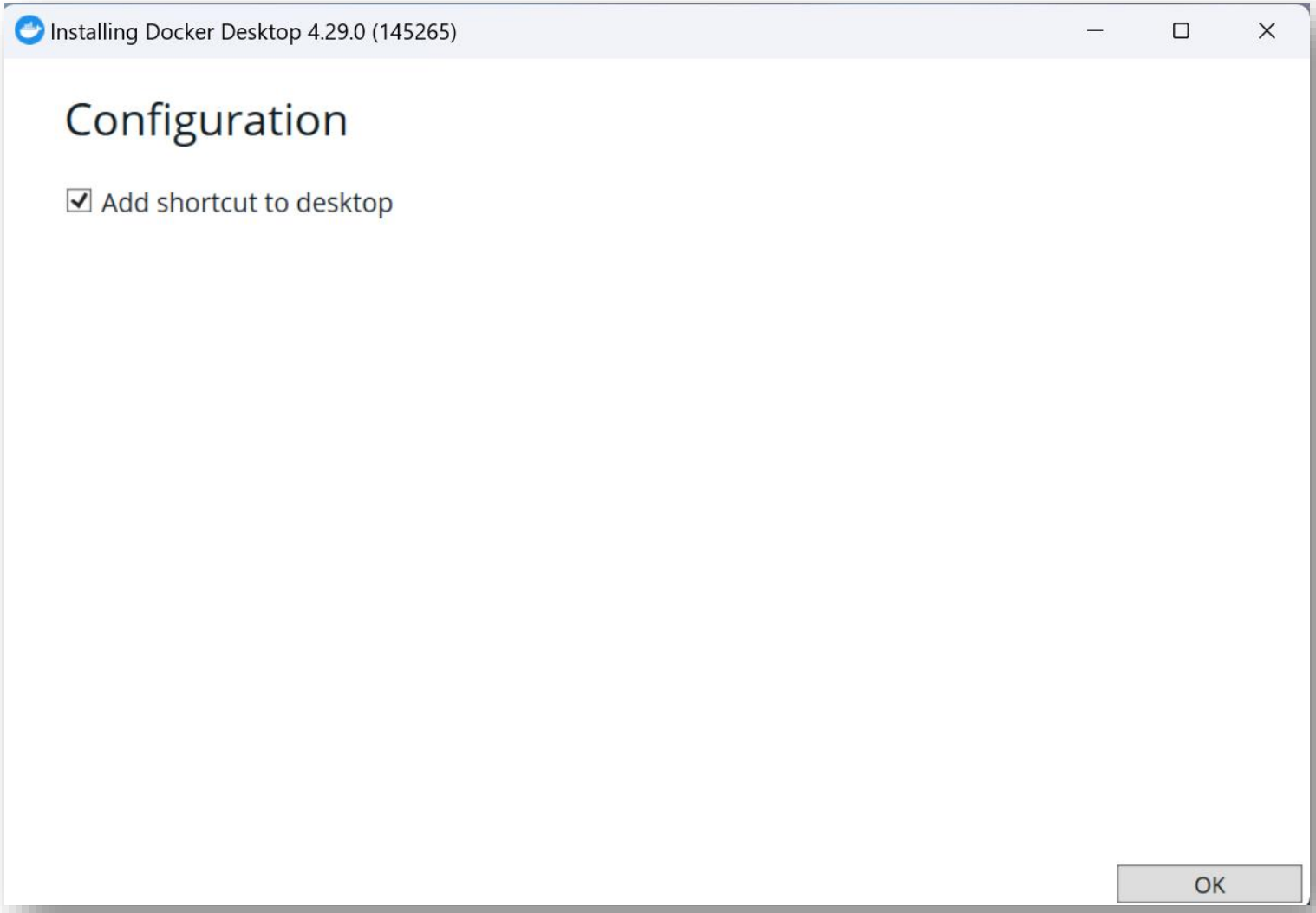
