DevOps Days Geneva

Speaker 2022

Olivier Crozier

Engineer @Kudelski Group

Setup WSL2 with care

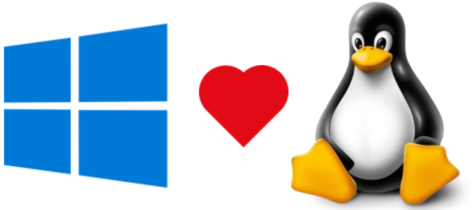
NAGRA
KUDELSKI GROUP

IN OTHER WORDS

Turn your Windows 10 laptop

as a great Linux dev platform

with WSL2



WE HAVE 1,5 HOUR TOGETHER

- 45’ pres, 45’ demo + Q/A

Introduction (4’)

Let’s start install and configure (19’)

Organize your environments (14’)

Advanced use (7’)

Conclusion (1’)

LET’S CLARIFY WHO AM I

- I am not affiliated with Microsoft
- I am not selling anything
- I am not a speaker or technical writer (on Internet)

- I am used to share knowledge within my company
- I am used to give presentations, training, guidance
- I am a developer who needs Linux on my Windows

TODAY I'M WORKING AT

NAGRA
KUDELSKI

Nagra Audio

1951

NAGRA
KUDELSKI GROUP

Multiscreen
& Digital TV

KUDELSKI
SECURITY

Cybersecurity

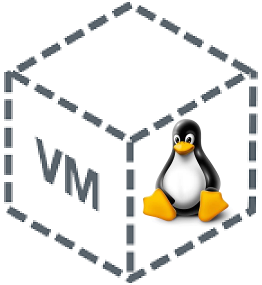
SKIDATA®
KUDELSKI GROUP

Public Access

KUDELSKI
IoTHINGS

Internet Of Things

WHY TALKING ABOUT WSL2 ?



- Why Linux ? Why Linux VM ?

  Most backends rely on Linux infra, especially if using containers → Be as close on Windows dev env

- Why WSL ?

  To provide a better developer experience on Windows

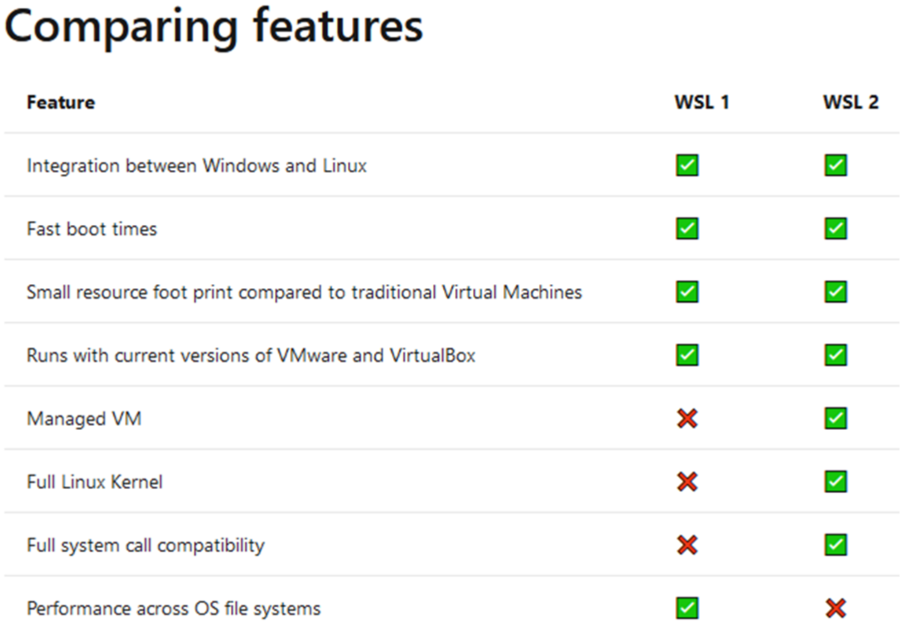
- Why WSL 2 ?

  Install any Linux app like docker

IN A NUTSHELL

WSL2 = Full Linux + seamless integration

- Install any Linux app
  Docker, Kubernetes…
- File sharing both directions
  +Linux perms to Win files
- Run Win/Linux executables
  On both systems
- Localhost forwarding
- Better use of resources

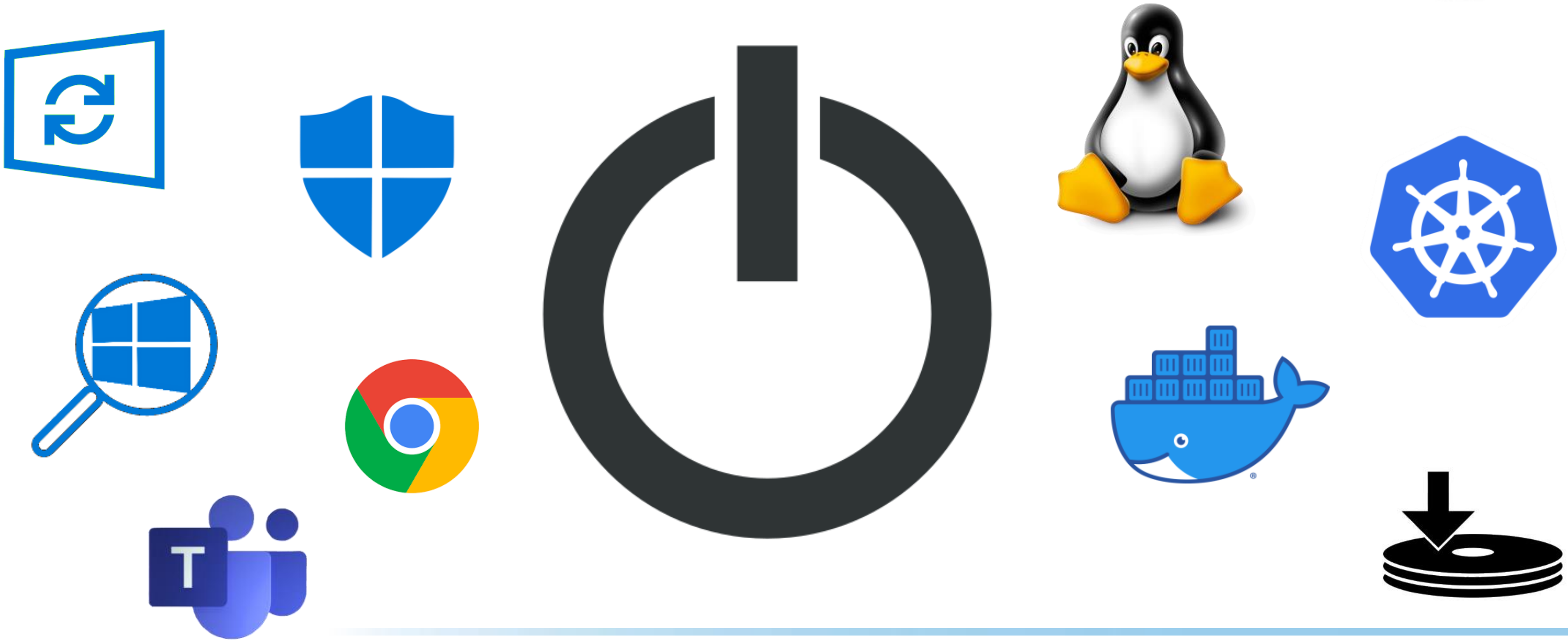


Comparing features

Feature	WSL 1	WSL 2
Integration between Windows and Linux	✓	✓
Fast boot times	✓	✓
Small resource foot print compared to traditional Virtual Machines	✓	✓
Runs with current versions of VMware and VirtualBox	✓	✓
Managed VM	✗	✓
Full Linux Kernel	✗	✓
Full system call compatibility	✗	✓
Performance across OS file systems	✓	✗

