

Experiment - 9

Aim: To understand Docker Architecture and Container Life Cycle, install Docker and execute docker commands to manage images and interact with containers.

Theory:

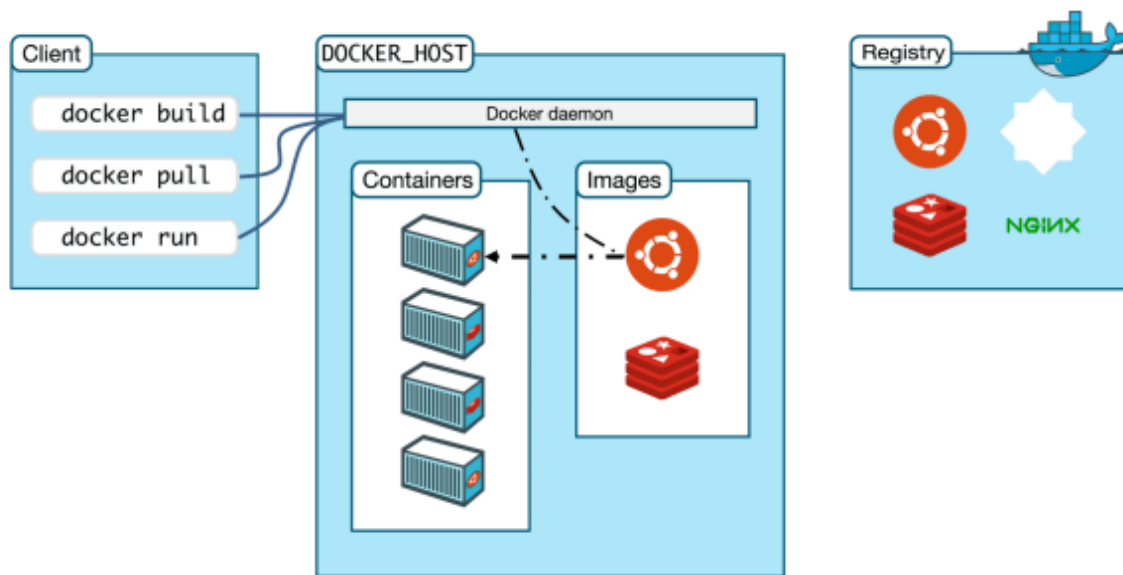
Containers are a software package into a logical box with everything that the application needs to run. That includes the operating system, application code, runtime, system tools, system libraries, and etc. Docker containers are built off Docker images. Since images are read-only, Docker adds a read-write file system over the read-only file system of the image to create a container.

Containers are compared with virtual machines (VMs). VMs are the guest operating system such as Linux or Windows that runs on top of a host operating system with virtualized access to the underlying hardware. Containers allow you to package your application together with libraries and other dependencies, providing isolated environments for running your software services.

Docker is an open-source platform based on Linux containers for developing and running applications inside containers. Docker is used to deploy many containers simultaneously on a given host. Containers are very fast and lightweight because they don't need the extra load of a hypervisor as they run directly within the host machine's kernel.

