Installing Docker: Windows, Mac, and Linux

Three different installers, one reason to bother

Every useful self-hosted tool you will ever want to run ships as a Docker image. Document e-signing, a transcription server, a PDF converter, an analytics dashboard, a private AI model. The vendor's install instructions are three lines of Docker and nothing else, because that is how software gets distributed now.

So the choice is not really "should I learn Docker." It is "am I going to keep paying $40 a month for a hosted tool I could run myself in ten minutes."

This tip covers the install only. If you want the concepts first, read *What Docker Is and Why "It Runs in a Container" Matters*; this one assumes you just want it working.

What you are actually installing

Two different products share the name.

Docker Engine is the thing that runs containers. Command line only, no GUI, Linux only. This is what runs on every server on earth.

Docker Desktop is Docker Engine plus a Linux virtual machine, a dashboard, and Docker Compose, bundled into one installer for Windows and Mac. Windows and macOS cannot run Linux containers natively, so Desktop quietly runs a small Linux VM in the background and points your `docker` command at it.

Rule of thumb: your laptop gets Docker Desktop; your server gets Docker Engine. Never install Docker Desktop on a headless Linux server.

Before you start: virtualization

On Windows and Mac, Docker needs hardware virtualization turned on in the firmware. On most machines it already is. On Windows, open Task Manager, go to Performance, click CPU, and look for "Virtualization: Enabled." If it says Disabled, reboot into BIOS/UEFI and turn on Intel VT-x or AMD-V (sometimes labeled SVM Mode). This one setting causes most failed installs.

Budget 8 GB of RAM and about 6 GB of free disk. Real container stacks eat more disk than you expect.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Windows

Windows 10 64-bit (21H2 or newer) or Windows 11. Home edition is fine; the WSL2 backend removed the old Pro requirement.

1. Open PowerShell as Administrator and install the Linux subsystem:

```
wsl --install
```

1. Reboot. Yes, actually reboot.
2. Install Docker Desktop. Either download the installer from docker.com and double-click it, or do it from the terminal:

```
winget install -e --id Docker.DockerDesktop
```

1. Launch Docker Desktop from the Start menu. The first start takes a minute while it provisions the WSL2 pieces. When the whale in the system tray stops animating and the dashboard says "Engine running," it is up.

Optional but worth it: WSL2 will happily take most of your RAM. Create `%USERPROFILE%\.wslconfig` with a ceiling:

```
[wsl2]
memory=8GB
processors=4
autoMemoryReclaim=gradual
```

Then run `wsl --shutdown` to apply it.

macOS

macOS 13 Ventura or newer. Apple silicon and Intel both work, but they are different downloads; grab the one matching your chip.

Download the `.dmg` from docker.com, open it, drag Docker to Applications, launch it, and approve the privileged-helper prompt. Or:

```
brew install --cask docker
```

One Apple silicon gotcha: a lot of images are still built for amd64 only. They will run under emulation and they will be slow. Prefer an `arm64` tag when the project publishes one, and turn on Rosetta emulation in Settings, General if a container refuses to start.

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

Linux (Ubuntu and Debian)

Do not run `apt install docker.io`. Do not run `apt install docker-compose`. Both are the distro's stale forks; the second one is Compose v1, which was retired years ago and does not read modern compose files. Use Docker's own repository.

```
sudo apt-get update
sudo apt-get install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/
keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/
docker.asc] https://download.docker.com/linux/ubuntu $(. /etc/os-release &&
echo "$VERSION_CODENAME") stable" \
  | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
sudo apt-get install -y docker-ce docker-ce-cli containerd.io docker-buildx-
plugin docker-compose-plugin
```

On Debian, swap both `ubuntu` paths for `debian`. On Ubuntu derivatives like Mint, `$VERSION_CODENAME` returns the derivative's name and the repository will 404; use `$UBUNTU_CODENAME` instead.

Start it and let it come up on boot:

```
sudo systemctl enable --now docker
```

By default every `docker` command needs `sudo`. To stop that:

```
sudo usermod -aG docker $USER
newgrp docker
```

Understand what you just did. The `docker` group is root equivalent; anyone in it can mount your filesystem into a container and read anything. On a personal workstation that is a fair trade. On a box holding client files, think about it first, or look at rootless mode.

Verify (all three platforms)

```
docker version
docker compose version
docker run hello-world
```

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

`docker version` should print both a Client and a Server block. Client only means the engine is not reachable yet. `docker compose version` should report v2 or higher; note it is `docker compose` with a space, not the old hyphenated `docker-compose`. And `hello-world` proves the whole chain works: the client talked to the daemon, the daemon pulled an image from the internet, and it ran.

Three commands, thirty seconds. Run them before you conclude anything is broken.

The licensing question, because you are a business

Docker Engine on Linux is open source and free, always, no asterisk.

Docker Desktop is free under the Docker Personal plan for individuals, education, and companies with fewer than 250 employees **and** under $10 million in annual revenue. Above either threshold, commercial use requires a paid seat. A solo firm or a five-person office is comfortably inside the free tier; a 300-lawyer firm is not, and "only two people use it" does not help, since the threshold is the organization, not the seat count.

If you are the IT decision maker at a firm that crosses the line, the cheap answer is usually Docker Engine on a Linux box plus SSH, which no license covers.

When it does not work

"Cannot connect to the Docker daemon." The engine is not running. Linux: `sudo systemctl start docker`. Windows and Mac: Docker Desktop is not open, or it is still starting.

"Permission denied while trying to connect to the Docker daemon socket." Linux only; your group membership has not taken effect. Log out and back in, or run `newgrp docker`.

"Package docker-ce has no installation candidate." The repository line did not land. Re-run the keyring and `echo` steps, then `sudo apt-get update`.

Docker Desktop refuses to start on Windows. Virtualization is off in BIOS, or WSL2 never installed. Run `wsl --status` and check Task Manager as described above.

Everything is slow on a Mac. You are probably running an amd64 image under emulation. Check with `docker inspect <image> | grep Architecture`.

Using AI to help with this

> **You do not need to memorize any of the above.** Paste the failure and let the model read it:

"I ran `docker run hello-world` on [Windows 11 / macOS Sequoia / Ubuntu 24.04] and got this error: [paste]. What is wrong and what exact command fixes it?"

"I want to self-host [tool] on my [OS]. Give me the docker compose file and the exact commands to bring it up, and tell me what ports it will open."

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.

"Explain what this install script is about to do to my machine before I run it: [paste]."

Educational only. Not legal advice. No attorney-client relationship. ©2026 Chariton Media.