MANAGE YOUR LAPTOP

like any cloud resource



LET’S START

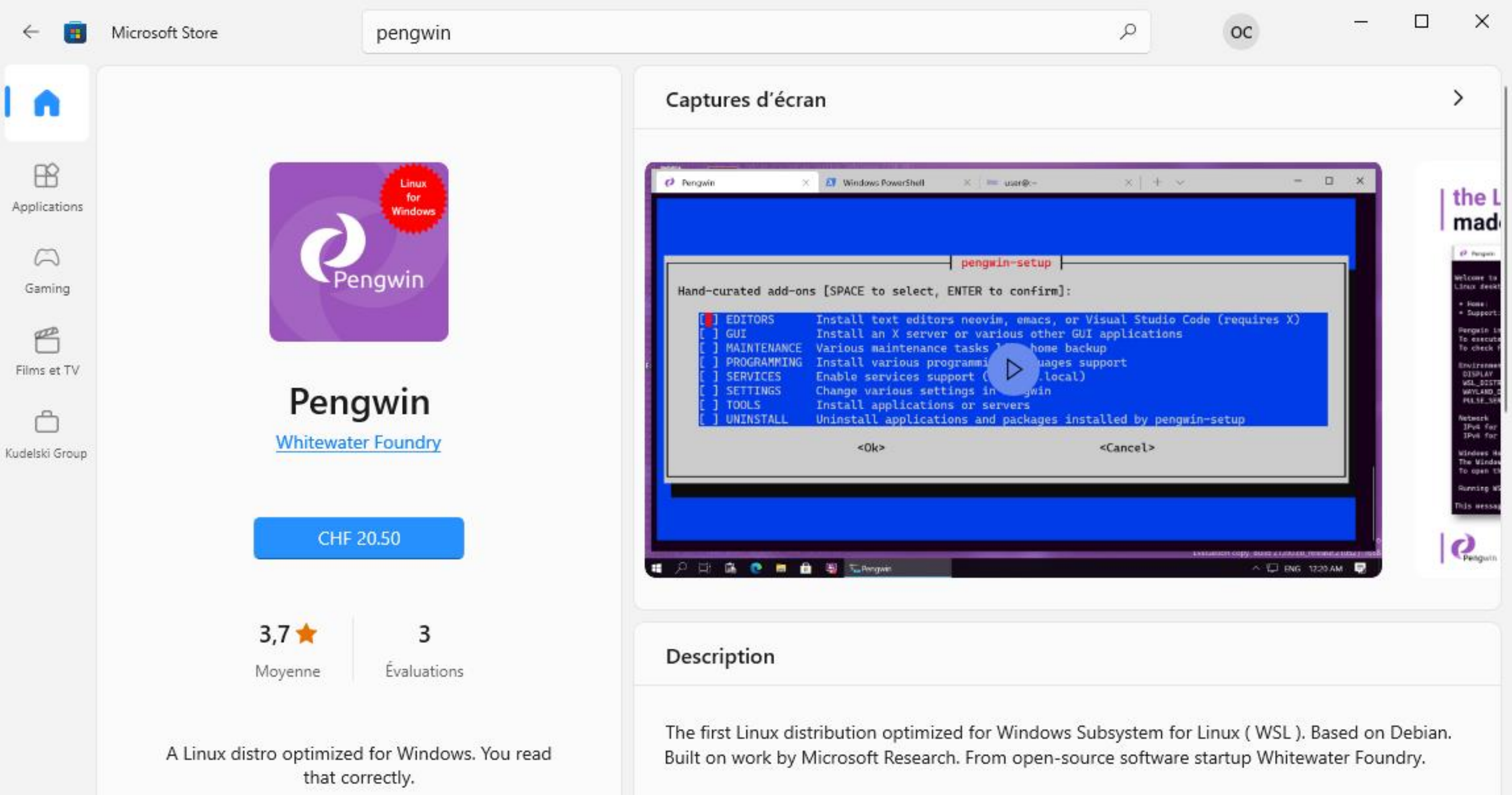
wsl commands

```
PS> wsl --set-default-version 2        # Make sure WSL generation 2 is the default
PS> wsl --list [--verbose|--online]    # List installed or online available distributions
PS> wsl --install -d Ubuntu-20.04      # Install this other distribution
PS> wsl -d|--distribution Ubuntu-20.04 # Launch this specific distribution
PS> wsl -t|--terminate Ubuntu-20.04    # Shutdown this specific distribution
PS> wsl --shutdown                     # Shutdown all distributions
PS> wsl --unregister Ubuntu-20.04      # Uninstall this specific distribution
PS> wsl --help                         # More options
```

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

https://github.com/microsoft/WSL

CHOOSE YOUR DISTRIBUTION



Pengwin

Whitewater Foundry

CHF 20.50

3,7 ★ Moyenne

3 Évaluations

A Linux distro optimized for Windows. You read that correctly.

```
                              pengwin-setup
Hand-curated add-ons [SPACE to select, ENTER to confirm]:

  [ ] EDITORS       Install text editors neovim, emacs, or Visual Studio Code (requires X)
  [ ] GUI           Install an X server or various other GUI applications
  [ ] MAINTENANCE   Various maintenance tasks l    home backup
  [ ] PROGRAMMING   Install various programmi      uages support
  [ ] SERVICES      Enable services support (      .local)
  [ ] SETTINGS      Change various settings in     gwin
  [ ] TOOLS         Install applications or servers
  [ ] UNINSTALL     Uninstall applications and packages installed by pengwin-setup

                <Ok>                                   <Cancel>
```

Description

The first Linux distribution optimized for Windows Subsystem for Linux (WSL). Based on Debian. Built on work by Microsoft Research. From open-source software startup Whitewater Foundry.

INSTALL WSL2

- Prerequisites

  Windows 10 version 1903+

  WSL2 Linux kernel installed, Virtual Machine feature enabled (Hyper-V…)

  PowerShell enabled and unrestricted

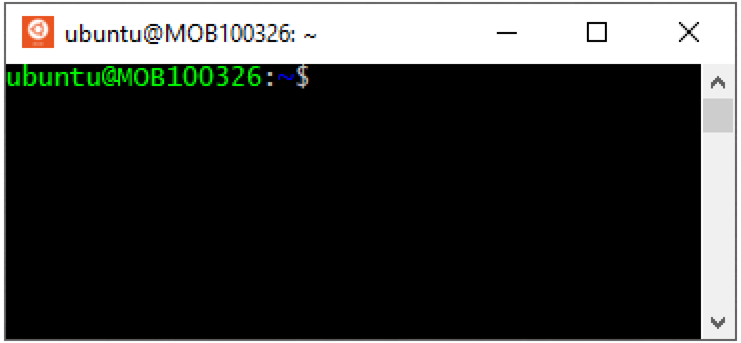
- Installation

  ```
  wsl --set-default-version 2
  wsl --install --distribution Ubuntu-20.04
  ```

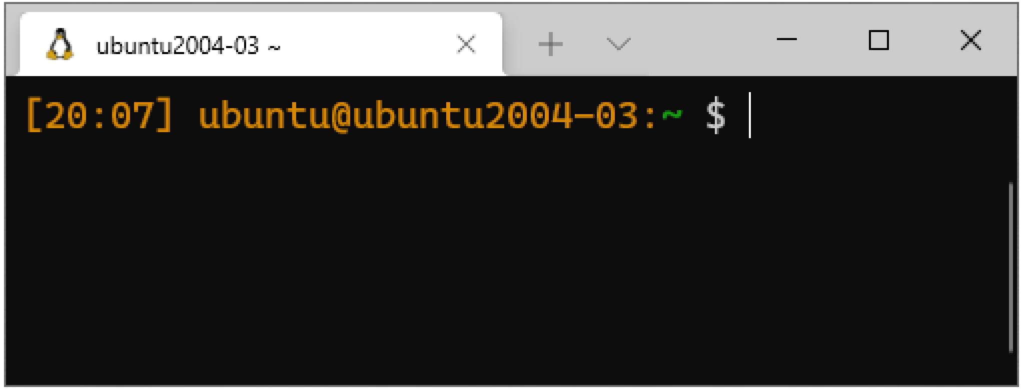
  Reboot your Windows 10

  UNIX username: ubuntu

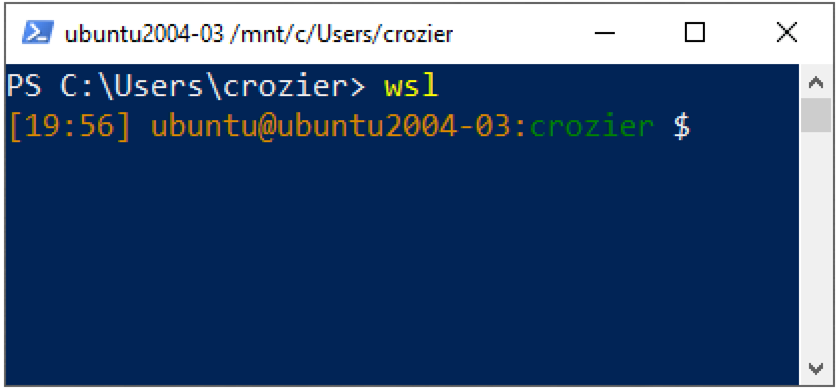
START WSL2 – WHICH TERMINAL ?