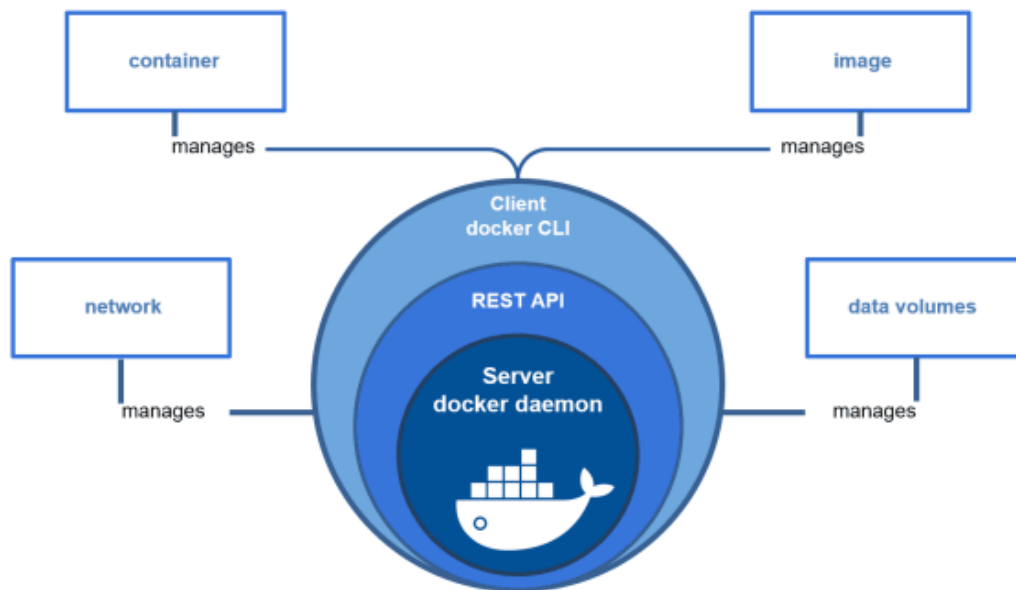
Docker Architecture and Components

Docker uses a client-server architecture. The docker client talks to the Docker daemon, which is used to building, running, and distributing the Docker containers. The Docker client and daemon communicate using a REST API, over UNIX sockets, or a network interface.

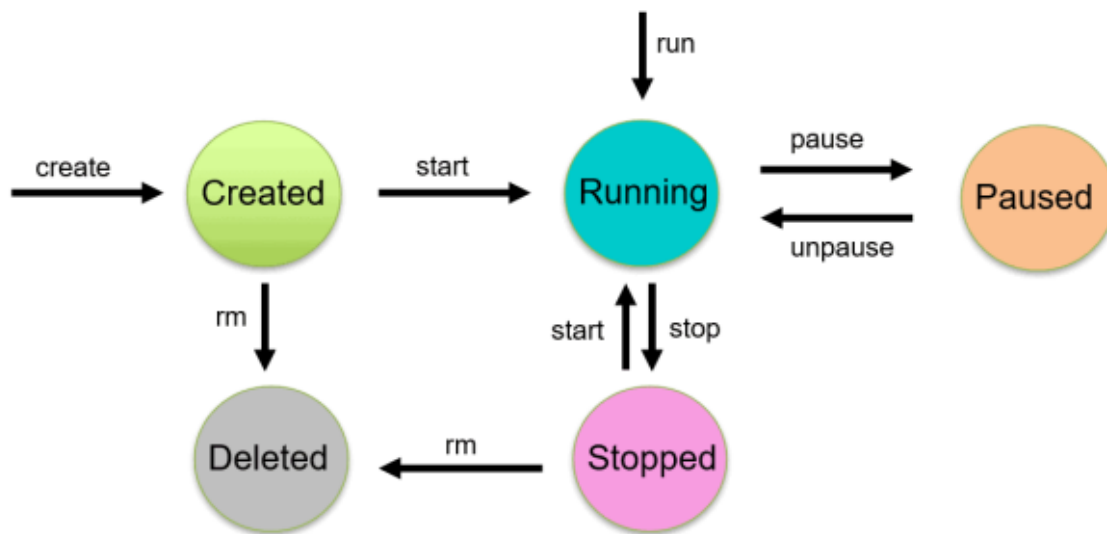


There are five major components in the Docker architecture:

- a) Docker Daemon listens to Docker API requests and manages Docker objects such as images, containers, networks and volumes.
- b) Docker Clients: With the help of Docker Clients, users can interact with Docker. Docker client provides a command-line interface (CLI) that allows users to run, and stop application commands to a Docker daemon.
- c) Docker Host provides a complete environment to execute and run applications. It comprises of the Docker daemon, Images, Containers, Networks, and Storage.
- d) Docker Registry stores Docker images. Docker Hub is a public registry that anyone can use, and Docker is configured to use images on Docker Hub by default. You can run your own registry on it.
- e) Docker Images are read-only templates that you build from a set of instructions written in Dockerfile. Images define both what you want your packaged application and its dependencies to look like what processes to run when it's launched.



