

Installing DDEV on Windows

This guide provides a comprehensive, step-by-step process for installing DDEV on a Windows machine.

Installing DDEV on Windows	1
Prerequisites	1
Install Docker Desktop	1
Install Windows Subsystem for Linux 2 (WSL2)	2
Install DDEV	2
Configure DDEV	2
Verify Installation	3
Troubleshooting	3

Prerequisites

Ensure your Windows system meets these requirements:

- Windows 10 version 2004 or higher (Build 19041 or higher) or Windows 11
- 64-bit processor with Second Level Address Translation (SLAT)
- 4GB system RAM or more
- BIOS-level hardware virtualization support must be enabled in the BIOS settings

Install Docker Desktop

1. Download Docker Desktop for Windows from the [official Docker website](#).
2. Run the installer and follow the prompts.
3. During installation, ensure the Use WSL 2 instead of Hyper-V option is selected.
4. After installation, restart your computer.

5. Start Docker Desktop and wait for it to finish loading.

Install Windows Subsystem for Linux 2 (WSL2)

1. Open PowerShell as Administrator.
2. Run the following command to install WSL2:

```
wsl --install
```

3. Restart your computer when prompted.
4. After restart, open the Microsoft Store and install Ubuntu (or your preferred Linux distribution).
5. Launch Ubuntu and set up your username and password when prompted.

Install DDEV

1. Open your WSL2 terminal (Ubuntu).
2. Run the following command to download and install DDEV:

```
curl -fsSL  
https://raw.githubusercontent.com/drud/ddev/master/scripts/install\_ddev.sh |  
bash
```

This script will download and install the latest version of DDEV.

Configure DDEV

1. Set up global configuration (optional but recommended):

```
ddev config global
```

2. Follow the prompts to configure global settings.

Verify Installation

1. Check the installed version of DDEV:

```
ddev --version
```

2. You should see output indicating the installed DDEV version.

Troubleshooting

If you encounter any issues during installation or use of DDEV, try these steps:

1. Docker Desktop Issues:
 - Ensure Docker Desktop is running.
 - Check if **Virtualization** is enabled in your BIOS settings.
 - Try restarting Docker Desktop.
2. WSL2 Issues:
 - Make sure you're using WSL2, not WSL1. Check with: `wsl -l -v`
 - If needed, upgrade to WSL2: `wsl --set-version Ubuntu 2`
3. DDEV Command Not Found:
 - Ensure the DDEV binary is in your PATH.
 - Try restarting your WSL2 terminal.
4. Performance Issues:
 - Ensure your project files are stored in the WSL2 filesystem, not on Windows drives.
5. Network Issues:
 - Check your firewall settings to ensure DDEV can access the network.
6. General Troubleshooting:
 - Run `ddev poweroff` and then `ddev start` to restart all DDEV projects.
 - Check DDEV logs with `ddev logs` for any error messages.