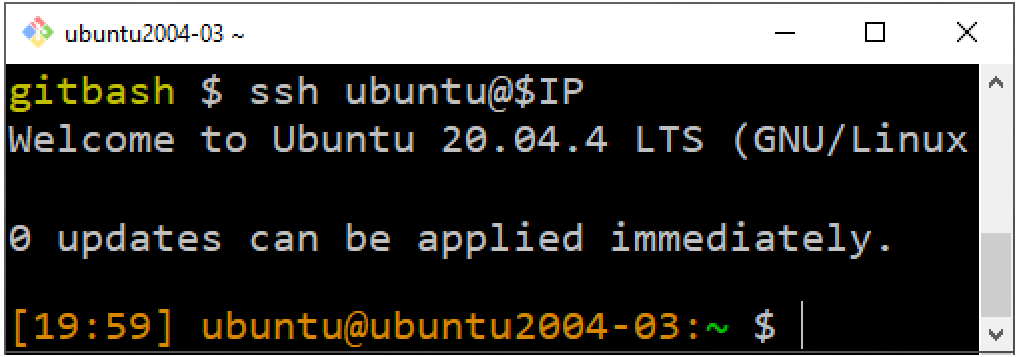
```
ubuntu@MOB100326:~$
```



```
[20:07] ubuntu@ubuntu2004-03:~ $
```



```
PS C:\Users\crozier> wsl
[19:56] ubuntu@ubuntu2004-03:crozier $
```



```
gitbash $ ssh ubuntu@$IP
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux

0 updates can be applied immediately.

[19:59] ubuntu@ubuntu2004-03:~ $
```

CONFIGURE WSL2

↓ %USERPROFILE%\.wslconfig

```
# Settings apply across all Linux distros running on WSL 2
[wsl2]

# Limits VM memory to use no more than 4 GB
memory=4GB

# Sets the VM to use two virtual processors
processors=2

# Specify whether ports bound to wildcard or localhost in the WSL 2 VM
# should be connectable from the host via localhost:port
localhostforwarding=true
```

/etc/wsl.conf →

```
# Automatically mount Windows drive when the distribution is launched
[automount]
options = "metadata,umask=022,fmask=11"

# Network host settings that enable the DNS server used by WSL 2.
[network]
hostname = ubuntu2004
generateHosts = true
generateResolvConf = false

# Set whether WSL supports interop process
# like launching Windows apps and adding path variables.
[interop]
enabled = true
appendWindowsPath = false

# Set the user when launching a distribution with WSL.
[user]
default = ubuntu

# Set a command to run when a new WSL instance launches.
[boot]
command = service sshd start
```

FIRST INSTALL

```
sudo visudo # NOPASSWD:

sudo rm /etc/resolv.conf # To remove symlink
sudo vi /etc/resolv.conf # nameserver 1.1.1.1

# Update and upgrade all packages before install new
sudo apt update
sudo apt upgrade -y
sudo apt autoremove -y
sudo apt install socat dos2unix -y

export SHELL=/bin/bash     # Or /bin/zsh

export PROMPT_COMMAND=... # Title (ubuntu2004 or root@ubuntu2004)
export PS1=...            # Prompt (orange/red, with/without __git_ps1)

vi ~/.vimrc               # syntax on, colorscheme desert
```

REBOOT ?

- «sudo reboot» does nothing

  → because there is no systemd

- Only Windows manages WSL2 state

```
wsl --terminate Ubuntu-20.04
wsl --shutdown # Terminate all WSL2 distribs +the lightweight utility VM
```

START SERVICES

- Default

  sudo service docker start # stop, restart, status

- Enable systemd at PID1

  one-script-wsl2-systemd

  systemd-genie in a subshell

  …

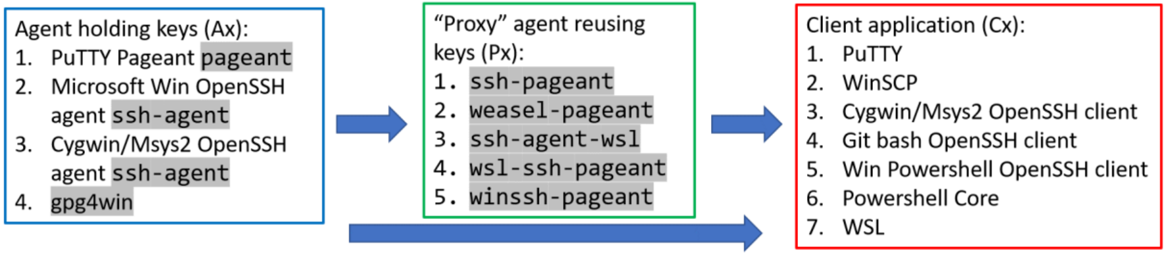
Start services ✓
Reboot ✗
Install systemd base softwares ⚠

- Should systemd be used?

  https://wsl.dev/wsl2init/

CONNECT PAGEANT FOR SSH

https://stackoverflow.com/questions/62937020



Agent holding keys (Ax):

1. PuTTY Pageant pageant
2. Microsoft Win OpenSSH agent ssh-agent
3. Cygwin/Msys2 OpenSSH agent ssh-agent
4. gpg4win

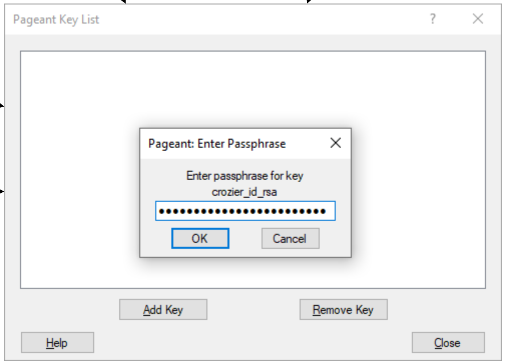
"Proxy" agent reusing keys (Px):

1. ssh-pageant
2. weasel-pageant
3. ssh-agent-wsl
4. wsl-ssh-pageant
5. winssh-pageant

Client application (Cx):

1. PuTTY
2. WinSCP
3. Cygwin/Msys2 OpenSSH client
4. Git bash OpenSSH client
5. Win Powershell OpenSSH client
6. Powershell Core
7. WSL

```
# GitBash: $HOME/.bash_profile
eval $(/usr/bin/ssh-pageant -r -a "/tmp/.ssh-pageant-$USERNAME")

# PowerShell:
# Create an environment variable SSH_AUTH_SOCK
# whose value is the output of following command:
(Get-ChildItem \\.\pipe\).FullName | findstr pageant

# List the keys loaded in pageant
ssh-add -l
```

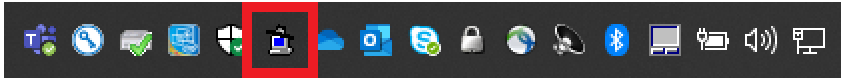


Pageant Key List

Pageant: Enter Passphrase

Enter passphrase for key crozier_id_rsa

••••••••••••••••••••••••



CONNECT PAGEANT FOR SSH IN WSL2

https://github.com/BlackReloaded/wsl2-ssh-pageant
+issues/23#issuecomment-882068132
+releases/download/v1.3.0/wsl2-ssh-pageant.exe

```
sudo apt install -y socat dos2unix
mkdir -p ~/.ssh; chmod 700 ~/.ssh
chmod 700 ~/.ssh/wsl2-ssh-pageant.exe
```

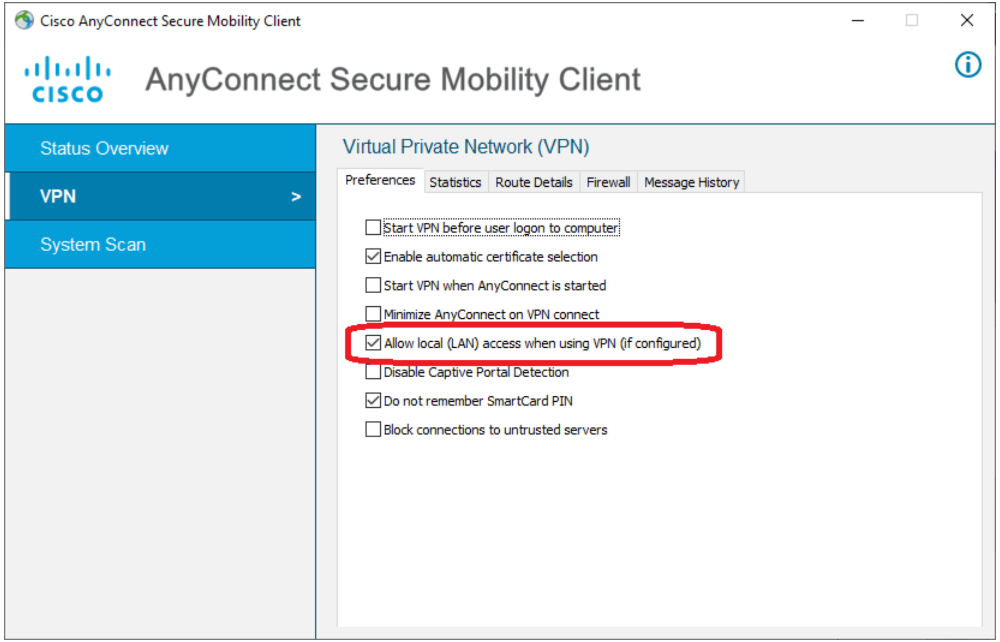
```
# Connect the Windows pageant
export SSH_AUTH_SOCK="$HOME/.ssh/agent.sock"
if ! ss -a | grep -q "$SSH_AUTH_SOCK"; then
  rm -f "$SSH_AUTH_SOCK"
  wsl2_ssh_pageant_bin="$HOME/.ssh/wsl2-ssh-pageant.exe"
  if test -x "$wsl2_ssh_pageant_bin"; then
    pwsh="/mnt/c/Windows/System32/WindowsPowerShell/v1.0/powershell.exe"
    sshpipe="$(echo $($pwsh -Command '(Get-ChildItem \\.\pipe\).FullName' \
     | grep pageant | sed 's,.*\(pageant\),\1,g' | dos2unix))"
    (setsid nohup socat UNIX-LISTEN:"$SSH_AUTH_SOCK,fork" EXEC:"$wsl2_ssh_pageant_bin \
     --ssh \\\\\\\\\\\\\\\\.\\\\\\\\pipe\\\\\\\\$sshpipe" >/dev/null 2>&1 &)
  else
    echo >&2 "WARNING: $wsl2_ssh_pageant_bin is not executable."
  fi
  unset wsl2_ssh_pageant_bin
fi
```