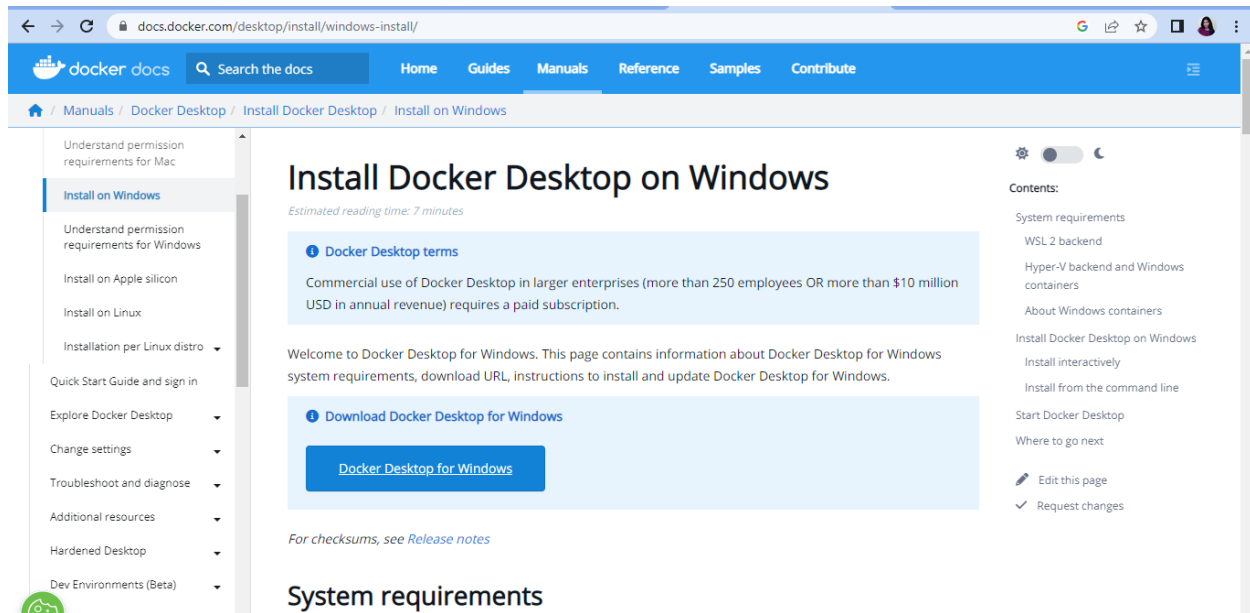
Docker Container Lifecycle Management



Implementation:

Installation of Docker:

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



The screenshot shows the Docker Docs website with the URL `docs.docker.com/desktop/install/windows-install/`. The page title is "Install Docker Desktop on Windows" with an estimated reading time of 7 minutes. The left sidebar contains a navigation menu with options like "Understand permission requirements for Mac", "Install on Windows", "Understand permission requirements for Windows", "Install on Apple silicon", "Install on Linux", "Installation per Linux distro", "Quick Start Guide and sign in", "Explore Docker Desktop", "Change settings", "Troubleshoot and diagnose", "Additional resources", "Hardened Desktop", and "Dev Environments (Beta)". The main content area includes a "Docker Desktop terms" section, a "Download Docker Desktop for Windows" button, and a "System requirements" section. The right sidebar lists "Contents" such as "System requirements", "WSL 2 backend", "Hyper-V backend and Windows containers", "About Windows containers", "Install Docker Desktop on Windows", "Install interactively", "Install from the command line", "Start Docker Desktop", and "Where to go next".

Installing Docker Desktop 4.0.1 (68347)