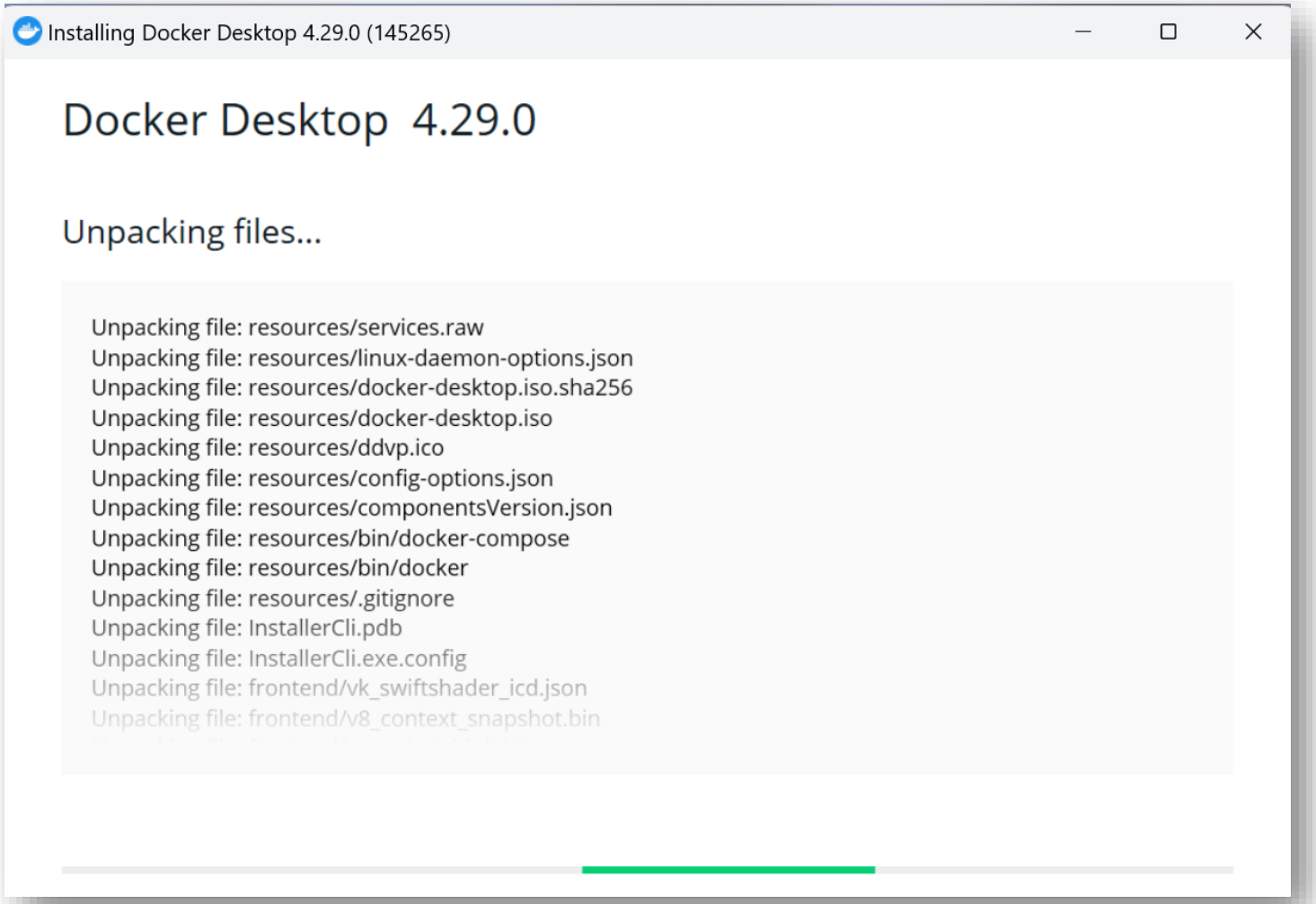
