

EXPERIMENT : 8

Date of Performance	
Date of Submission	

AIM

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

PROBLEM DEFINITION

Gain insight into Docker architecture and container lifecycle, including installing Docker and managing images and containers.

THEORY

Jenkins is an open-source automation tool created with Java. It is extensively used as a Continuous Integration (CI) and Continuous Deployment (CD) tool.

Maven:

- Maven primarily provides developers with:
 1. A comprehensive, reusable, and easily maintainable model for projects.
 2. Plugins or tools to interact with and operate within this model.
- Maven is a POM (Project Object Model)-based build automation and project management tool written in Java. However, it is compatible with projects written in C#, Python, Ruby, etc.

A few Maven features worth mentioning are:

1. Maven can be used to build projects into predefined output types like .jar, .war, metadata, etc.
2. Maven can automatically download necessary files from the repository when building a project.

Selenium Using Maven in Jenkins

Selenium is a widely used test automation framework for validating web applications across different combinations of browsers, platforms, and devices (or emulators). It is extensively used for testing areas such as functional testing, end-to-end testing, and more.

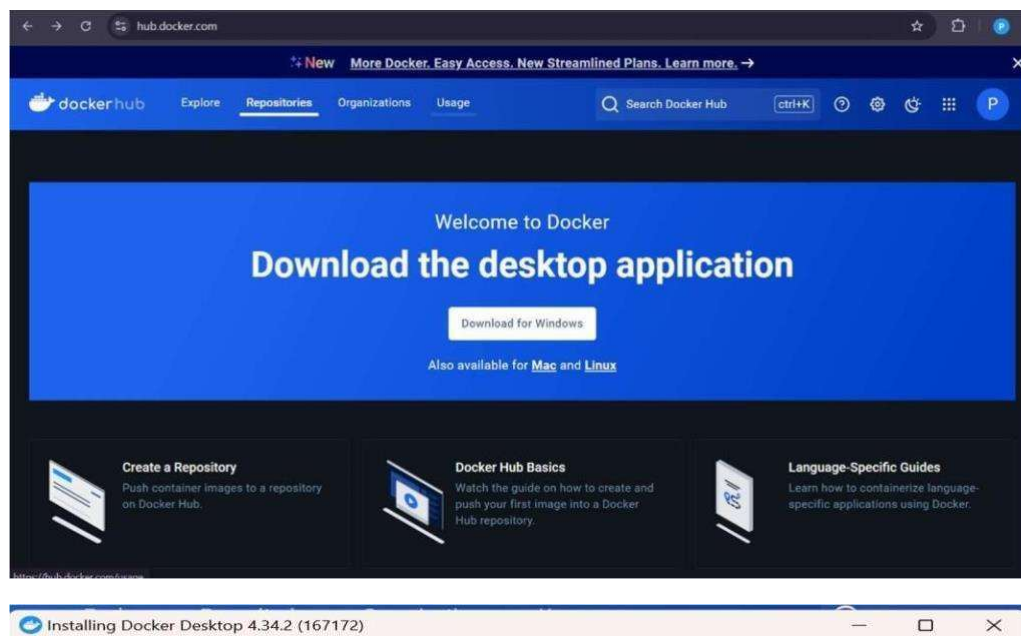
Why Jenkins and Selenium?

- Running Selenium tests in Jenkins allows you to execute your tests every time your software changes and deploy the software to new environments when the tests pass.

Advantages of Using Maven and Jenkins with Selenium:

- Whenever a change is made in the implementation, the changes are deployed in the test environment. Automation testing is performed continuously, and developers are kept informed about the build and test stage results.
- Test suites comprising many test scenarios might take a longer duration for testing. In such cases, a nightly build run can be scheduled for build and execution on the Jenkins server.