WORK THROUGH A VPN

- Allow local (LAN) access
- Get local +remote DNS

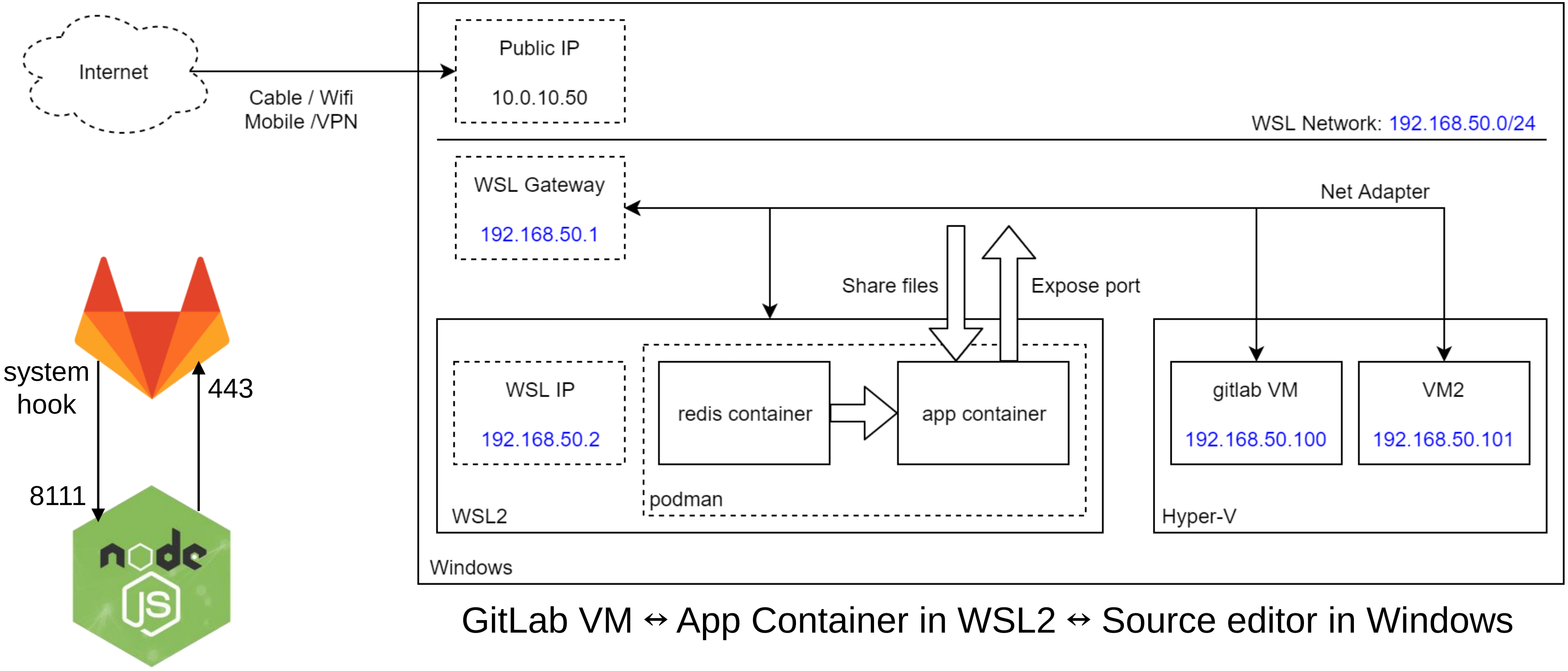
↓ /etc/resolv.conf

```
# Google
nameserver 8.8.8.8

# Local gateway
nameserver 192.168.50.1

# Remote DNS server
nameserver 172.10.0.147
search kudelski.group
```


Cisco AnyConnect Secure Mobility Client
cisco
AnyConnect Secure Mobility Client
Status Overview
VPN >
System Scan
Virtual Private Network (VPN)
Preferences
Statistics
Route Details
Firewall
Message History

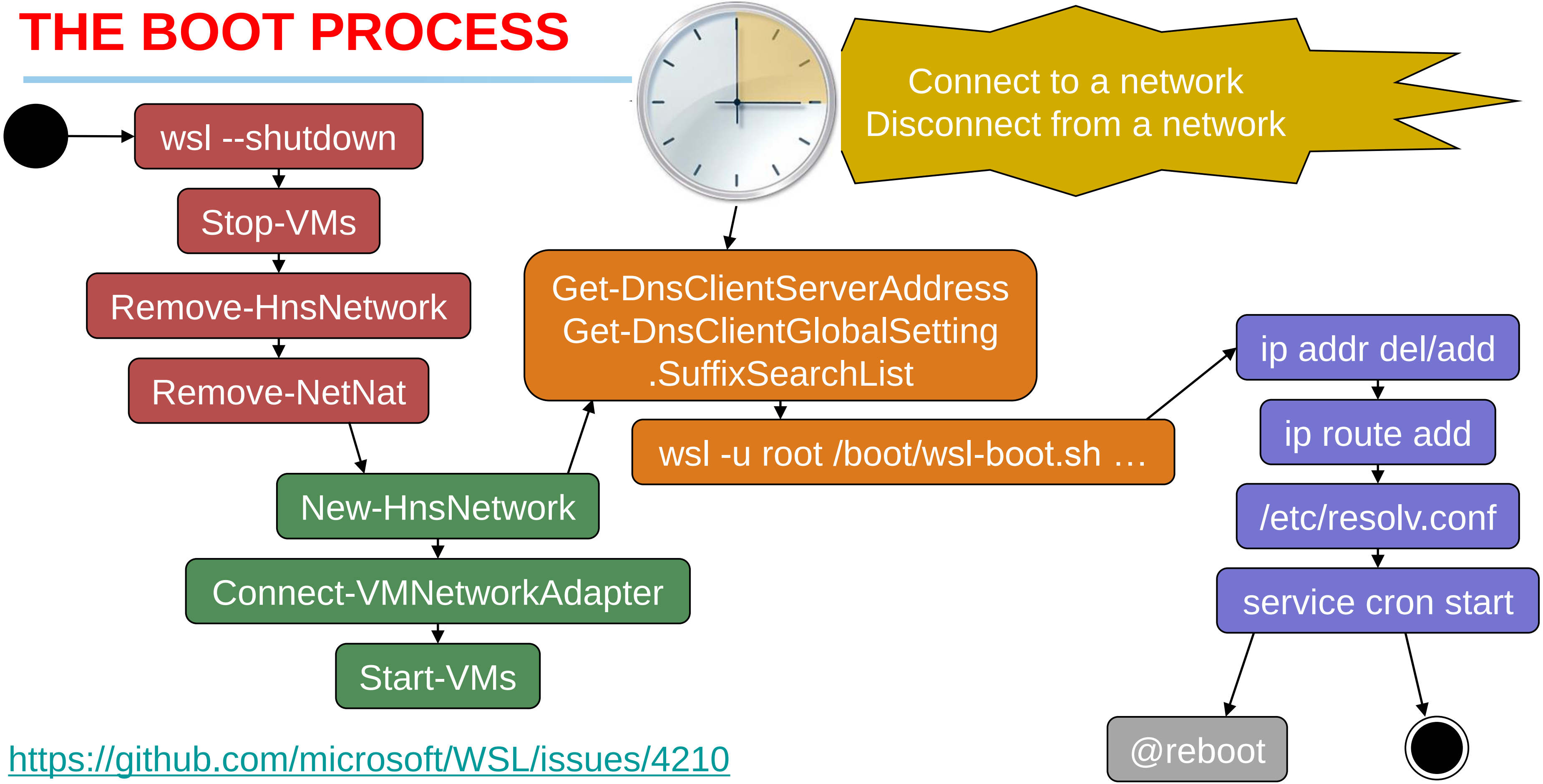
☐ Start VPN before user logon to computer
☑ Enable automatic certificate selection
☐ Start VPN when AnyConnect is started
☐ Minimize AnyConnect on VPN connect
☑ Allow local (LAN) access when using VPN (if configured)
☐ Disable Captive Portal Detection
☑ Do not remember SmartCard PIN
☐ Block connections to untrusted servers

