```

Install Docker On

After running :



Your Firefox is critically out of date. An update is required to stay secure.

Welcome to nginx!

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

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

Thank you for using nginx.