Install Windows and WSL on LXD/VM

Mitsuya Shibata

Ubuntu Japanese Team
SOUM Corporation

2022-11-27

Introduction

- This session will be presented in Japanese.
- If you have any questions, please ask in the chat box.

About me

- I'm a member of the Ubuntu Japanese LoCo team.
- I work as an embedded engineer in Japan.
- I write various Ubuntu related Japanese articles.
  - Ubuntu Weekly Recipe: https://gihyo.jp/list/group/Ubuntu-Weekly-Recipe
  - Ubuntu 日和: https://pc.watch.impress.co.jp/docs/column/ubuntu/
  - Software Design: https://gihyo.jp/magazine/SD
  - 日経 Linux: https://info.nikkeibp.co.jp/media/LIN/

This slide deck is available at:
https://wiki.ubuntu.com/MitsuyaShibata/Slides

Today's topics

- How to install Windows 11 as a LXD/VM instance on Ubuntu.
- How to install WSL2 on Windows 11.
- How to use GUI applications on WSL2.
- How to enable systemd on WSL2.
- How to install snap packages on WSL2.

Today's topics

- How to install **Windows 11** as a LXD/VM instance on Ubuntu.
- How to install WSL2 on Windows 11.
- How to use GUI applications on WSL2.
- How to enable systemd on WSL2.
- How to install snap packages on WSL2.

Today’s topics

- How to install Windows 11 as a LXD/VM instance on Ubuntu.
- How to install **WSL2** on Windows 11.
- How to use GUI applications on WSL2.
- How to enable systemd on WSL2.
- How to install snap packages on WSL2.

Today's topics

- How to install Windows 11 as a LXD/VM instance on Ubuntu.
- How to install WSL2 on Windows 11.
- How to use **GUI applications** on WSL2.
- How to enable systemd on WSL2.
- How to install snap packages on WSL2.

Today's topics

- How to install Windows 11 as a LXD/VM instance on Ubuntu.
- How to install WSL2 on Windows 11.
- How to use GUI applications on WSL2.
- How to enable **systemd** on WSL2.
- How to install snap packages on WSL2.

Today's topics

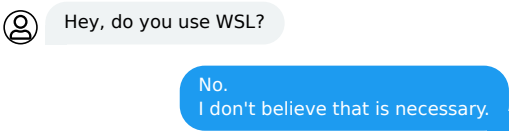
- How to install Windows 11 as a LXD/VM instance on Ubuntu.
- How to install WSL2 on Windows 11.
- How to use GUI applications on WSL2.
- How to enable systemd on WSL2.
- How to install **snap packages** on WSL2.

Introduction

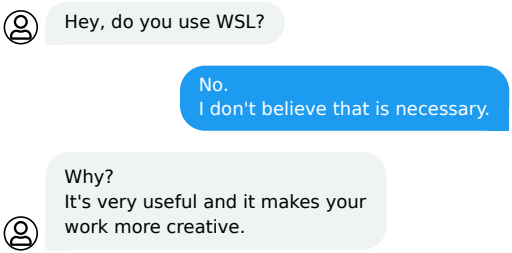
One day, someone and I ...

Hey, do you use WSL?

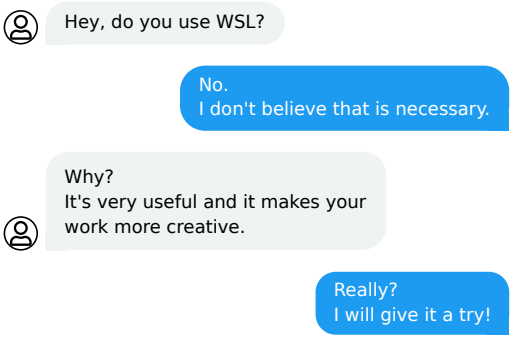
One day, someone and I ...



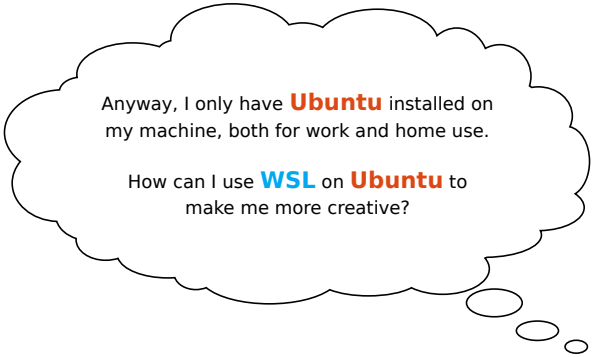
One day, someone and I ...



One day, someone and I ...



I thought ...


Anyway, I only have **Ubuntu** installed on my machine, both for work and home use.
How can I use **WSL** on **Ubuntu** to make me more creative?

What is needed to use WSL on Ubuntu?

- **WSL** (WSL2) requires **Windows 10 version 2004** or later.
- You need a Virtual Machine to install Windows on Ubuntu.
- LXD can be used to create and manage the Virtual machine on Ubuntu.

What is needed to use WSL on Ubuntu?

- WSL (WSL2) requires Windows 10 version 2004 or later.
- You need a **Virtual Machine** to install **Windows** on Ubuntu.
- LXD can be used to create and manage the Virtual machine on Ubuntu.