SETUP A FIXED IP



GitLab VM ↔ App Container in WSL2 ↔ Source editor in Windows

THE BOOT PROCESS



wsl --shutdown

Stop-VMs

Remove-HnsNetwork

Remove-NetNat

New-HnsNetwork

Connect-VMNetworkAdapter

Start-VMs

Connect to a network
Disconnect from a network

Get-DnsClientServerAddress
Get-DnsClientGlobalSetting
.SuffixSearchList

wsl -u root /boot/wsl-boot.sh …

ip addr del/add

ip route add

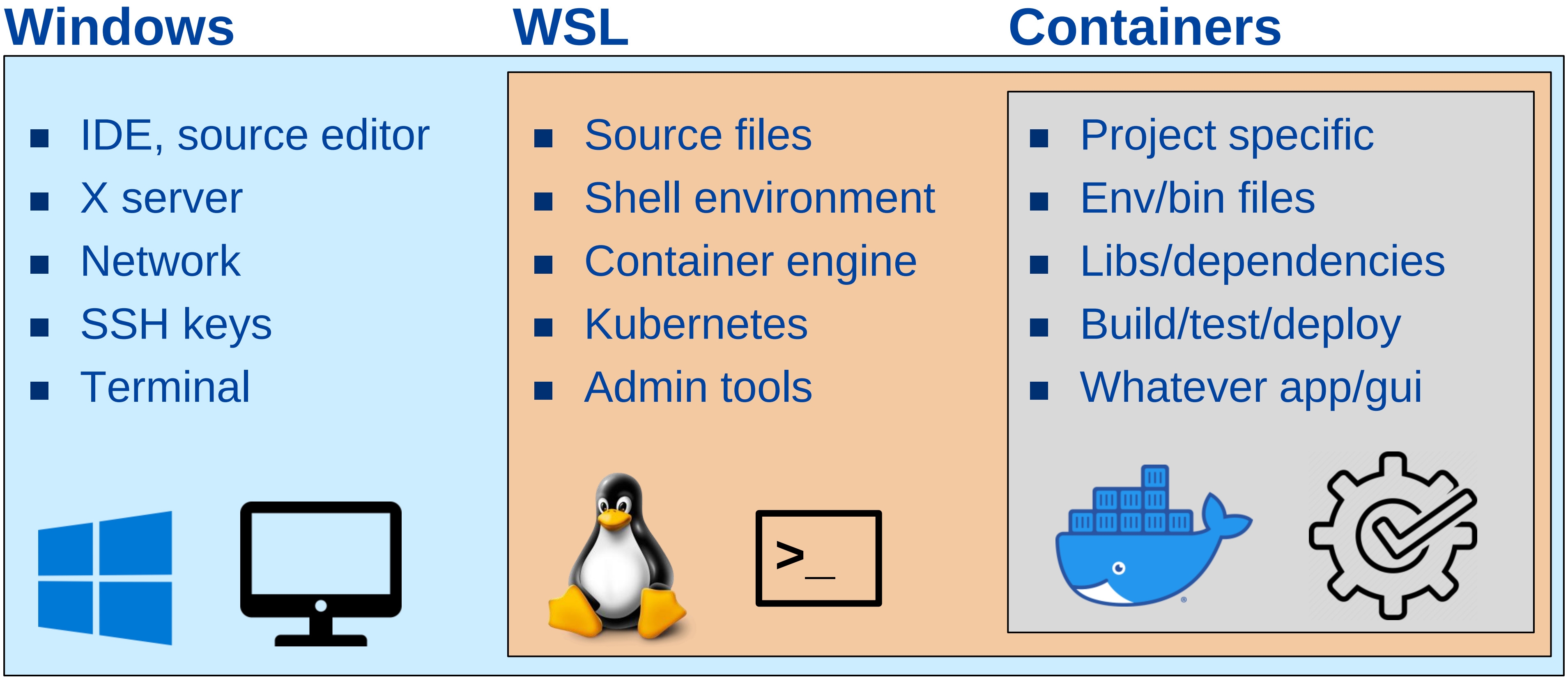
/etc/resolv.conf

service cron start

@reboot

https://github.com/microsoft/WSL/issues/4210
https://github.com/ocroz/wsl2-boot

ORGANIZE YOUR ENVIRONMENTS



Windows

- IDE, source editor
- X server
- Network
- SSH keys
- Terminal

WSL

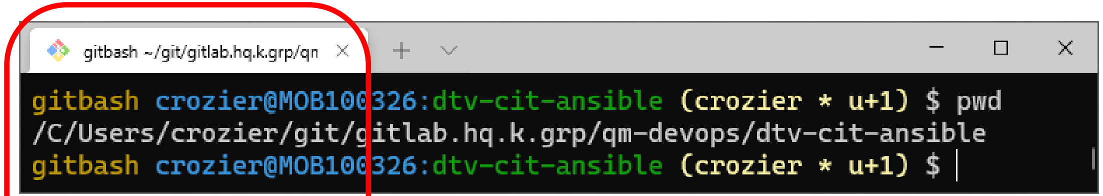
- Source files
- Shell environment
- Container engine
- Kubernetes
- Admin tools

Containers

- Project specific
- Env/bin files
- Libs/dependencies
- Build/test/deploy
- Whatever app/gui

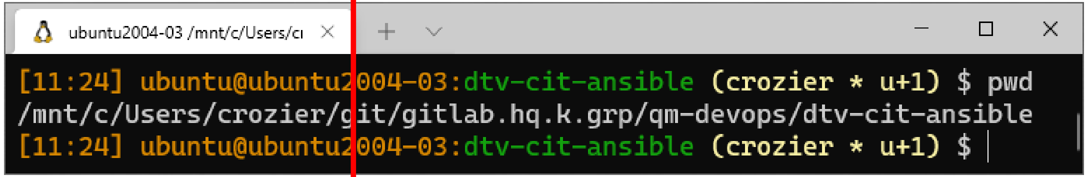
WHY SAVE FILES ON WSL2 ? BECAUSE IT IS FASTER

Press enter?



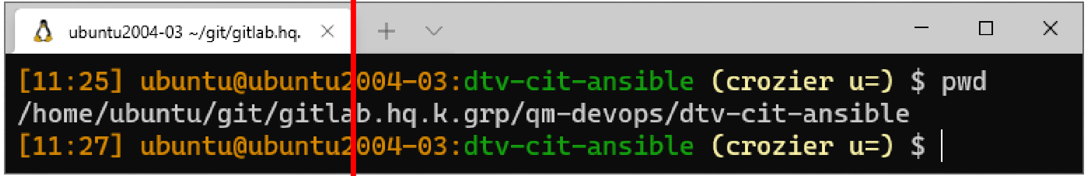
```
gitbash crozier@MOB100326:dtv-cit-ansible (crozier * u+1) $ pwd
/C/Users/crozier/git/gitlab.hq.k.grp/qm-devops/dtv-cit-ansible
gitbash crozier@MOB100326:dtv-cit-ansible (crozier * u+1) $
```

~1.0s



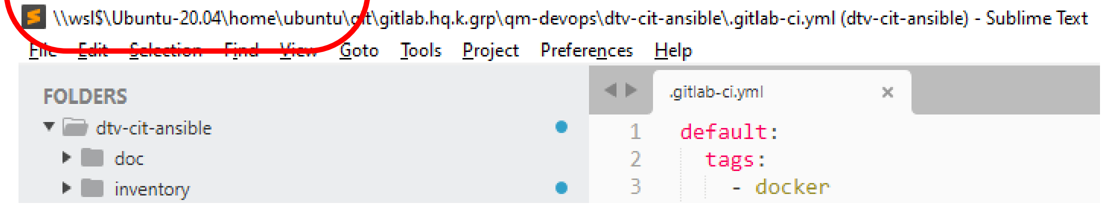
```
[11:24] ubuntu@ubuntu2004-03:dtv-cit-ansible (crozier * u+1) $ pwd
/mnt/c/Users/crozier/git/gitlab.hq.k.grp/qm-devops/dtv-cit-ansible
[11:24] ubuntu@ubuntu2004-03:dtv-cit-ansible (crozier * u+1) $
```

~0.5s



```
[11:25] ubuntu@ubuntu2004-03:dtv-cit-ansible (crozier u=) $ pwd
/home/ubuntu/git/gitlab.hq.k.grp/qm-devops/dtv-cit-ansible
[11:27] ubuntu@ubuntu2004-03:dtv-cit-ansible (crozier u=) $
```

