# Install Docker Desktop on Windows

Table of contents



---

**Docker Desktop terms**

Commercial use of Docker Desktop in larger enterprises (more than 250 employees OR more than $10 million USD in annual revenue) requires a paid subscription.

This page provides download links, system requirements, and step-by-step installation instructions for Docker Desktop on Windows.

Docker Desktop for Windows - x86_64

Docker Desktop for Windows - x86_64 on the Microsoft Store

Docker Desktop for Windows - Arm (Early Access)

*For checksums, see Release notes*

## System requirements

**Tip**

**Should I use Hyper-V or WSL?**

Docker Desktop's functionality remains consistent on both WSL and Hyper-V, without a preference for either architecture. Hyper-V and WSL have their own advantages and disadvantages, depending on your specific setup and your planned use case.

WSL 2 backend, x86_64 | Hyper-V backend, x86_64

WSL 2 backend, Arm (Early Access)

- WSL version 2.1.5 or later. To check your version, see WSL: Verification and setup
- Windows 10 64-bit: Enterprise, Pro, or Education version 22H2 (build 19045).
- Windows 11 64-bit: Enterprise, Pro, or Education version 23H2 (build 22631) or higher.
- Turn on the WSL 2 feature on Windows. For detailed instructions, refer to the Microsoft documentation.
- The following hardware prerequisites are required to successfully run WSL 2 on Windows 10 or Windows 11:
  - 64-bit processor with Second Level Address Translation (SLAT)
  - 4GB system RAM
  - Enable hardware virtualization in BIOS/UEFI. For more information, see Virtualization.

For more information on setting up WSL 2 with Docker Desktop, see WSL.

> **Note**
>
> Docker only supports Docker Desktop on Windows for those versions of Windows that are still within Microsoft's servicing timeline. Docker Desktop is not supported on server versions of Windows, such as Windows Server 2019 or Windows Server 2022. For more information on how to run containers on Windows Server, see Microsoft's official documentation.

> **Important**
>
> To run Windows containers, you need Windows 10 or Windows 11 Professional or Enterprise edition. Windows Home or Education editions only allow you to run Linux containers.

Containers and images created with Docker Desktop are shared between all user accounts on machines where it is installed. This is because all Windows accounts use the same VM to build and run containers. Note that it is not possible to share containers and images between user accounts when using the Docker Desktop WSL 2 backend.

Running Docker Desktop inside a VMware ESXi or Azure VM is supported for Docker Business customers. It requires enabling nested virtualization on the hypervisor first. For more information, see Running Docker Desktop in a VM or VDI environment.

# Install Docker Desktop on Windows

## Install interactively

1. Download the installer using the download button at the top of the page, or from the release notes.

2. Double-click `Docker Desktop Installer.exe` to run the installer. By default, Docker Desktop is installed at `C:\Program Files\Docker\Docker`.

3. When prompted, ensure the **Use WSL 2 instead of Hyper-V** option on the Configuration page is selected or not depending on your choice of backend.

On systems that support only one backend, Docker Desktop automatically selects the available option.

4. Follow the instructions on the installation wizard to authorize the installer and proceed with the installation.
5. When the installation is successful, select **Close** to complete the installation process.
6. Start Docker Desktop.

If your administrator account is different to your user account, you must add the user to the **docker-users** group to access features that require higher privileges, such as creating and managing the Hyper-V VM, or using Windows containers:

1. Run **Computer Management** as an **administrator**.
2. Navigate to **Local Users and Groups** > **Groups** > **docker-users**.
3. Right-click to add the user to the group.
4. Sign out and sign back in for the changes to take effect.

## Install from the command line

After downloading `Docker Desktop Installer.exe`, run the following command in a terminal to install Docker Desktop:

```
$ "Docker Desktop Installer.exe" install
```

If you're using PowerShell you should run it as:

```
Start-Process 'Docker Desktop Installer.exe' -Wait install
```

If using the Windows Command Prompt:

```
start /w "" "Docker Desktop Installer.exe" install
```

By default, Docker Desktop is installed at `C:\Program Files\Docker\Docker`.