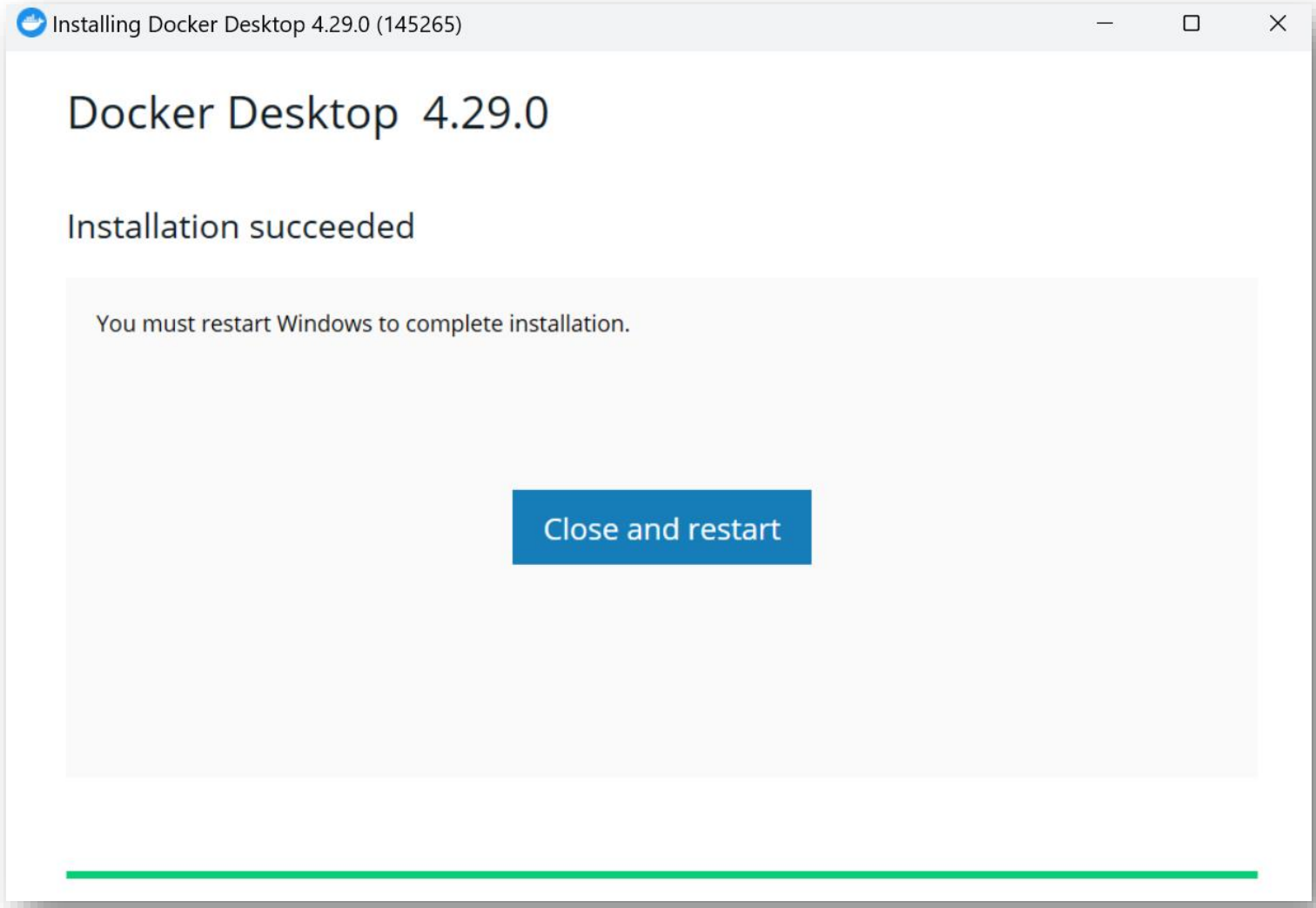
Docker Installation