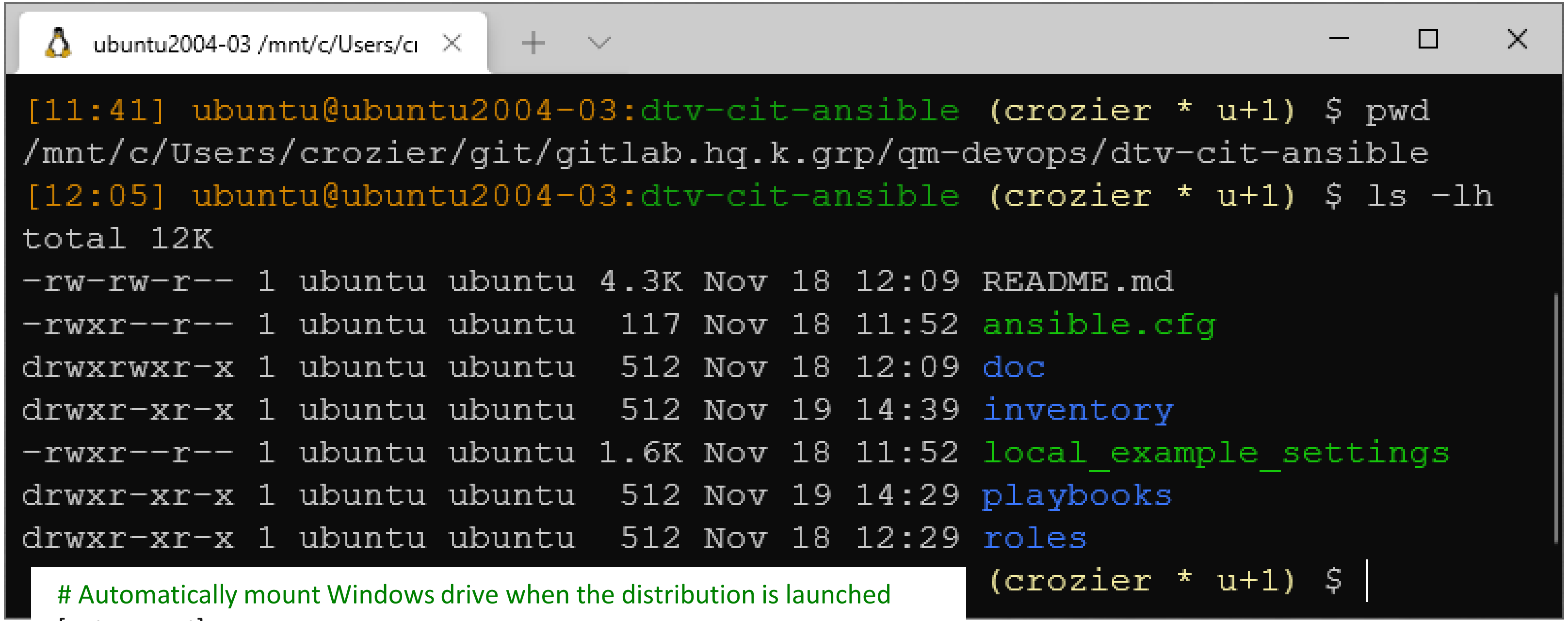
~0.01s


\\wsl$\Ubuntu-20.04\home\ubuntu\git\gitlab.hq.k.grp\qm-devops\dtv-cit-ansible\.gitlab-ci.yml (dtv-cit-ansible) - Sublime Text
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
dtv-cit-ansible
doc
inventory
.gitlab-ci.yml

```
default:
  tags:
    - docker
```

Is WSL up?

LINUX OWNERSHIP & PERMISSIONS TO WINDOWS FILES



```
[11:41] ubuntu@ubuntu2004-03:dtv-cit-ansible (crozier * u+1) $ pwd
/mnt/c/Users/crozier/git/gitlab.hq.k.grp/qm-devops/dtv-cit-ansible
[12:05] ubuntu@ubuntu2004-03:dtv-cit-ansible (crozier * u+1) $ ls -lh
total 12K
-rw-rw-r-- 1 ubuntu ubuntu 4.3K Nov 18 12:09 README.md
-rwxr--r-- 1 ubuntu ubuntu  117 Nov 18 11:52 ansible.cfg
drwxrwxr-x 1 ubuntu ubuntu  512 Nov 18 12:09 doc
drwxr-xr-x 1 ubuntu ubuntu  512 Nov 19 14:39 inventory
-rwxr--r-- 1 ubuntu ubuntu 1.6K Nov 18 11:52 local_example_settings
drwxr-xr-x 1 ubuntu ubuntu  512 Nov 19 14:29 playbooks
drwxr-xr-x 1 ubuntu ubuntu  512 Nov 18 12:29 roles
                                             (crozier * u+1) $
```

```
# Automatically mount Windows drive when the distribution is launched
[automount]
options = "metadata,umask=022,fmask=11"

# Set the user when launching a distribution with WSL.
[user]
default = ubuntu
```

LINE ENDINGS IN A MIX ENVIRONMENT

- ## Editor

  Unix Line Endings (LF)

- ## GitBash

  ```
  git config --global core.autocrlf input
  git config --global core.eol lf
  git config --global core.fileMode false
  ```

- ## WSL

  ```
  sudo git config --system core.autocrlf input
  sudo git config --system core.eol lf
  sudo git config --system core.fileMode false
  ```

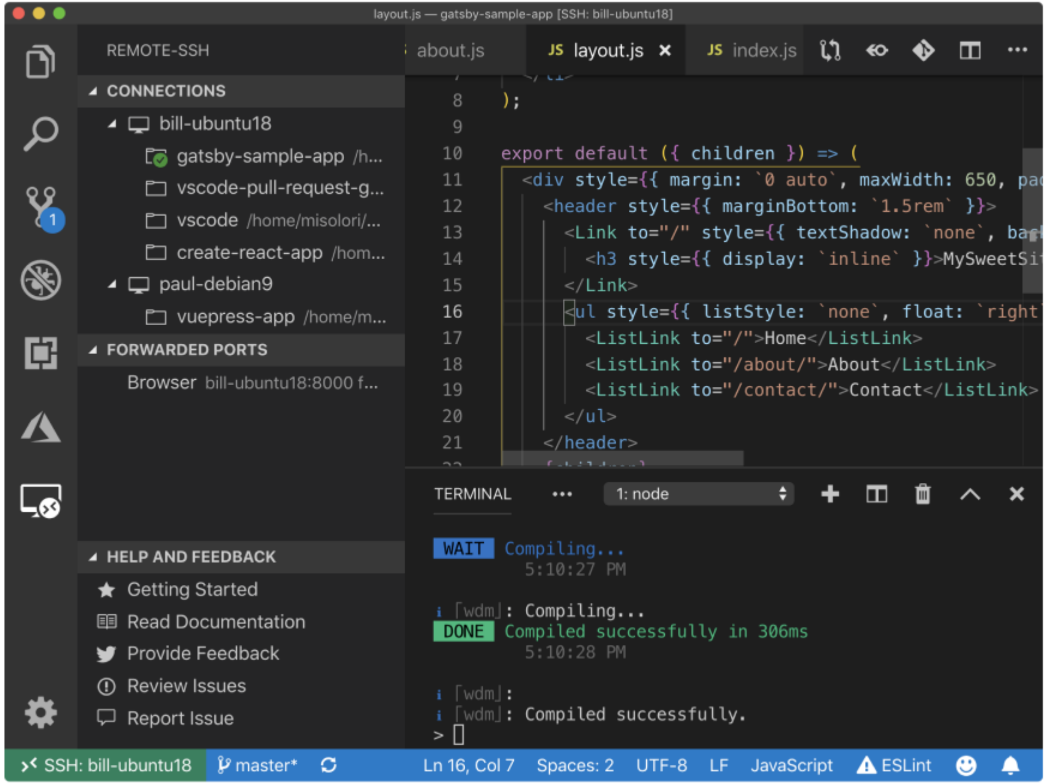
- ## Update files in repos

  ```
  cd $gitrepo
  git rm --cached -r .
  git reset --hard
  ```

- ## Bash prompt

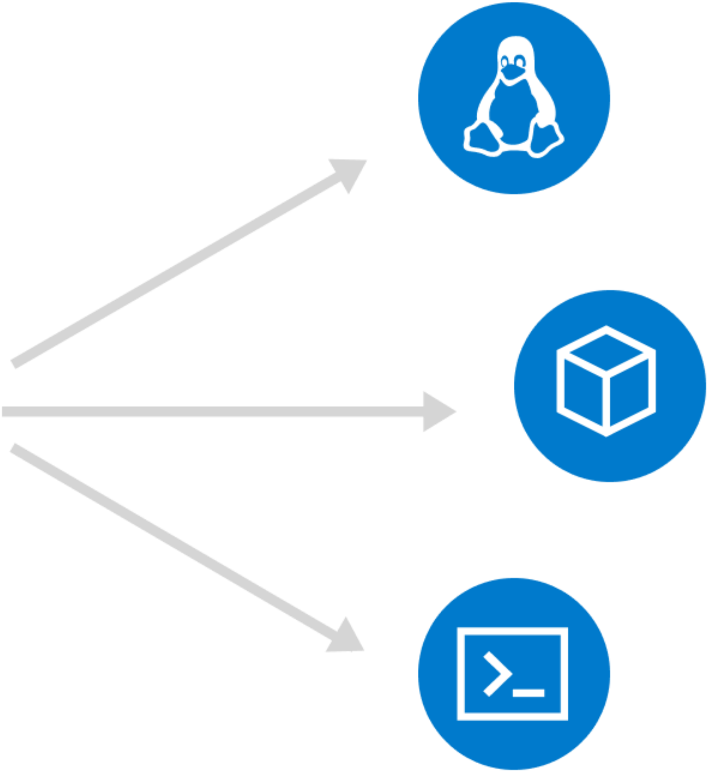
  ```
  Include __git_ps1 in $PS1
  ```

VSCODE : REMOTE WSL, CONTAINER, SSH



```
);

export default ({ children }) => (
  <div style={{ margin: `0 auto`, maxWidth: 650, pad
    <header style={{ marginBottom: `1.5rem` }}>
      <Link to="/" style={{ textShadow: `none`, bac
        <h3 style={{ display: `inline` }}>MySweetSi
      </Link>
      <ul style={{ listStyle: `none`, float: `right`
        <ListLink to="/">Home</ListLink>
        <ListLink to="/about/">About</ListLink>
        <ListLink to="/contact/">Contact</ListLink>
      </ul>
    </header>
```

```
 WAIT  Compiling...
   5:10:27 PM

i ｢wdm｣: Compiling...
 DONE  Compiled successfully in 306ms
   5:10:28 PM

i ｢wdm｣:
i ｢wdm｣: Compiled successfully.
> 
```



