

ORACLE
Certified Professional

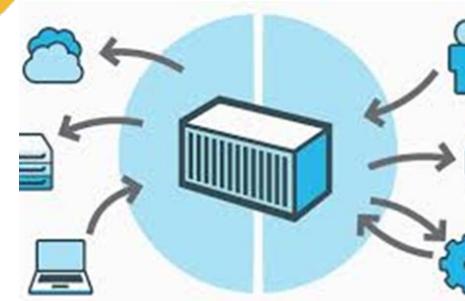
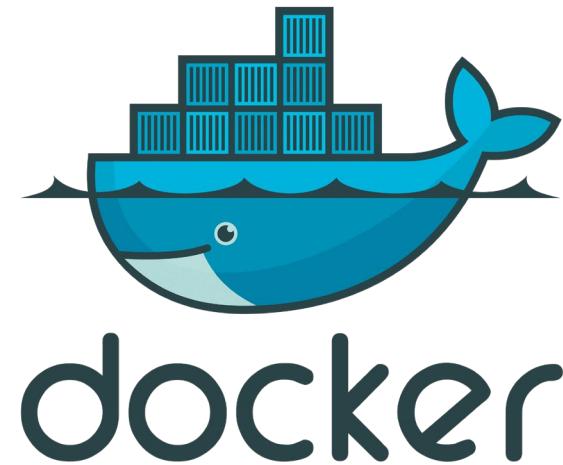


Microsoft Certified
Professional



sohailimran2@yahoo.com

Ref: Web



Dr. Sohail IMRAN



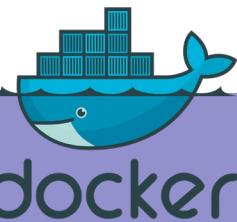
Docker Installation

Docker Account

The screenshot shows the Docker Home page in a browser. At the top, there's a banner for the Docker MCP Catalog and Toolkit. Below it, a large blue header bar displays a welcome message and a user profile icon with the letter 'S' and the name 'sohailimran'. On the left, there's a sidebar for 'Docker products' featuring links to 'docker:desktop', 'scout', and 'docker:hub'. The main content area has sections for 'What's new', 'My profile', 'Account settings' (which is currently selected), 'Billing', and 'Sign out'. Below these is a link to 'Go to Build Cloud'.

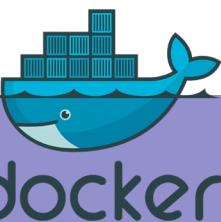
AVOID connecting with a GOOGLE Account.

Create a Docker Account using Username and Password



update

```
sohailimran@SI:~$ sudo apt update
Hit:1 http://pk.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://pk.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://pk.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [615 kB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2,222 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [342 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main amd64 DEP-11 Metadata [43.1 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.6 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted i386 Packages [40.1 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [3,145 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [558 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 DEP-11 Metadata [208 B]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [624 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [657 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [972 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [209 kB]
Get:18 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 DEP-11 Metadata [125 kB]
Get:19 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [21.7 kB]
Get:20 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [39.6 kB]
Get:21 http://security.ubuntu.com/ubuntu jammy-security/multiverse i386 Packages [1,720 B]
Get:22 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [8,716 B]
Get:23 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 DEP-11 Metadata [208 B]
Get:24 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [368 B]
Fetched 9,144 kB in 8s (1,088 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
144 packages can be upgraded. Run 'apt list --upgradable' to see them.
sohailimran@SI:~$
```

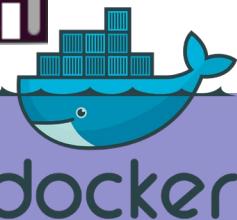


Ensure prerequisites

Run these commands one by one:

```
sudo apt install apt-transport-https  
sudo apt install ca-certificates  
sudo apt install curl  
sudo apt install software-properties-common
```

```
sohailimran@SI:~$ sudo apt install apt-transport-https ca-certificates curl software-properties-common  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
software-properties-common is already the newest version (0.99.22.9).  
software-properties-common set to manually installed.  
ca-certificates is already the newest version (20240203~22.04.1).  
ca-certificates set to manually installed.  
The following additional packages will be installed:  
  libcurl4  
The following NEW packages will be installed:  
  apt-transport-https curl  
The following packages will be upgraded:  
  libcurl4  
1 upgraded, 2 newly installed, 0 to remove and 143 not upgraded.  
Need to get 485 kB of archives.  
After this operation, 625 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://pk.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 apt-transport-https all 2.4.12 [1,510 B]  
7% [Waiting for headers] 220 B/s 36min 37s
```