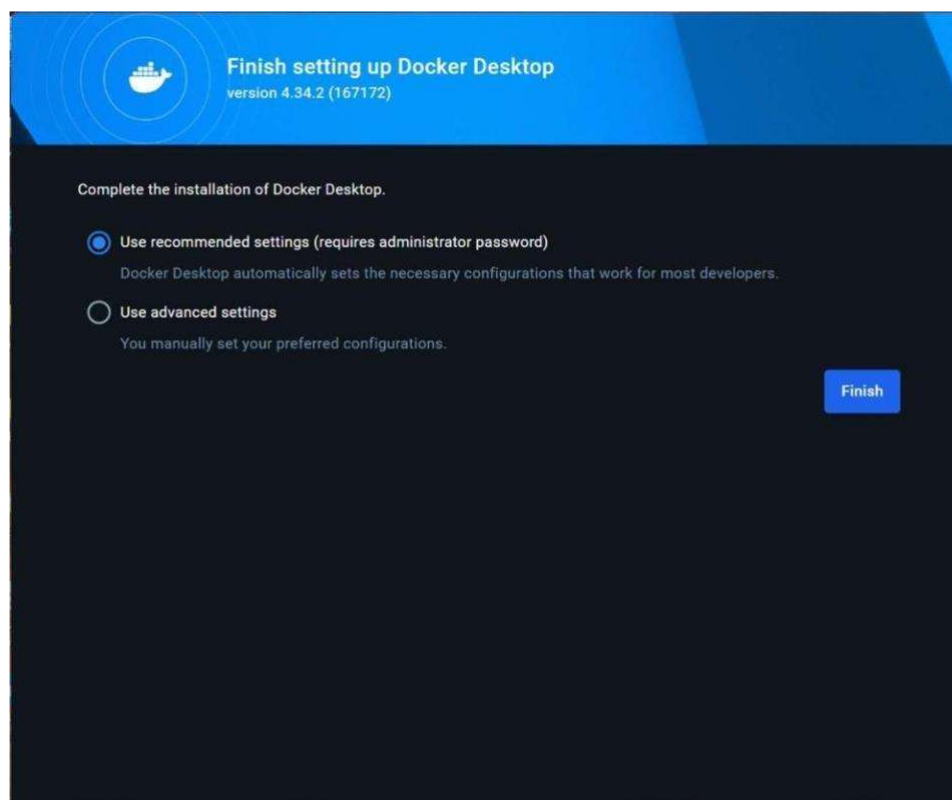
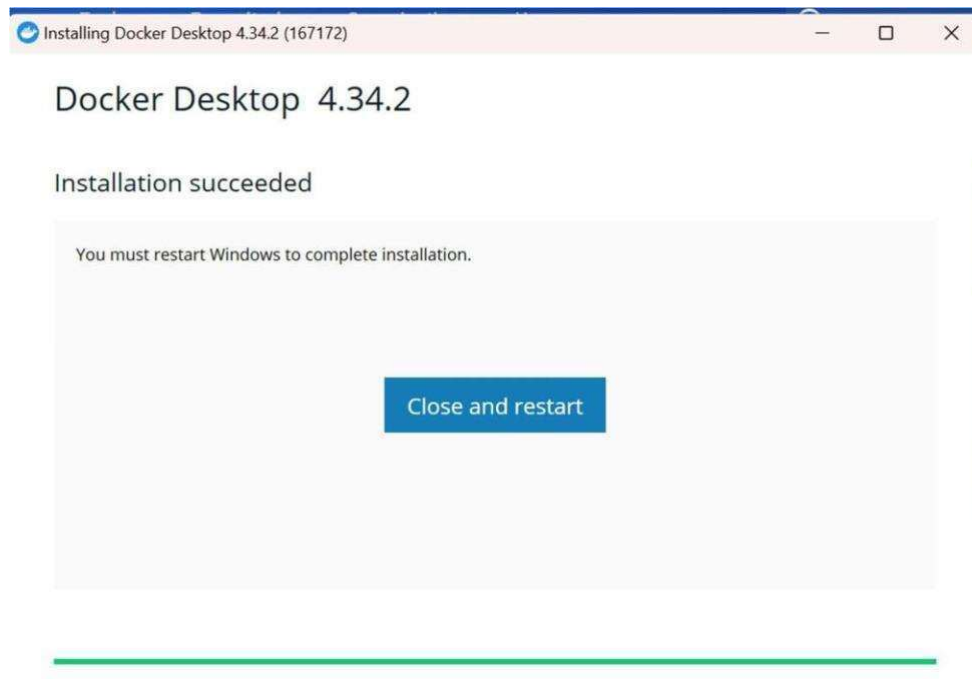
OUTPUT



Docker Desktop 4.34.2

Unpacking files...

```
Unpacking file: resources/docker-desktop.iso
Unpacking file: resources/ddvp.ico
Unpacking file: resources/config-options.json
Unpacking file: resources/componentsVersion.json
Unpacking file: resources/bin/docker-compose
Unpacking file: resources/bin/docker
Unpacking file: resources/.gitignore
Unpacking file: InstallerCli.pdb
Unpacking file: InstallerCli.exe.config
Unpacking file: frontend/vk_swiftshader_icd.json
Unpacking file: frontend/v8_context_snapshot.bin
Unpacking file: frontend/snapshot_blob.bin
Unpacking file: frontend/resources/regedit/vbs/util.vbs
Unpacking file: frontend/resources/regedit/vbs/regUtil.vbs
```



