

Install Docker Desktop

Step-by-step installation of Docker Desktop on
Windows

Marianne Gillfillan, Senior DBA & Cloud Architect

October 2022

Strata Information Group
3935 Harney St., Suite 203
San Diego, CA 92110



© 2022. All rights reserved. Reports, deliverables, or training materials that SIG provides to the organization are only for the internal use of the organization. They may not be distributed to any other person or party, for any purpose, without prior written consent from SIG

Contents

1 Setting up Docker Desktop on Windows 4

1.1 Install Windows Subsystem for Linux 2 (WSL2) 5

1.2 Download Docker Desktop for Windows 6

1.3 Install Docker Desktop for Windows 7

1.3.1 Licensing 7

1.3.2 Rate Limiting 8

1.4 Testing the Docker Desktop Installation 9

1. Setting up Docker Desktop on Windows

Before installing Docker Desktop on Windows, we will need to install the Windows Subsystem for Linux 2. This will create a Ubuntu Linux environment from which we will interact with Docker once it is installed. Before proceeding, make sure you have Administrative rights on the machine where these installations will occur. Windows 11 was used as the basis for these illustrations, however, Windows 10 can also be used. However, Windows 10 has been known to cause issues with the need to run/install Hyper-V which can cause conflicts with other virtualization software like VirtualBox.

1.1. Install Windows Subsystem for Linux 2 (WSL2)

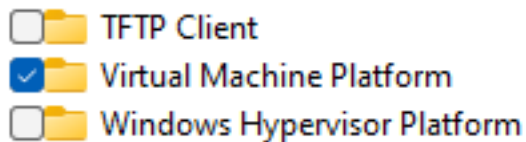
To install WSL2 on Windows 10 or Windows 11, please review the prerequisites at:

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

Please note, you do not need to run Hyper-V in order to run Docker Desktop on Windows. If you're going down the path of turning on Hyper-V stop. It's not needed for this exercise.

However, you still do need to have the Virtual Machine Platform enabled. Most Windows 10 and Windows 11 pre-installations come with this feature already turned on. Before going too much further, verify this feature is enabled.

1. Select **Start**, enter **Windows features**, and select **Turn Windows features on or off** from the list of results.
2. In the Windows Features window that just opened, find **Virtual Machine Platform**. If the box is already checked, great, your job is done and you can skip the next part of the installation, otherwise, select this feature to be enabled.



3. Select **OK**. Restart your PC if necessary.

Review the installation procedure at:

<https://docs.microsoft.com/en-us/windows/wsl/install>

Open Windows Powershell as administrator and enter the following command to install wsl (yes it is that simple.)

```
wsl --install
```

Reboot the machine. You should notice that Ubuntu is installing during the reboot process. Once the reboot is completed, a Ubuntu window will open and you will be asked to enter a username and password for your new Linux environment. Try to keep track of these credentials as they will be needed again.

With WSL2 installed, you can now install Docker Desktop for Windows.

1.2. Download Docker Desktop for Windows

Download the Docker installer executable at:

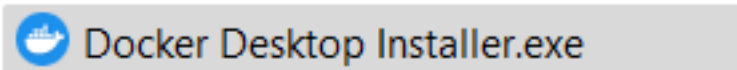
<https://desktop.docker.com/win/main/amd64/Docker%20Desktop%20Installer.exe>

The file will attempt to download to your Downloads directory. Feel free to redirect that as needed, just remember where you put it.

1.3. Install Docker Desktop for Windows

Once the download completes, double click the executable and follow the prompts. There are not too many, but the most important one to note is the Configuration screen making sure that you want to install the version for WSL2 instead of Hyper-V. If WSL2 is installed properly, the installer should default to this option without issue.

Name



- ***Since WSL 2 is installed, verify that the box for “Use WSL 2 instead of Hyper-V” is automatically checked.***

Configuration

- ☒ Use WSL 2 instead of Hyper-V (recommended)
- ☒ Add shortcut to desktop

After answering all the prompts, restart your machine.

Once the machine is restarted, startup Docker Desktop. As Docker Desktop starts up, it will present a licensing window.

1.3.1. Licensing

If you're using Docker Desktop for personal use, Docker Desktop remains free to use. For the purposes of this workshop, let's consider this personal study.

Please review the FAQs for more details related to the licensing of Docker Desktop:


<https://www.docker.com/pricing/faq/>.

1.3.1.1. Getting around the licensing issue If you really want to get around the licensing restrictions for Docker Desktop, you can manually install the Docker CLI components into your Ubuntu environment once WSL2 is configured, then install Portainer as a GUI view into your containers. Here's a reference on how to get started:

<https://betterprogramming.pub/how-to-install-docker-without-docker-desktop-on-windows-a2bbb65638a1>

1.3.2. Rate Limiting

Docker Hub introduced rate limiting on image pulls from their repositories. It is possible that you may encounter messages like **"You have reached your pull limit."** First, make sure you are logged in

to Docker Hub via Docker Desktop. Open a Linux command window , check your current pull rate statistics:

```
sudo apt-get update
sudo apt-get install jq
TOKEN=$(curl
  ↪ "https://auth.docker.io/token?service=registry.docker.io&scope=repository:ratelimitpr
  ↪ | jq -r .token)
curl --head -H "Authorization: Bearer $TOKEN"
  ↪ https://registry-1.docker.io/v2/ratelimitpreview/test/manifests/latest
```

This will produce the header information from the pull request. The lines of interest will be:

```
ratelimit-limit: 100;w=21600
ratelimit-remaining: 100;w=21600
```

This means that the rate limit is 100 pulls per 21600 seconds (6 hours) and there are 100 pulls remaining. We will want to keep an eye on this during the course of this workshop.

1.4. Testing the Docker Desktop Installation

From a command line check the docker version

```
docker --version
```

```
Docker version 20.10.17, build 100c701
```

Let's run a very simple container

```
docker run hello-world
```

```
Unable to find image 'hello-world:latest' locally
```

```
latest: Pulling from library/hello-world
```

```
2db29710123e: Pull complete
```

```
Digest:
```

```
⇒ sha256:62af9efd515a25f84961b70f973a798d2eca956b1b2b026d0a4a63a3b0b6a3f2
```

```
Status: Downloaded newer image for hello-world:latest
```

```
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/
```

Ok, so it looks like we have a good installation of Docker Desktop. We can continue with our lessons.