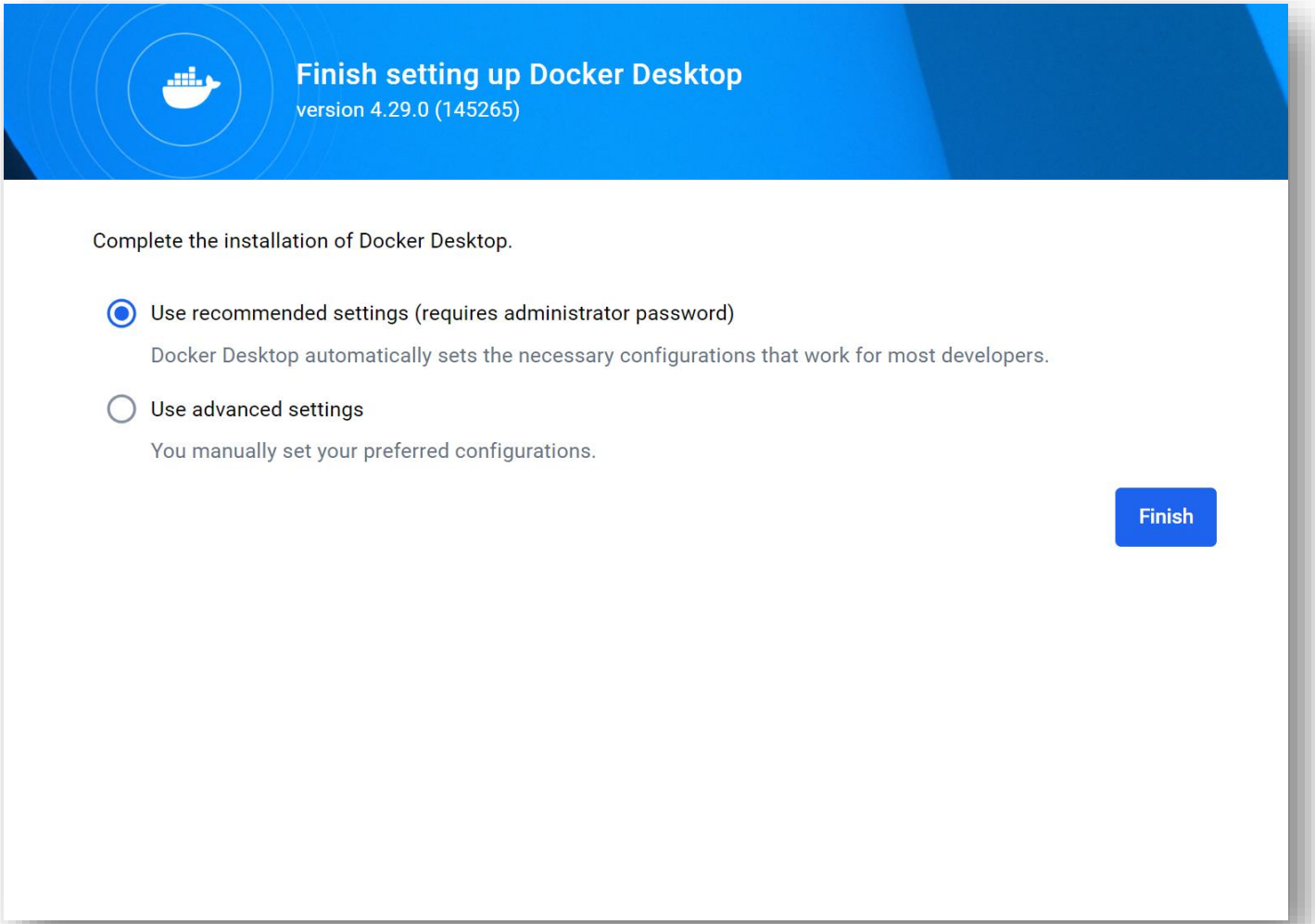
Step 1: Download & Install Docker

https://www.docker.com/products/docker-desktop/


Installing Docker Desktop 4.29.0 (145265)
Configuration
Add shortcut to desktop
OK


Installing Docker Desktop 4.29.0 (145265)
Docker Desktop 4.29.0
Unpacking files...
Unpacking file: resources/services.raw
Unpacking file: resources/linux-daemon-options.json
Unpacking file: resources/docker-desktop.iso.sha256
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