Configuration

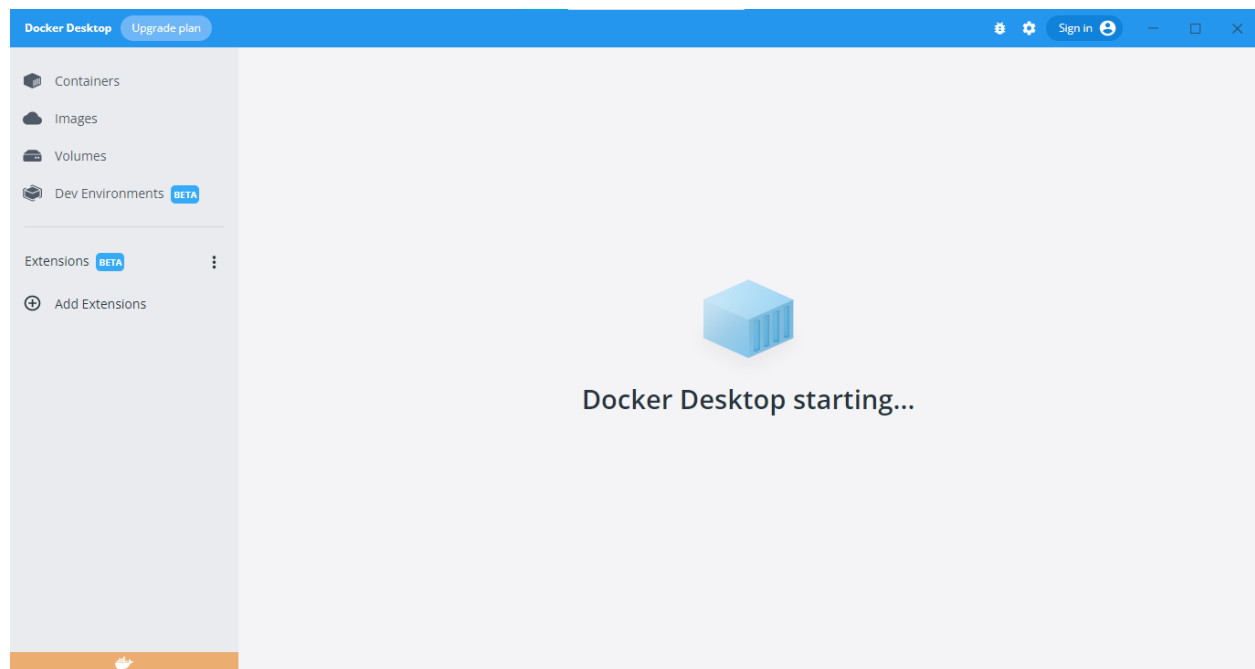
- ☒ Install required Windows components for WSL 2
- ☒ Add shortcut to desktop

Docker Desktop 4.0.1

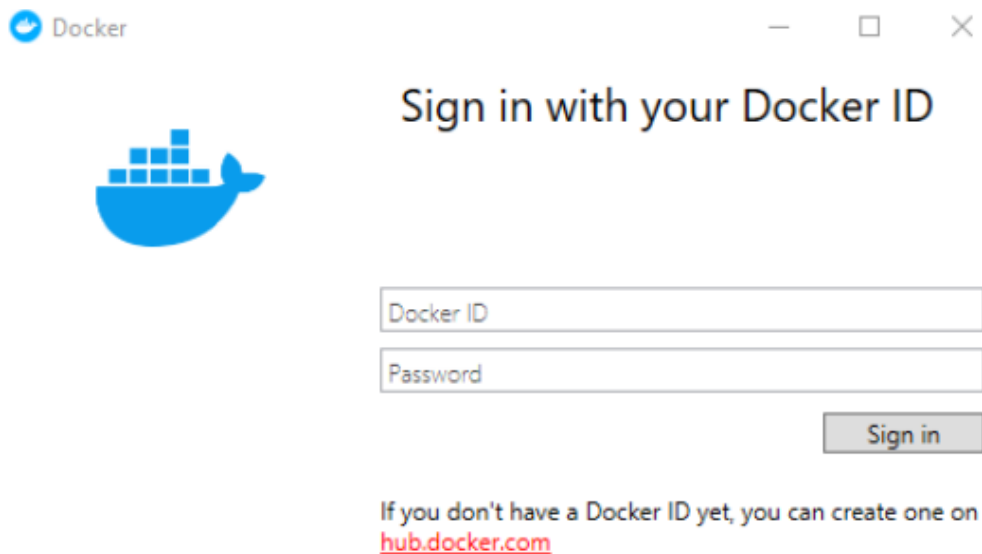
Installation succeeded

You must restart Windows to complete installation.