If your administrator account is different to your user account, you must add the user to the **docker-users** group to access features that require higher privileges, such as creating and managing the Hyper-V VM, or using Windows containers.

```
$ net localgroup docker-users <user> /add
```

See the Installer flags section to see what flags the `install` command accepts.

## Start Docker Desktop

Docker Desktop does not start automatically after installation. To start Docker Desktop:

1. Search for Docker, and select **Docker Desktop** in the search results.
2. The Docker menu (  ) displays the Docker Subscription Service Agreement.

   Here's a summary of the key points:

   - Docker Desktop is free for small businesses (fewer than 250 employees AND less than $10 million in annual revenue), personal use, education, and non-commercial open source projects.
   - Otherwise, it requires a paid subscription for professional use.
   - Paid subscriptions are also required for government entities.
   - The Docker Pro, Team, and Business subscriptions include commercial use of Docker Desktop.
3. Select **Accept** to continue. Docker Desktop starts after you accept the terms.

   Note that Docker Desktop won't run if you do not agree to the terms. You can choose to accept the terms at a later date by opening Docker Desktop.

   For more information, see Docker Desktop Subscription Service Agreement. It is recommended that you read the FAQs.

**Tip**

As an IT administrator, you can use endpoint management (MDM) software to identify the number of Docker Desktop instances and their versions within your environment.

This can provide accurate license reporting, help ensure your machines use the latest version of Docker Desktop, and enable you to enforce sign-in.

- Intune
- Jamf
- Kandji
- Kolide
- Workspace One

# Advanced system configuration and installation options

## WSL: Verification and setup

If you have chosen to use WSL, first verify that your installed version meets system requirements by running the following command in your terminal:

```
wsl --version
```

If version details do not appear, you are likely using the inbox version of WSL. This version does not support modern capabilities and must be updated.

You can update or install WSL using one of the following methods:

### Option 1: Install or update WSL via the terminal

1. Open PowerShell or Windows Command Prompt in administrator mode.
2. Run either the install or update command. You may be prompted to restart your machine. For more information, refer to Install WSL.

```
wsl --install

wsl --update
```

### Option 2: Install WSL via the MSI package

If Microsoft Store access is blocked due to security policies:

1. Go to the official WSL GitHub Releases page.

2. Download the `.msi` installer from the latest stable release (under the Assets drop-down).
3. Run the downloaded installer and follow the setup instructions.

## Installer flags

> **Note**
>
> If you're using PowerShell, you need to use the `ArgumentList` parameter before any flags. For example:

```
Start-Process 'Docker Desktop Installer.exe' -Wait -ArgumentList 'install'
```

### Installation behavior

- `--quiet` : Suppresses information output when running the installer
- `--accept-license` : Accepts the Docker Subscription Service Agreement now, rather than requiring it to be accepted when the application is first run
- `--installation-dir=<path>` : Changes the default installation location ( `C:\Program Files\Docker\Docker` )
- `--backend=<backend name>` : Selects the default backend to use for Docker Desktop, `hyper-v` , `windows` or `wsl-2` (default)
- `--always-run-service` : After installation completes, starts `com.docker.service` and sets the service startup type to Automatic. This circumvents the need for administrator privileges, which are otherwise necessary to start `com.docker.service` . `com.docker.service` is required by Windows containers and Hyper-V backend.

### Security and access control

- `--allowed-org=<org name>` : Requires the user to sign in and be part of the specified Docker Hub organization when running the application
- `--admin-settings` : Automatically creates an `admin-settings.json` file which is used by admins to control certain Docker Desktop settings on client machines within their organization. For more information, see Settings Management.
    - It must be used together with the `--allowed-org=<org name>` flag.

    - For example: `--allowed-org=<org name> --admin-settings="{'configurationFileVersion': 2, 'enhancedContainerIsolation': {'value': true, 'locked': false}}"`
- `--no-windows-containers` : Disables the Windows containers integration. This can improve security. For more information, see Windows containers.