Docker –version, docker ps, docker run

```
Command Prompt
C:\Users\Piyush>docker --version
Docker version 27.2.0, build 3ab4256

C:\Users\Piyush>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES

C:\Users\Piyush>docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
hello-world   latest   91fb4b041da2  17 months ago  24.4kB

C:\Users\Piyush>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/

C:\Users\Piyush>
```

Docker login, docker search apache

```
Command Prompt
C:\Users\Piyush>docker login
Authenticating with existing credentials...
Login Succeeded

C:\Users\Piyush>docker search apache
NAME                                DESCRIPTION                                STARS   OFFICIAL
apache/airflow                     Apache Airflow                             540
apache/superset                     Apache Superset                             267
apache/nifi                         Apache NiFi unofficial binary build         295
apache/tika                         Container images for Apache Tika Server (htt... 35
apache/zeppelin                     Apache Zeppelin                             176
apache/druid                        Apache Druid                                58
apache/arrow-dev                    Apache Arrow convenience images for developm... 3
apache/skywalking-ui                Apache SkyWalking Web UI                    105
bitnami/apache                      Bitnami container image for Apache           94
apache/skywalking-oap-server        Apache SkyWalking OAP Server                159
apache/apisix                       Apache APISIX: Cloud-Native API Gateway      92
apache/cassandra-testing-ubuntu2004-javall-w-dependencies https://github.com/apache/cassandra-builds/t... 1
apache/nifi-registry                Unofficial convenience binaries for Apache N... 43
apache/couchdb                      Unofficial convenience binaries for CouchDB,... 26
apache/spark                        Apache Spark                                75
apache/fineract                     Apache Fineract                             14
apache/apisix-dashboard              https://github.com/apache/apisix-dashboard   36
apache/yunikorn                     Apache Yunikorn                              2
apache/kafka                        Apache Kafka                                 79
apache/solr-operator                A Kubernetes Operator to manage and maintain... 2
apache/cassandra-testing-ubuntu2004-javall https://github.com/apache/cassandra-builds/t... 2
apache/camel-k                      Apache Camel                                 7
apache/karaf                        Apache Karaf                                 6
apache/pegasus                      Apache Pegasus (incubating)                 4
apache/beam_python3.8_sdk           Apache Beam SDK - Python 3.8                9
```

Docker search redis

Docker inspect

```
Command Prompt
C:\Users\Piyush>docker inspect 579e7d5047
[
  {
    "Id": "579e7d5047569d8808f1ff967e31f392961dfe76a02d147dab6574c7732e1c51",
    "Created": "2024-09-30T14:35:25.8853047Z",
    "Path": "/hello",
    "Args": [],
    "State": {
      "Status": "exited",
      "Running": false,
      "Paused": false,
      "Restarting": false,
      "OOMKilled": false,
      "Dead": false,
      "Pid": 0,
      "ExitCode": 0,
      "Error": "",
      "StartedAt": "2024-09-30T14:35:25.988555497Z",
      "FinishedAt": "2024-09-30T14:35:26.24237903Z"
    },
    "Image": "sha256:91fb8b01da273d5a3273b6d587d62d518308a6ad268b28628f7897b93171b2",
    "ResolvConfPath": "/var/lib/docker/containers/579e7d5047569d8808f1ff967e31f392961dfe76a02d147dab6574c7732e1c51/resolv.conf",
    "HostnamePath": "/var/lib/docker/containers/579e7d5047569d8808f1ff967e31f392961dfe76a02d147dab6574c7732e1c51/hostname",
    "HostsPath": "/var/lib/docker/containers/579e7d5047569d8808f1ff967e31f392961dfe76a02d147dab6574c7732e1c51/hosts",
    "LogPath": "/var/lib/docker/containers/579e7d5047569d8808f1ff967e31f392961dfe76a02d147dab6574c7732e1c51/579e7d5047569d8808f1ff967e31f392961dfe76a02d147dab6574c7732e1c51-json.log",
    "Name": "/zealous_edison",
    "RestartCount": 0,
    "Driver": "overlayfs",
    "Platform": "linux",
    "MountLabel": "",
    "ProcessLabel": "",
    "AppArmorProfile": "",
    "ExecIDs": null,
    "HostConfig": {
      "Binds": null,
      "ContainerIDFile": "",
      "LogConfig": {
        "Type": "json-file",
        "Config": {}
      },
      "NetworkMode": "bridge",
      "PortBindings": {},
      "RestartPolicy": {

