

# Windows

- [Usefull stuff](#)
- [WSL Windows subsystem for linux](#)
- [My Basic Webdev WSL setup](#)

# Usefull stuff

## open startup folder

alias: autorun, autostart

- press Windows + r
- enter `shell:startup`
- hit Enter

## edit hosts file

- press Windows + r
- enter `notepad C:\Windows\System32\drivers\etc\hosts`
- hit Ctrl + Shift + Enter

# WSL Windows subsystem for linux

## install

[manual](#)

```
wsl --install Ubuntu-24.04 --name Number2
```

## backup and restore

[xda developers](#)

commands from link:

```
# backup
wsl -l
wsl --export (distribution) (file path)

# remove
wsl -v
#-> uninstall dist form startmenu
wsl --unregister <distribution>

# restore
wsl --import (distribution) (install location) (file path)
wsl --distribution (distribution)
```

## move / clone distribution

not tested !!!

```
wsl --export Debian C:\temp\Debian.tar
wsl --import MovedDebian C:\WSL\Debian C:\temp\Debian.tar --version 2
wsl --distribution MovedDebian
```

# settings

## Mirrored Mode WSL2 Networking

from [superuser.com](https://superuser.com)

Note: Requires Windows 11 23H2 or later

In Mirrored mode, localhost should "just work". To enable:

- Add the following to your `<windows_user_profile>/.wslconfig`:

```
[wsl2]
networkingMode=mirrored
#autoProxy=false # only if necessary ?
```

- Exit your WSL distribution
- Run `wsl --shutdown` from PowerShell then restart WSL

You should then be able to access services running in Windows via localhost.

Note that when running in Mirrored mode, mDNS will not work.

## hosts configs:

edit `C:\Windows\System32\drivers\etc\hosts` in admin mode add:

```
127.0.0.1 shop.local
```

## non mirrored wsl2 mode

- simple check that the `./wslconfig` file is empty or at least `networkingMode=mirrored` is not set.
- check the ip address: `wsl hostname -I`

TODO

# My Basic Webdev WSL setup

## install

```
# install wsl
wsl --install

# list available distros
wsl -l -o

# install ubuntu
wsl --install Ubuntu-24.04

# run specific distro and user
wsl --distribution <Distribution Name> --user <User Name>
```

## first commands

## install must haves

```
# update and installs
sudo apt update && sudo apt upgrade
sudo apt install git curl net-tools dnsutils
```

## git

```
git config --global user.name "Your Name Here"
git config --global user.email your@email.example
```

## ssh

```
## generate pub/private keys
ssh-keygen -C "{short description}"
## or paste keys and check permissions
sudo chmod 400 ~/.ssh/id_ed25516
## add configs
nano ~/.ssh/config
```

## docker

```
curl -sSL https://get.docker.com/ | sh
## put user into docker group
sudo usermod -aG docker $USER
newgrp docker # to add docker in your current shell session
## docker desktop not necessary -> not free
docker ps
```

## zsh

```
if -f ~/.zshrc; then
  echo "Installing zsh with oh my zsh and powerlevel10kTheme"

  sudo apt update && sudo apt install curl git zsh
  sh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)" "" --unattended
  git clone --depth=1 https://github.com/romkatv/powerlevel10k.git ${ZSH_CUSTOM:-$HOME/.oh-my-
zsh/custom}/themes/powerlevel10k
  sed -i '/ZSH_THEME="/c\ZSH_THEME="powerlevel10k/powerlevel10k"' ~/.zshrc
fi
```

## java

```
sudo apt-get update
sudo apt-get install curl zip unzip
curl -s "https://get.sdkman.io" | bash
```

```
# source "$HOME/.sdkman/bin/sdkman-init.sh" # or reopen the terminal

sdk list java
sdk install java 25.0.3-tem # latest lts version of eclipse, best alternative would be amzn
corretto
sdk default java 25.0.3-tem
```

# node

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.4/install.sh | bash
# reopen the terminal
nvm install --lts
nvm use --lts
```

# pnpm

```
sudo apt-get install -y libatomic1
curl -fsSL https://get.pnpm.io/install.sh | sh -
# reopen the terminal
```