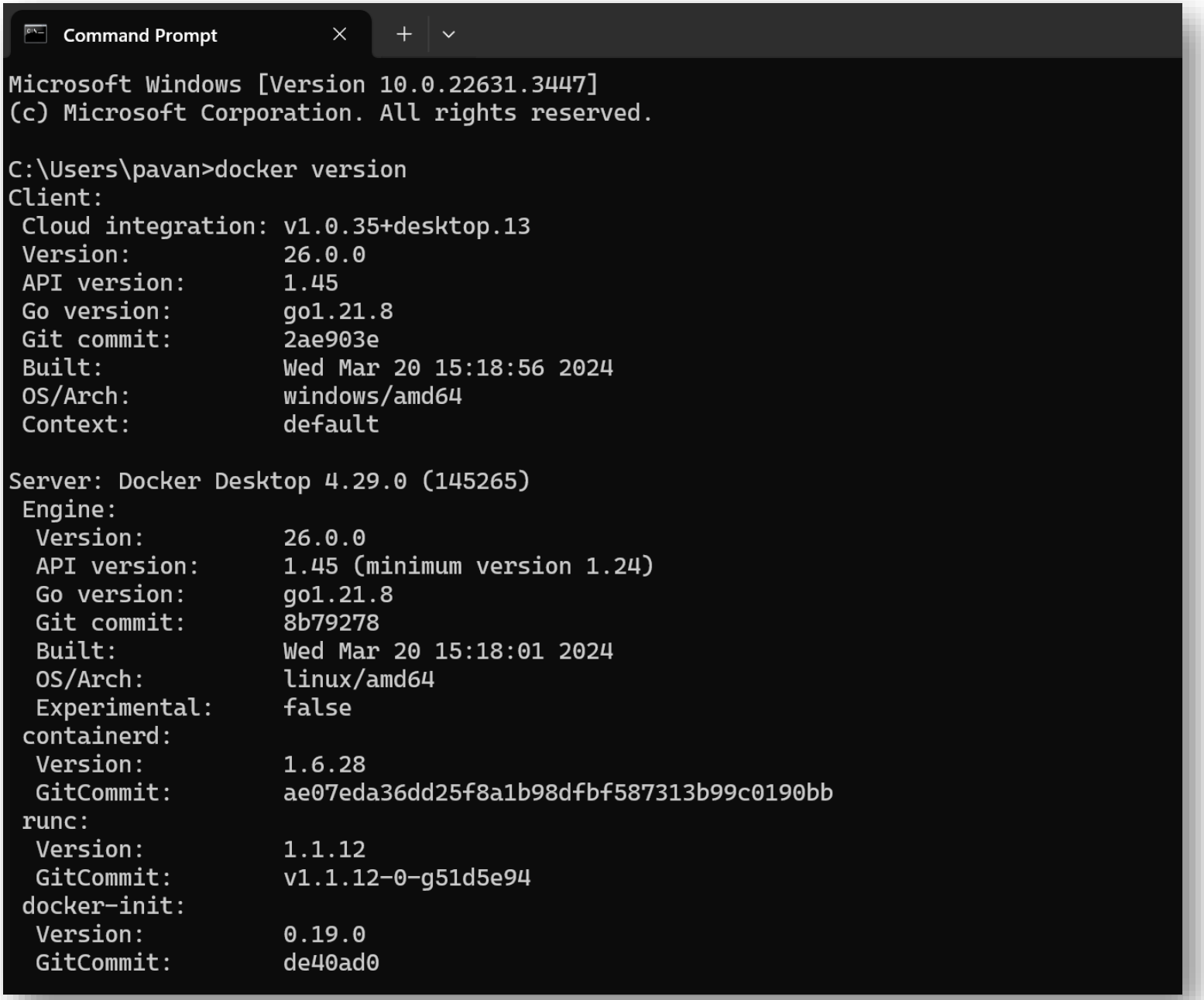

Installing Docker Desktop 4.29.0 (145265)
Docker Desktop 4.29.0
Installation succeeded
You must restart Windows to complete installation.
Close and restart