Close and restart

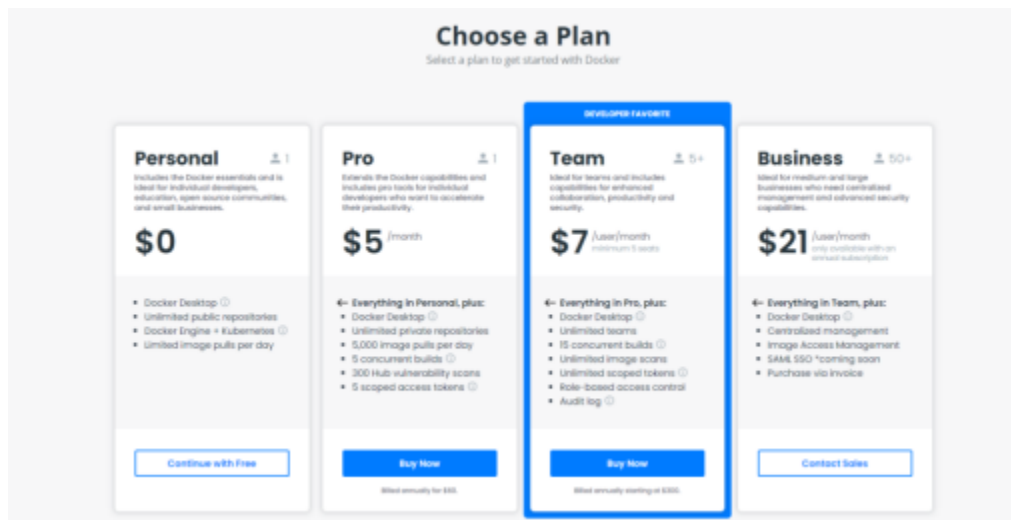


Now sign in to your account



The screenshot shows the Docker login interface. At the top left is the Docker logo. In the center, it says "Sign in with your Docker ID" with the Docker whale logo below it. There are two input fields: "Docker ID" and "Password". To the right of the password field is a "Sign in" button. Below the input fields, there is a link that says "If you don't have a Docker ID yet, you can create one on hub.docker.com".

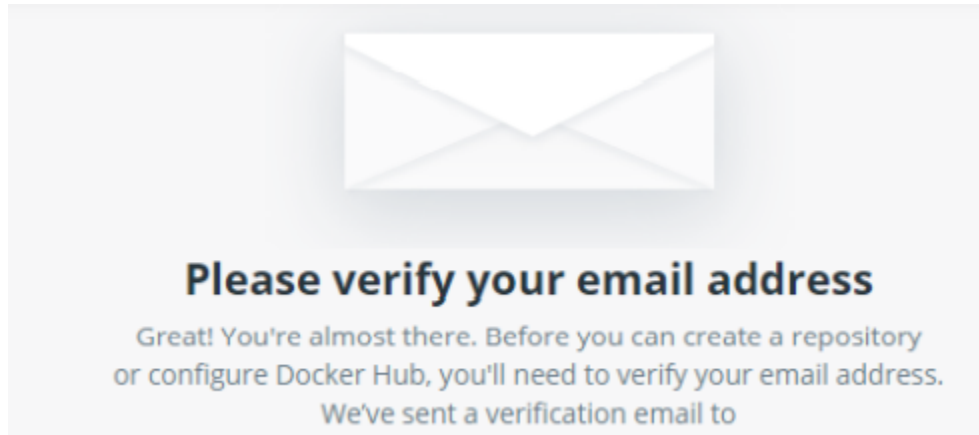
Choose plan as “personal” which is a free plan



The screenshot shows the "Choose a Plan" page with four options: Personal, Pro, Team, and Business. The Team plan is highlighted with a blue border. Below the plans, there is a "Verify your email address" link.

Plan	Price	Features	Button
Personal	\$0	Includes the Docker essentials and is ideal for individual developers, education, open source communities, and small businesses.	Continue with Free
Pro	\$5 /month	Everything in Personal, plus: <ul style="list-style-type: none">Docker DesktopUnlimited private repositories5,000 image pulls per day5 concurrent builds300 Hub vulnerability scans5 scoped access tokens	Buy Now
Team	\$7 /user/month (minimum 5 seats)	Everything in Pro, plus: <ul style="list-style-type: none">Docker DesktopUnlimited teams15 concurrent buildsUnlimited image scansUnlimited scoped tokensRole-based access controlAudit log	Buy Now
Business	\$21 /user/month (only available with an on-premise subscription)	Everything in Team, plus: <ul style="list-style-type: none">Docker DesktopCentralized managementImage Access ManagementSAML SSO (coming soon)Purchase via invoice	Contact Sales