```

```
Command Prompt
"Ent
ctrl+alt+I
"OnB
"Labels": {}
},
"NetworkSettings": {
  "Bridge": "",
  "SandboxID": "",
  "SandboxKey": "",
  "Ports": {},
  "HairpinMode": false,
  "LinkLocalIPv6Address": "",
  "LinkLocalIPv6PrefixLen": 0,
  "SecondaryIPAddresses": null,
  "SecondaryIPv6Addresses": null,
  "EndpointID": "",
  "Gateway": "",
  "GlobalIPv6Address": "",
  "GlobalIPv6PrefixLen": 0,
  "IPAddress": "",
  "IPPrefixLen": 0,
  "IPv6Gateway": "",
  "MacAddress": "",
  "Networks": {
    "bridge": {
      "IPAMConfig": null,
      "Links": null,
      "Aliases": null,
      "MacAddress": "",
      "DriverOpts": null,
      "NetworkID": "5622d1d089a420441c9bb2e4b0c56912487ac12ad75f18d7b423863fdeedb125",
      "EndpointID": "",
      "Gateway": "",
      "IPAddress": "",
      "IPPrefixLen": 0,
      "IPv6Gateway": "",
      "GlobalIPv6Address": "",
      "GlobalIPv6PrefixLen": 0,
      "DNSNames": null
    }
  }
}
}
```

Docker system df

```
Command Prompt
C:\Users\Piyush>docker system df

```

TYPE	TOTAL	ACTIVE	SIZE	RECLAIMABLE
Images	3	3	269MB	~8.284e+07B (~30%)
Containers	5	0	4.198MB	4.198MB (100%)
Local Volumes	0	0	0B	0B
Build Cache	0	0	0B	0B

Docker info

```
Command Prompt
C:\Users\Piyush>docker info
Client:
Version:      27.2.0
Context:      desktop-linux
Debug Mode:   false
Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.16.2-desktop.1
    Path:      C:\Program Files\Docker\cli-plugins\docker-buildx.exe
  compose: Docker Compose (Docker Inc.)
    Version:  v2.29.2-desktop.2
    Path:      C:\Program Files\Docker\cli-plugins\docker-compose.exe
  debug: Get a shell into any image or container (Docker Inc.)
    Version:  0.9.34
    Path:      C:\Program Files\Docker\cli-plugins\docker-debug.exe
  desktop: Docker Desktop commands (Alpha) (Docker Inc.)
    Version:  v0.0.15
    Path:      C:\Program Files\Docker\cli-plugins\docker-desktop.exe
  dev: Docker Dev Environments (Docker Inc.)
    Version:  v0.1.2
    Path:      C:\Program Files\Docker\cli-plugins\docker-dev.exe
  extension: Manages Docker extensions (Docker Inc.)
    Version:  v0.2.25
    Path:      C:\Program Files\Docker\cli-plugins\docker-extension.exe
  feedback: Provide feedback, right in your terminal! (Docker Inc.)
    Version:  v1.0.5
    Path:      C:\Program Files\Docker\cli-plugins\docker-feedback.exe
  init: Creates Docker-related starter files for your project (Docker Inc.)
    Version:  v1.3.0
    Path:      C:\Program Files\Docker\cli-plugins\docker-init.exe
  sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc.)
    Version:  0.6.0
    Path:      C:\Program Files\Docker\cli-plugins\docker-sbom.exe
  scout: Docker Scout (Docker Inc.)
    Version:  v1.13.0
    Path:      C:\Program Files\Docker\cli-plugins\docker-scout.exe
```

```
Command Prompt
Logging Driver: json-file
Cgroup Driver: cgroupfs
Cgroup Version: 1
Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog
Swarm: inactive
Runtimes: io.containerd.runc.v2 nvidia runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 8fc6bfff51318944179630522a095cc9dbf9f353
runc version: v1.1.13-0-g58aa920
init version: de40ad0
Security Options:
  seccomp
    Profile: unconfined
Kernel Version: 5.15.153.1-microsoft-standard-WSL2
Operating System: Docker Desktop
OSType: linux
Architecture: x86_64
CPUs: 14
Total Memory: 7.573GiB
Name: docker-desktop
ID: 53aa8b0b-9138-4142-b186-6eed9a791eca
Docker Root Dir: /var/lib/docker
Debug Mode: false
HTTP Proxy: http.docker.internal:3128
HTTPS Proxy: http.docker.internal:3128
No Proxy: hubproxy.docker.internal
Labels:
  com.docker.desktop.address=npipe://\\.\pipe\docker_cli
Experimental: false
Insecure Registries:
  hubproxy.docker.internal:5555
127.0.0.0/8
Live Restore Enabled: false

WARNING: No blkio throttle.read_bps.device support
WARNING: No blkio throttle.write_bps.device support
WARNING: No blkio throttle.read_iops.device support
WARNING: No blkio throttle.write_iops.device support
WARNING: daemon is not using the default seccomp profile

C:\Users\Piyush>
```

CONCLUSION:

As a result, we understood Docker architecture and container life cycle by installing Docker and executing commands to manage images and interact with containers effectively.