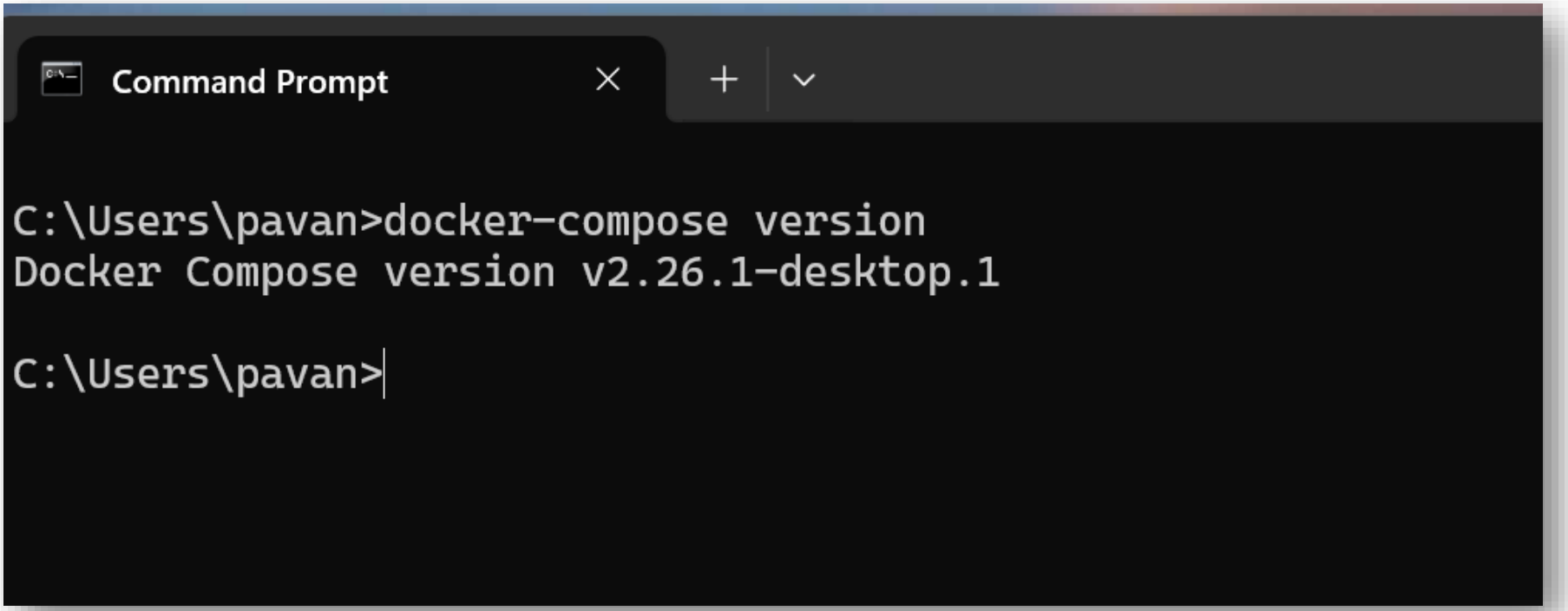

Finish setting up Docker Desktop
version 4.29.0 (145265)
Complete the installation of Docker Desktop.
Use recommended settings (requires administrator password)
Docker Desktop automatically sets the necessary configurations that work for most developers.
Use advanced settings
You manually set your preferred configurations.
Finish

Step 2: Verify Installation of Docker

```
Docker version

Docker-compose version
```


Command Prompt
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.
C:\Users\pavan>docker version
Client:
 Cloud integration: v1.0.35+desktop.13
 Version: 26.0.0
 API version: 1.45
 Go version: go1.21.8
 Git commit: 2ae903e
 Built: Wed Mar 20 15:18:56 2024
 OS/Arch: windows/amd64
 Context: default
Server: Docker Desktop 4.29.0 (145265)
 Engine:
  Version: 26.0.0
  API version: 1.45 (minimum version 1.24)
  Go version: go1.21.8
  Git commit: 8b79278
  Built: Wed Mar 20 15:18:01 2024
  OS/Arch: linux/amd64
  Experimental: false
 containerd:
  Version: 1.6.28
  GitCommit: ae07eda36dd25f8a1b98dfbf587313b99c0190bb
 runc:
  Version: 1.1.12
  GitCommit: v1.1.12-0-g51d5e94
 docker-init:
  Version: 0.19.0
  GitCommit: de40ad0


Command Prompt
C:\Users\pavan>docker-compose version
Docker Compose version v2.26.1-desktop.1
C:\Users\pavan>

Docker Commands

Basic Commands

1. **docker version**
    - Displays the Docker version installed on the system.
2. **docker -v**
    - Short form of **docker version**. It shows the Docker version.
3. **docker info**
    - Provides detailed information about the Docker installation.
4. **docker --help**
    - Displays general help information.
    - **Example**: To get information about specific commands:
        - **docker images --help**: Details about managing images.
        - **docker run --help**: Details about running containers.
5. **docker login**
    - Logs into a Docker registry, such as Docker Hub. Used for push or pull docker images from Docker Hub.