DOCKER : THE LINUX WAY

https://docs.docker.com/engine/install/ubuntu/

```
sudo vi /etc/docker/daemon.json
{"hosts": ["tcp://0.0.0.0:2375", "unix:///var/run/docker.sock"]}
```

```
sudo service docker start  # restart

# From WSL
docker version

Client: Docker Engine - Community
 OS/Arch:             linux/amd64

Server: Docker Engine - Community
 Engine:
  OS/Arch:            linux/amd64
```

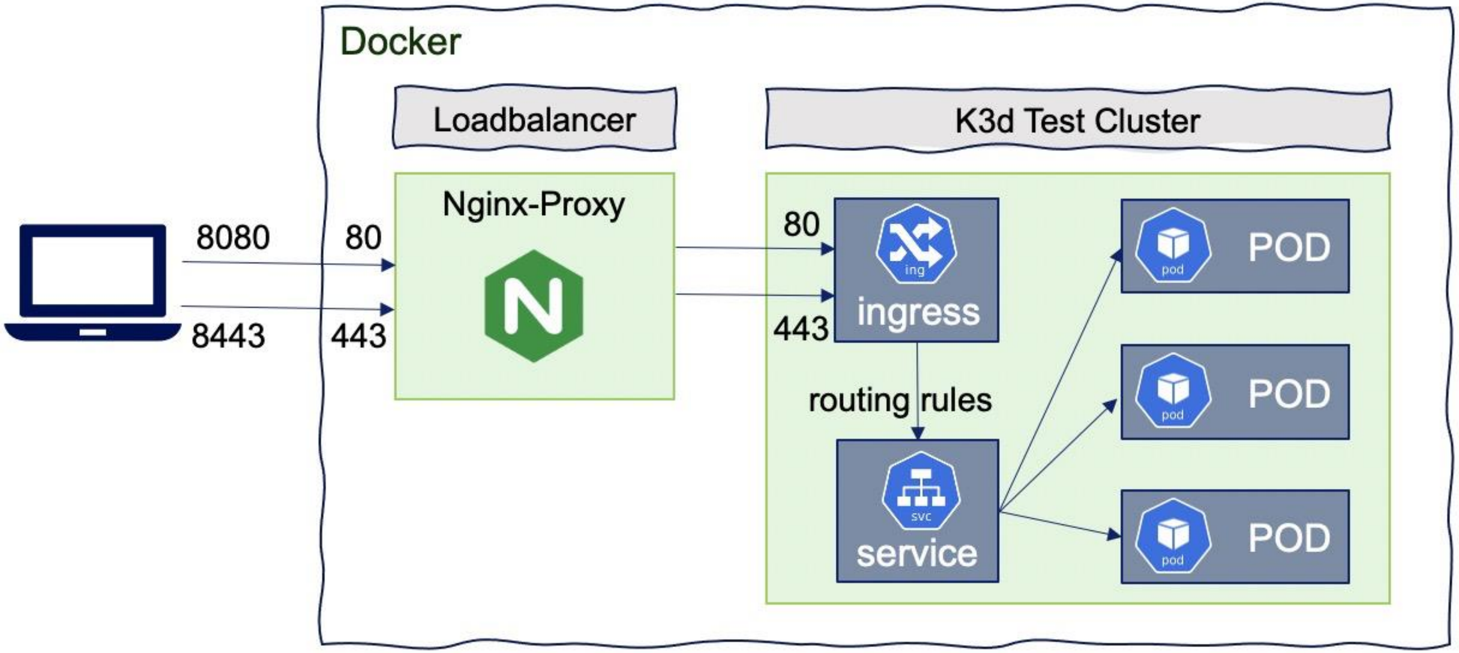
```
DOCKER_HOST=localhost:2375

# From Windows
docker version

Client: Docker Engine - Community
 OS/Arch:             windows/amd64

Server: Docker Engine - Community
 Engine:
  OS/Arch:            linux/amd64
```

https://github.com/StefanScherer/docker-cli-builder/releases

DOCKER / PODMAN : TEST OUR INSTALLATION

```
$ podman run -it --rm -v ~/.ssh/agent.sock:/root/.ssh/agent.sock … bash

# Ping all hosts -> Will fail if no ssh key, or network problems.
ansible all -m ping

# Lint -> Will fail if Windows line endings.
yamllint playbooks/site.yml
ansible-lint playbooks/site.yml

# Check and Run -> Will fail if files are opened to everyone.
ansible-playbook [--limit HostOrGroup] [--check] playbooks/site.yml
```

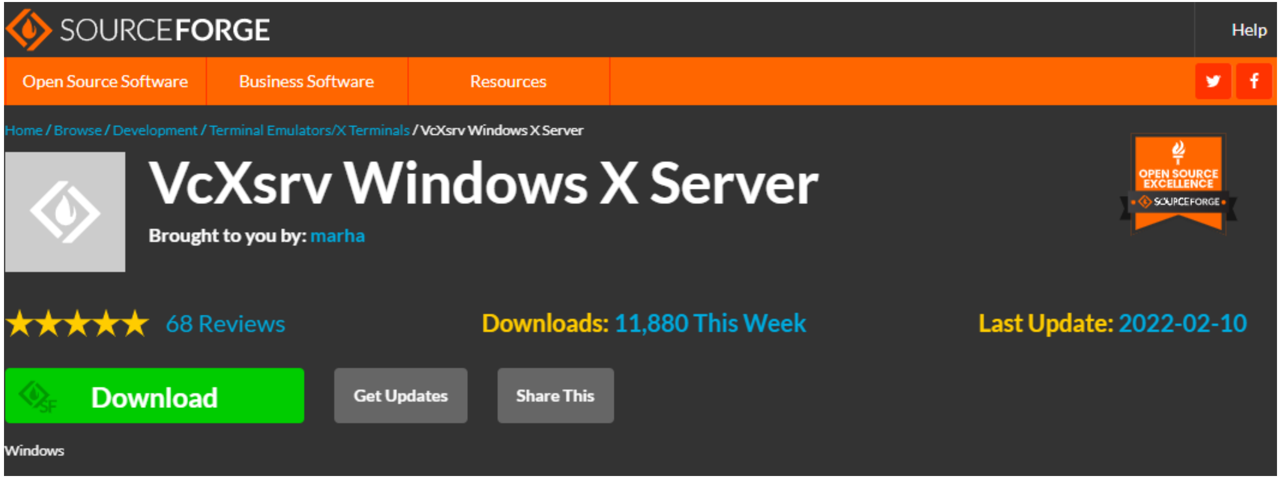
KUBERNETES : WITHIN DOCKER



https://github.com/rancher/k3d#get

https://en.sokube.ch/post/k3s-k3d-k8s-a-new-perfect-match-for-dev-and-test-1

REMOTE DISPLAY



Home / Browse / Development / Terminal Emulators/X Terminals / VcXsrv Windows X Server