Add Docker's official GPG (GNU Privacy Guard) key

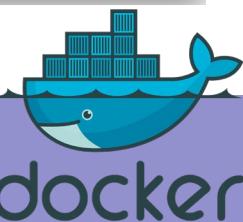
Run the following 3 commands:

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] \
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

sudo apt update

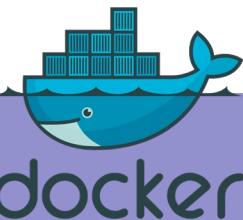
```
sohailimran@SI:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
sohailimran@SI:~$ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] http://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sohailimran@SI:~$ sudo apt update
Hit:1 http://pk.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://pk.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://pk.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 https://download.docker.com/linux/ubuntu jammy InRelease [48.8 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [47.6 kB]
Fetched 225 kB in 2s (125 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
143 packages can be upgraded. Run 'apt list --upgradable' to see them.
sohailimran@SI:~$
```



Ensure installation from the Docker repo

apt-cache policy docker-ce

```
sohailimran@SI:~$ apt-cache policy docker-ce
docker-ce:
  Installed: (none)
  Candidate: 5:28.0.4-1~ubuntu.22.04~jammy
  Version table:
    5:28.0.4-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:28.0.3-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:28.0.2-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:28.0.1-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:28.0.0-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:27.5.1-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:27.5.0-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:27.4.1-1~ubuntu.22.04~jammy 500
      500 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages
    5:27.4.0-1~ubuntu.22.04~jammy 500
```



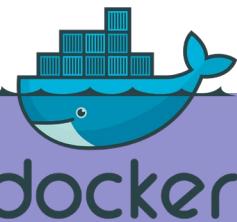
install Docker

`sudo apt install docker-ce`

```
sohailimran@SI:~$ sudo apt install docker-ce
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 containerd.io docker-buildx-plugin docker-ce-cli docker-ce-rootless-extras
 docker-compose-plugin git git-man liberror-perl libslirp0 pigz slirp4netns
Suggested packages:
 cgroupfs-mount | cgroup-lite git-daemon-run | git-daemon-sysvinit git-doc
 git-email git-gui gitk gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
 containerd.io docker-buildx-plugin docker-ce docker-ce-cli
 docker-ce-rootless-extras docker-compose-plugin git git-man liberror-perl
 libslirp0 pigz slirp4netns
0 upgraded, 12 newly installed, 0 to remove and 143 not upgraded.
Need to get 125 MB of archives.
After this operation, 463 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

In case of Failure of the above command, run the Following command:

`sudo snap install docker`



Verify



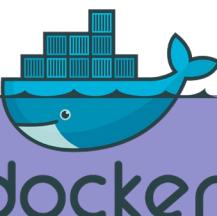
```
sohailimran@SI:~$ docker --version  
Docker version 28.0.4, build b8034c0
```



```
sohailimran@SI:~$
```



```
sohailimran@SI:~$ docker version  
Client: Docker Engine - Community  
Version:          28.0.4  
API version:     1.48  
Go version:      go1.23.7  
Git commit:      b8034c0  
Built:            Tue Mar 25 15:07:11 2025  
OS/Arch:          linux/amd64  
Context:          default  
  
Server: Docker Engine - Community  
Engine:  
  Version:          28.0.4  
  API version:     1.48 (minimum version 1.24)  
  Go version:      go1.23.7  
  Git commit:      6430e49  
  Built:            Tue Mar 25 15:07:11 2025  
  OS/Arch:          linux/amd64  
  Experimental:    false  
containerd:  
  Version:          1.7.27  
  GitCommit:        05044ec0a9a75232cad458027ca83437aae3f4da  
runc:  
  Version:          1.2.5  
  GitCommit:        v1.2.5-0-g59923ef  
docker-init:  
  Version:          0.19.0  
  GitCommit:        de40ad0  
sohailimran@SI:~$
```



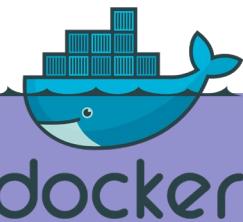
Setup

```
sudo systemctl enable docker
```

```
sudo systemctl start docker
```

```
sudo systemctl restart docker
```

```
sohailimran@SI:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
sohailimran@SI:~$ sudo systemctl start docker
sohailimran@SI:~$ sudo systemctl restart docker
sohailimran@SI:~$
```



Testing

```
sohailimran@SI: ~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete

Digest: sha256:fc08e727181e2668370f47db6319815c279ed887e2f01be96b94106bc2781430
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/
sohailimran@SI: ~$
```