Images Commands

6. **docker images**
    - Lists all the Docker images present on the machine.
7. **docker pull**
    - Pulls an image from a Docker registry. You can find Docker images here: https://hub.docker.com/search?q=&type=image
        - **Example**: **docker pull ubuntu**
8. **docker rmi**
    - Removes Docker images.
        - **docker images -q**: Lists image IDs.
        - **docker rmi <image ID>**: Deletes the specified image.
        - After deletion, confirm with **docker images**.

Container Commands

9. **docker ps** & **docker run**
    - **docker ps**: Lists running containers.
    - **docker run <image>**: Creates a container from a specified image. If local image is not available then it will pull from Docker Hub automatically.
        - **Example**: **docker run ubuntu**.
        - **Example**: **docker run -it ubuntu** //For interaction
10. **docker start**
    - Starts a stopped container.
        - **Example**: **docker start <container id>**.
11. **docker stop**
    - Stops a running container.
        - **Example**: **docker stop <container id>**.
12. **docker rm**
    - Removes a container.
        - **Example**: **docker rm <container id or name>**.

System Commands

12. **docker stats**
    - Provides resource usage statistics for running containers, such as CPU, memory, etc.
13. **docker system df**
    - Displays disk usage related to Docker.
14. **docker system prune**
    - Cleans up unused data, such as stopped containers.
        - **docker system prune -f**: Forcefully removes all stopped containers.

These commands form the basic toolkit for managing Docker containers and images, as well as maintaining the Docker environment.

Selenium Grid Setup with Docker Containers

Pull Docker Images

Pull Selenium-hub image using command

docker pull selenium/hub

Pull FireFox image using command

docker pull selenium/node-firefox

Pull Chrome image using below command

docker pull selenium/node-chrome

Verify Images

docker images

Running Docker Containers by using below commands.

```
docker network create grid

docker run -d -p 4442-4444:4442-4444 --net grid --name selenium-hub selenium/hub

docker run -d --net grid -e SE_EVENT_BUS_HOST=selenium-hub -e SE_EVENT_BUS_PUBLISH_PORT=4442 -e SE_EVENT_BUS_SUBSCRIBE_PORT=4443 selenium/node-chrome

docker run -d --net grid -e SE_EVENT_BUS_HOST=selenium-hub -e SE_EVENT_BUS_PUBLISH_PORT=4442 -e SE_EVENT_BUS_SUBSCRIBE_PORT=4443 selenium/node-firefox
```

When you are done using the Grid, and the containers have exited, the network can be removed with the following command:

`docker network rm grid` # Removes the grid network

Selenium Grid Setup with

docker-compose.yaml file

1) Create a file docker-compose.yaml with Required config (Ref: https://github.com/SeleniumHQ/docker-selenium)

docker-compose.yaml

```
version: '3'
services:
  selenium-hub:
    image: selenium/hub
    ports:
      - "4442-4444:4442-4444"
    networks:
      - grid


  node-chrome:
    image: selenium/node-chrome
    environment:
      - SE_EVENT_BUS_HOST=selenium-hub
      - SE_EVENT_BUS_PUBLISH_PORT=4442
      - SE_EVENT_BUS_SUBSCRIBE_PORT=4443
    networks:
```

```
    - grid


  node-firefox:
    image: selenium/node-firefox
    environment:
      - SE_EVENT_BUS_HOST=selenium-hub
      - SE_EVENT_BUS_PUBLISH_PORT=4442
      - SE_EVENT_BUS_SUBSCRIBE_PORT=4443
    networks:
      - grid


networks:
  grid:
    driver: bridge
```

2) Run docker-compose.yaml

docker-compose up

3) To check hub & nodes running state:

http://localhost:4444/grid/console

4) To stop the grid and cleanup the created containers:

docker-compose down