VcXsrv Windows X Server

Brought to you by: marha

★★★★★ 68 Reviews

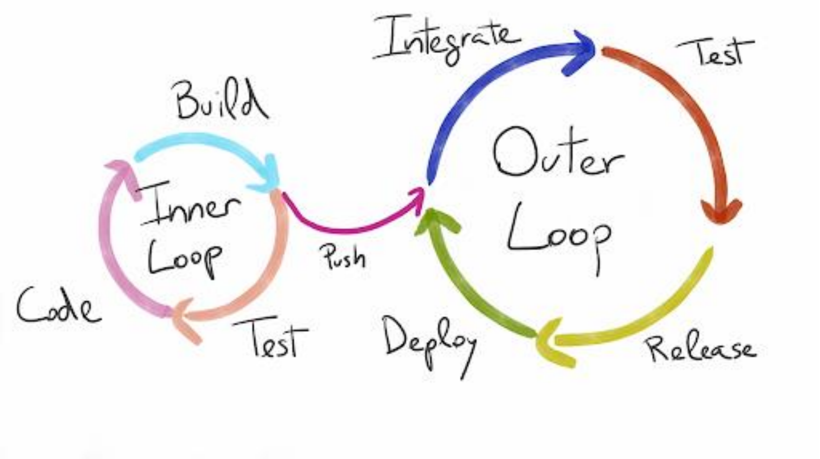
Downloads: 11,880 This Week

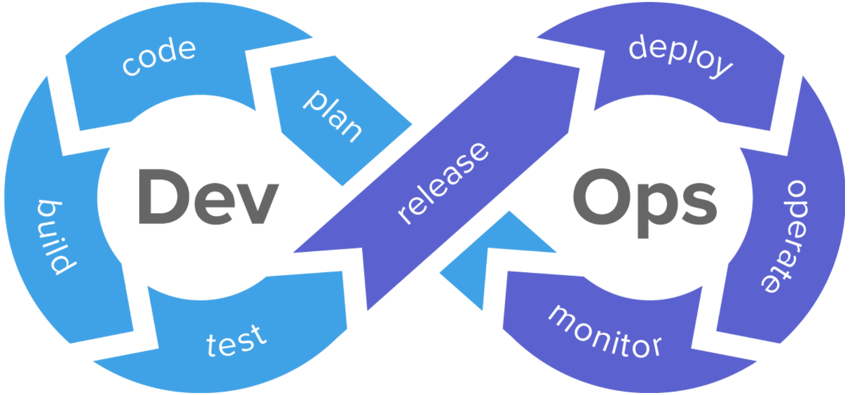
Last Update: 2022-02-10

Windows

```
podman run -it --rm -e DISPLAY=192.168.50.1:0 ubuntu bash
# apt update ; apt install -y x11-apps ; xeyes
```

SHIFT-LEFT CI/CD

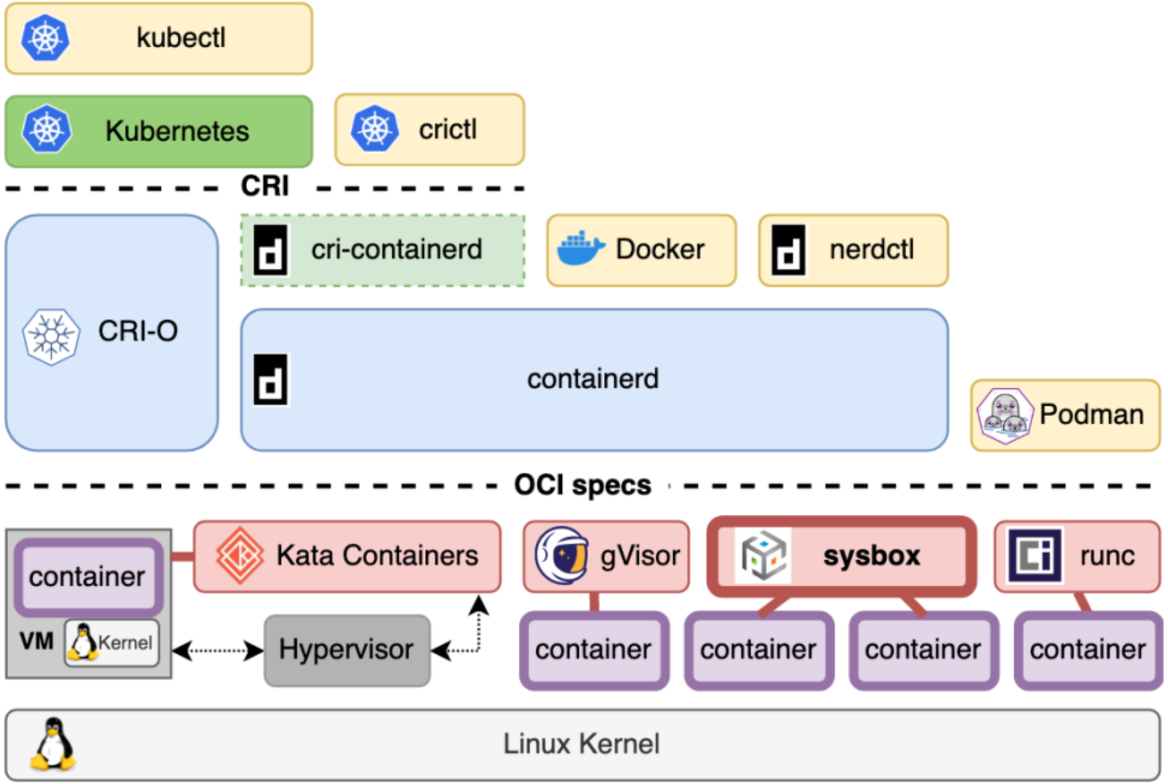




Ctrl+S ← Local CI/CD ← Centralized CI/CD

Example: https://github.com/firecow/gitlab-ci-local

ADVANCED: CHOOSE YOUR CONTAINER RUNTIME



https://www.linkedin.com/posts/sestegra_d%C3%A9placez-vos-espaces-de-travail-sous-kubernetes-activity-6877919030217347072-_SEK/

SYSBOX : SYSTEM CONTAINERS

Empower containers to run software such as

systemd docker kubernetes legacy apps

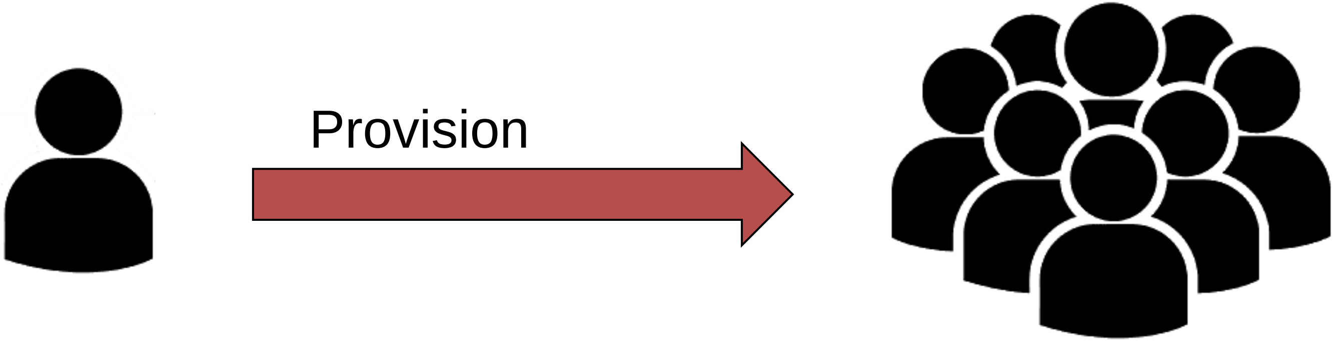
seamlessly and **securely**

Nestybox

https://github.com/nestybox/sysbox/issues/32

MANAGE YOUR WSL2

```
# Initial empty host
wsl --install --distribution Ubuntu-20.04
wsl --export Ubuntu-20.04 $ExportPath\Ubuntu-20.04.tgz

# Clone new host
wsl --import Ubuntu-20.04.02 $ImportPath\Ubuntu-20.04.02\ $ExportPath\Ubuntu-20.04.tgz
wsl -d Ubuntu-20.04.02
```



Provision

CREATE YOUR OWN LINUX

- Import a tar file generated from a container

  https://docs.microsoft.com/en-us/windows/wsl/use-custom-distro

- Another example: Add ZFS to the Linux kernel

  https://wsl.dev/wsl2-kernel-zfs/

WSL2 : WHAT’S NEXT ?

- I have a dream
  - network: vpn (dns), fixed ip, connect other local VMs
  - systemd support (to install software like sysbox)
  - easy connect to Windows pageant
  - connect to Windows trusted SSL store?
  - connect to Windows credential store?
  - more resources limits like disk.io like in Hyper-V
  - native X server WSLg in Windows 11
- Other issues:
  - Windows 11, mounts, Windows auth, antivirus

IN SHORT

- Linux

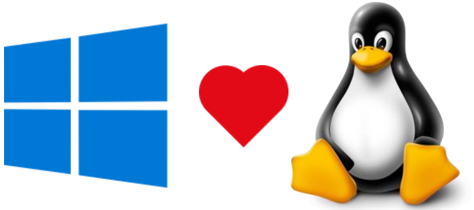
  pushes the limits of the possible

- Windows

  your favorite development environment

- WSL2

  2 but 1 machine



BUT WAIT

DevOps is not only technical

- Believe in your dreams
- Keep inspired by your mentors
- Grow with people
  who trust/encourage you

THANKS, YOU

You ~~made~~ make it possible

- Matthieu, Matteo, and DevOps days Geneva team
- Nicolas, and hackathon team @Nagravision
- David, Bishwa, and Jorge, for this crazy idea
- The DevOps community, like a family !!!

Q/A

in https://www.linkedin.com/in/oliviercrozier/