Verify your email address



Now perform following commands:-

Listing out Containers

```
Windows PowerShell
PS C:\Users\Admin> docker container ls
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                    NAMES
142f84660449   nginx     "/docker-entrypoint..." 29 seconds ago Up 27 seconds  0.0.0.0:80->80/tcp      boring_kalam
PS C:\Users\Admin>
```

Listing out All Containers

- We can use the -a flag to the previous list command to find a list of all containers, even those which have stopped docker container ls -a

```

Windows PowerShell
PS C:\Users\Admin> docker container ls -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
142f84660449   nginx      "/docker-entrypoint..." About a minute ago Exited (0) 14 seconds ago
0239332cf480   sonarqube:latest "/opt/sonarqube/bin/..." 7 days ago     Exited (255) 24 minutes ago 0.0.0.0:9000->9000/tcp
PS C:\Users\Admin>

```

Show container logs

- We can use the command logs to show logs for a specified container - docker container logs <id>

```

Windows PowerShell
00->9000/tcp sonarqube
PS C:\Users\Admin> docker container logs 142f
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2022/10/06 03:37:29 [notice] 1#1: using the "epoll" event method
2022/10/06 03:37:29 [notice] 1#1: nginx/1.23.1
2022/10/06 03:37:29 [notice] 1#1: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/10/06 03:37:29 [notice] 1#1: OS: Linux 5.10.16.3-microsoft-standard-WSL2
2022/10/06 03:37:29 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/10/06 03:37:29 [notice] 1#1: start worker processes
2022/10/06 03:37:29 [notice] 1#1: start worker process 32
2022/10/06 03:37:29 [notice] 1#1: start worker process 33
2022/10/06 03:37:29 [notice] 1#1: start worker process 34
2022/10/06 03:37:29 [notice] 1#1: start worker process 35
2022/10/06 03:37:29 [notice] 1#1: start worker process 36
2022/10/06 03:37:29 [notice] 1#1: start worker process 37
2022/10/06 03:37:29 [notice] 1#1: start worker process 38
2022/10/06 03:37:29 [notice] 1#1: start worker process 39
2022/10/06 03:37:29 [notice] 1#1: start worker process 40
2022/10/06 03:37:29 [notice] 1#1: start worker process 41
2022/10/06 03:37:29 [notice] 1#1: start worker process 42
2022/10/06 03:37:29 [notice] 1#1: start worker process 43
2022/10/06 03:37:29 [notice] 1#1: start worker process 44
2022/10/06 03:37:29 [notice] 1#1: start worker process 45
2022/10/06 03:37:29 [notice] 1#1: start worker process 46
2022/10/06 03:37:29 [notice] 1#1: start worker process 47
172.17.0.1 - - [06/Oct/2022:03:37:40 +0000] "GET / HTTP/1.1" 200 615 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36" "-"
172.17.0.1 - - [06/Oct/2022:03:37:40 +0000] "GET /favicon.ico HTTP/1.1" 404 555 "http://localhost/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/105.0.0.0 Safari/537.36" "-"
2022/10/06 03:37:40 [error] 32#32: *1 open() "/usr/share/nginx/html/favicon.ico" failed (2: No such file or directory), client: 172.17.0.1, server: localhost, request: "GET /favicon.ico HTTP/1.1", host: "localhost", referer: "http://localhost/"
2022/10/06 03:38:36 [notice] 1#1: signal 3 (SIGQUIT) received, shutting down
2022/10/06 03:38:36 [notice] 32#32: gracefully shutting down
2022/10/06 03:38:36 [notice] 33#33: gracefully shutting down
2022/10/06 03:38:36 [notice] 34#34: gracefully shutting down

```

Listing out Images

- We can use the docker images command to show a list of docker images we locally have docker images

```

Windows PowerShell
PS C:\Users\Admin> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
nginx          latest    51086ed63d8c   15 hours ago   142MB
sonarqube      latest    2cf2f2494695   5 weeks ago    534MB
PS C:\Users\Admin>

```

Conclusion: Thus we successfully installed docker and executed docker commands to manage images and interact with containers.