### Proxy configuration

- `--proxy-http-mode=<mode>` : Sets the HTTP Proxy mode, `system` (default) or `manual`
- `--override-proxy-http=<URL>` : Sets the URL of the HTTP proxy that must be used for outgoing HTTP requests, requires `--proxy-http-mode` to be `manual`
- `--override-proxy-https=<URL>` : Sets the URL of the HTTP proxy that must be used for outgoing HTTPS requests, requires `--proxy-http-mode` to be `manual`
- `--override-proxy-exclude=<hosts/domains>` : Bypasses proxy settings for the hosts and domains. Uses a comma-separated list.
- `--proxy-enable-kerberosntlm` : Enables Kerberos and NTLM proxy authentication. If you are enabling this, ensure your proxy server is properly configured for Kerberos/NTLM authentication. Available with Docker Desktop 4.32 and later.
- `--override-proxy-pac=<PAC file URL>` : Sets the PAC file URL. This setting takes effect only when using `manual` proxy mode.
- `--override-proxy-embedded-pac=<PAC script>` : Specifies an embedded PAC (Proxy Auto-Config) script. This setting takes effect only when using `manual` proxy mode and has precedence over the `--override-proxy-pac` flag.

### Example of specifying PAC file

```
"Docker Desktop Installer.exe" install --proxy-http-mode="manual" --override-prc
```

### Example of specifying PAC script

```
"Docker Desktop Installer.exe" install --proxy-http-mode="manual" --override-prc
```

### Data root and disk location

- `--hyper-v-default-data-root=<path>` : Specifies the default location for the Hyper-V VM disk.

- `--windows-containers-default-data-root=<path>` : Specifies the default location for the Windows containers.
- `--wsl-default-data-root=<path>` : Specifies the default location for the WSL distribution disk.

## Administrator privileges

Installing Docker Desktop requires administrator privileges. However, once installed, it can be used without administrative access. Some actions, though, still need elevated permissions. See Understand permission requirements for Windows for more detail.

See the FAQs on how to install and run Docker Desktop without needing administrator privileges.

If you're an IT admin and your users do not have administrator rights and plan to perform operations that require elevated privileges, be sure to install Docker Desktop using the `--always-run-service` installer flag. This ensures those actions can still be executed without prompting for User Account Control (UAC) elevation. See Installer Flags for more detail.

## Windows containers

From the Docker Desktop menu, you can toggle which daemon (Linux or Windows) the Docker CLI talks to. Select **Switch to Windows containers** to use Windows containers, or select **Switch to Linux containers** to use Linux containers (the default).

For more information on Windows containers, refer to the following documentation:

- Microsoft documentation on Windows containers.
- Build and Run Your First Windows Server Container (Blog Post) gives a quick tour of how to build and run native Docker Windows containers on Windows 10 and Windows Server 2016 evaluation releases.
- Getting Started with Windows Containers (Lab) shows you how to use the MusicStore application with Windows containers. The MusicStore is a standard .NET application and, forked here to use containers, is a good example of a multi-container application.
- To understand how to connect to Windows containers from the local host, see I want to connect to a container from Windows

> **Note**
>
> When you switch to Windows containers, **Settings** only shows those tabs that are active and apply to your Windows containers.

If you set proxies or daemon configuration in Windows containers mode, these apply only on Windows containers. If you switch back to Linux containers, proxies and daemon configurations return to what you had set for Linux containers. Your Windows container settings are retained and become available again when you switch back.

## Where to go next

- Explore Docker's subscriptions to see what Docker can offer you.
- Get started with Docker.
- Explore Docker Desktop and all its features.
- Troubleshooting describes common problems, workarounds, and how to get support.
- FAQs provide answers to frequently asked questions.
- Release notes lists component updates, new features, and improvements associated with Docker Desktop releases.
- Back up and restore data provides instructions on backing up and restoring data related to Docker.