

Cheat sheet

Red Hat Enterprise Linux in Microsoft Windows Subsystem for Linux (WSL)

[Red Hat Enterprise Linux](#) (RHEL) is the world's leading enterprise Linux platform. It evolves to bring flexibility and reliability to new environments, from the public cloud to the edge.

Microsoft's Windows Subsystem for Linux (WSL) is a validated software platform for Red Hat Enterprise Linux, intended for development use. It eliminates the need for a traditional virtual machine, eases operations by allowing both Windows and RHEL to run simultaneously, and can save time when building Linux applications on Windows for production deployment on RHEL.

You can download Red Hat Enterprise Linux as a WSL image during installation or build a customized image using Red Hat image builder and install it in WSL. The latest versions of RHEL 8, RHEL 9, and RHEL 10 are available for WSL.

WSL basic management (PowerShell/CMD)

Run the following commands in PowerShell or Windows Command Prompt (CMD) on your Windows host.

Task	Command
Install WSL	<code>wsl --install -no-distribution</code>
Import the specified RHEL image as a new distribution	<code>wsl --import RHEL C:\WSL\RHEL\Downloads\rhel_image.tar.gz</code>
List installed distributions with details (WSL version, state)	<code>wsl -l -v</code>
Set RHEL as the specified distribution as the default for <code>wsl.exe</code> commands	<code>wsl -s RHEL</code>
Launch into RHEL	<code>wsl -d RHEL</code>

Task	Command
Immediately terminate all running distributions and the WSL 2 lightweight utility virtual machine	<code>wsl --shutdown</code>
Terminate just your RHEL instance of WSL	<code>wsl -t RHEL<version></code>
Unregister (delete) a distribution CAUTION: All data will be lost.	<code>wsl -unregister RHEL</code>
Update the WSL kernel	<code>wsl --update</code>
Get general information about your WSL configuration, such as default distribution, default version, and kernel version	<code>wsl --status</code>

Essential Linux commands

The following are common commands RHEL users are likely to use every day.

Task	Command	Example
Change directory	<code>cd</code>	<code>cd ~/Documents</code>
List files and directories	<code>ls</code>	<code>ls</code>
Get current working directory	<code>pwd</code>	<code>pwd</code>
Get URL	<code>curl</code>	<code>curl http://example.com</code>
View file contents	<code>cat</code>	<code>cat ~/Documents/Readme.md</code>

Task	Command	Example
Find a file by part of the file name	<code>find</code>	<code>find ~ -name "* .txt"</code>
Unzip a file	<code>unzip</code>	<code>unzip example.zip</code>
Untar a file	<code>tar</code>	<code>tar --extract --file example.tar.gz</code>
Rename or move a file	<code>mv</code>	<code>mv notes.txt Readme.md</code>
Log in as tux to a remote system	<code>ssh</code>	<code>ssh tux@10.0.1.172</code>
Elevate permissions	<code>sudo</code>	<code>sudo mv my.config /etc/</code>

RHEL package management

Task	Command
Update all installed packages to their latest versions	<code>sudo dnf update</code>
Similar to update, but can remove obsolete packages	<code>sudo dnf upgrade</code>
Install a new package (e.g., <code>sudo dnf install httpd</code>)	<code>sudo dnf install <package_name></code>
Remove an installed package	<code>sudo dnf remove <package_name></code>
Search for packages matching the keyword	<code>sudo dnf search <keyword></code>
Display detailed information about a package	<code>sudo dnf info <package_name></code>
List installed packages	<code>sudo dnf list installed</code>

Task	Command
Show a history of DNF transactions	<code>sudo dnf history</code>
Check for available updates without installing them	<code>sudo dnf check-update</code>
Install a package group (e.g., <code>sudo dnf groupinstall "Development Tools"</code>)	<code>sudo dnf groupinstall "<group_name>"</code>

RHEL service management

WSL2 enables systemd support by default to allow you to control which services are started automatically when a WSL instance is started. To enable systemd in RHEL on WSL if it's not already, you might need to edit `/etc/wsl.conf` and add:

```
[boot]
systemd=true
```

Then restart your WSL instance with `wsl --shutdown`.

WSL performance tuning and configuration (.wslconfig)

Create or edit the `.wslconfig` file in your Windows user profile directory (`%UserProfile%\.\wslconfig`, e.g., `C:\Users\<YourUserName>\.wslconfig`). These settings apply globally to all WSL 2 distributions. You need to run `wsl --shutdown` and restart your WSL instances for changes to take effect.

Example `.wslconfig` content:

```
[wsl2]
memory=4GB          # Limit RAM to 4GB (e.g., memory=8GB, memory=16GB)
processors=2         # Limit to 2 CPU cores (e.g., processors=4)
# swap=2GB          # Optional: Configure swap space size
# localhostForwarding=true # Default, forwards WSL 2 localhost to Windows
localhost
# pageReporting=true      # Default, allows WSL to release unused memory
back to Windows
```

```
# guiApplications=true      # Default, enables WSLg for GUI apps
# nestedVirtualization=true # Enable nested virtualization (e.g., for running
Podman in WSL)
# vmIdleTimeout=60000       # Time in ms before an idle VM is shut down.
(Default is 60000ms or 1 minute)
# Experimental features (use with caution, check Microsoft docs for latest)
# [experimental]
# autoMemoryReclaim=gradual # Or `dropcache`. Helps in reclaiming memory.
# sparseVhd=true           # Enables sparse VHD to reclaim disk space
automatically.
```

Key .wslconfig options

- `memory=<size>`: Maximum memory WSL 2 can use (e.g., 4 GB, 8192 MB).
- `processors=<number>`: Maximum number of logical processors for WSL 2.
- `swap=<size>`: Amount of swap space. `0` for no swap file.
- `pageReporting=<true|false>`: Set to `true` to allow WSL to release cached memory back to Windows when it's no longer needed.
- `autoMemoryReclaim=<disabled|gradual|dropcache>` (Experimental) Controls how WSL reclaims unused memory. `gradual` is often preferred.
- `sparseVhd=<true|false>`: (Experimental) If set to `true`, WSL virtual disks will be automatically set to `sparse`, which can help reclaim disk space more effectively. Requires `wsl --manage <distro> --set-sparse <true|false>` per distro after enabling.

To check current resource usage from within RHEL:

- Memory: `free -h`
- CPU: `nproc` or `htop`
- Disk: `df -h`

WSL networking

From Windows:

Task	Command (PowerShell/CMD)
Show Windows IP configuration, including the vEthernet (WSL) adapter	<code>ipconfig</code> (Microsoft Windows)
Get IP details for the WSL virtual adapter	<code>Get-NetIPAddress -InterfaceAlias "vEthernet (WSL)"</code> (PowerShell)

From RHEL:

Task	Command (Linux/Bash)
Show IP addresses for interfaces within WSL (e.g., <code>eth0</code>)	<code>ip addr</code>
Show the IP address(es) of the WSL instance	<code>hostname -I</code>
Show DNS resolver configuration. WSL typically mirrors Windows DNS.	<code>cat /etc/resolv.conf</code>
Ping the Windows host machine from WSL. You can find Windows host IP from <code>ipconfig</code> (Windows).	<code>ping <windows_host_ip></code>
Open Windows File Explorer in the current RHEL directory	<code>explorer.exe . (from RHEL)</code>

Accessing WSL services from Windows

Use `localhost:<port>` (e.g., `localhost:8080`) if `localhostForwarding` is enabled (default).

Accessing Windows services from WSL

Use the IP address of the Windows host. You can generally find this by looking at the `/etc/resolv.conf` file in WSL; the nameserver listed is usually the Windows host.

```
cat /etc/resolv.conf | grep nameserver | awk '{print $2}'
```

File system interaction

Access Windows files from RHEL

Windows drives are mounted under `/mnt/`. For example:

```
cd /mnt/c/Users/<YourUserName>/Documents
```

Access RHEL files from Windows

Open File Explorer and type `\wsl$` or `\wsl.localhost\<DistroName>\` in the address bar. For example:

```
\wsl.localhost\RHEL\
```

The root of your RHEL file system is accessible here.

Best practice for performance

Store project files inside the Linux file system (e.g., `~/projects`) for significantly faster I/O performance compared to accessing files across the WSL/Windows boundary (e.g., in `/mnt/c/`).

Troubleshooting commands

Command (PowerShell/CMD or RHEL)	Description	Context
<code>dmesg</code>	Kernel ring buffer messages. Useful for diagnosing hardware or driver issues within WSL.	RHEL
<code>journalctl -xe</code>	Systemd journal logs. Useful for diagnosing service issues if systemd is enabled.	RHEL
<code>wsl.exe --debug-shell</code>	(Advanced) Opens a diagnostic shell for the WSL virtual machine.	Windows (PowerShell)