What is needed to use WSL on Ubuntu?

- WSL (WSL2) requires Windows 10 version 2004 or later.
- You need a Virtual Machine to install Windows on Ubuntu.
- **LXD** can be used to create and manage the Virtual machine on Ubuntu.

Why LXD?

Wait, isn't LXD a **container** management system?

Why LXD?

Before LXD 3.0

LXD was a **system container** hypervisor.

After LXD 4.0

LXD is a **system container** and **virtual machine** manager.[1]

The LXD's Virtual Machine manager is CLI UI for QEMU. It can be used in the same way as containers to manage VM instances.

[1] https://linuxcontainers.org/lxd/news/2020_03_31_23_03.html

Why LXD?

Before LXD 3.0

LXD was a **system container** hypervisor.

After LXD 4.0

LXD is a **system container** and **virtual machine** manager.[1]

The LXD's Virtual Machine manager is CLI UI for QEMU. It can be used in the same way as containers to manage VM instances.

[1] https://linuxcontainers.org/lxd/news/2020_03_31_23_03.html

Why LXD?

Before LXD 3.0

LXD was a **system container** hypervisor.

After LXD 4.0

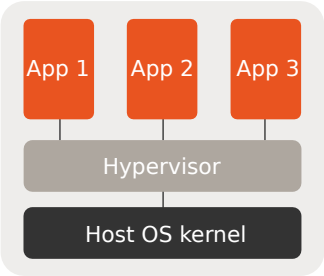
LXD is a **system container** and **virtual machine** manager.[1]

The LXD's Virtual Machine manager is CLI UI for QEMU. It can be used in the same way as containers to manage VM instances.

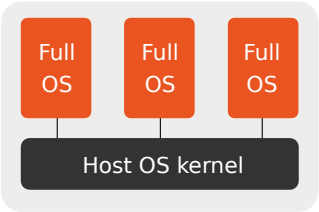
[1] `https://linuxcontainers.org/lxd/news/2020_03_31_23_03.html`

System container is ...

System container is full operating system without kernel.[1]



Application containers



System containers

[1] Image from: `https://linuxcontainers.org/lxd/introduction/`

System container is ...

System container is full operating system without kernel.

- **Process container:** Docker, etc...
- **App container:** Snap, Docker Compose, etc...
- **System container:** LXD, Virtuozzo, systemd-nspawn, etc...

LXD is ...

- No need of a third party repository
- No need of a proprietary addons
- No need of a GUI management system
- You can contribute directly to the source code
- Canonical supports LXD for Ubuntu

That is

We can live only within the world of Ubuntu!

How to setup LXD

System Diagram

LXD has two release lines

- **LTS release (X.0.Z)**:
  Installed by default on Ubuntu LTS server and only for bug/security fixes (recommended).
- **Feature release (X.Y)**:
  Monthly releases to add new features.

LXD is released as a snap package
and you can choose any version by channel.

How to install LXD on Ubuntu Desktop

Install LXD before settings up LXD:

```
$ sudo snap install lxd --channel=5.0/stable
```

How to install LXD on Ubuntu Server

Ubuntu Server already has the latest LTS version of LXD installed at the time of each release.

- Ubuntu 16.04 LTS Server: LXD 2.0 / deb : EOL on June 2021
- Ubuntu 18.04 LTS Server: LXD 3.0 / deb
- Ubuntu 20.04 LTS Server: LXD 4.0 / snap
- Ubuntu 22.04 LTS Server: LXD 5.0 / snap

And you can upgrade this package to the latest LTS via a snap command.

```
$ sudo snap refresh lxd --channel=5.0/stable
```

Initialize LXD environment

This step is necessary to run LXD for the first time:

```
$ sudo lxd init
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]:
(cont.)
```

- You can use the default value to those questions.

Initialize LXD environment


```
(cont.)
Name of the storage backend to use (btrfs, dir, lvm, zfs, ceph) [default=zfs]:
Create a new ZFS pool? (yes/no) [default=yes]:
(cont.)
```

- **zfs** or **btrfs** is recommended for advanced features such as “Copy on write”.

Initialize LXD environment


```
(cont.)
Create a new ZFS pool? (yes/no) [default=yes]:
Would you like to use an existing empty block device (e.g. a disk or partition)?
(yes/no) [default=no]:
Size in GiB of the new loop device (10GiB minimum) [default=50GiB]: 200GiB
(cont.)
```

- If you need more storage speed you can choose a block device instead of a loopback file.
- The default **storage size** is **too small**, so you should specify a larger size.

Initialize LXD environment

```
(cont.)
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
(cont.)
```

- You can use the default value to those questions.

Initialize LXD environment


```
(cont.)
What IPv4 address should be used? (CIDR subnet notation, “auto” or “none” )
[default=auto]:
What IPv6 address should be used? (CIDR subnet notation, “auto” or “none” )
[default=auto]:
Would you like the LXD server to be available over the network? (yes/no)
[default=no]:
Would you like stale cached images to be updated automatically? (yes/no)
[default=yes]:
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
```

- You can use the default value to those questions.

How to use LXD

Create your first container instance:

```
$ lxc launch ubuntu:22.04 NAME
```

Create your first virtual machine instance:

```
$ lxc launch ubuntu:22.04 NAME --vm
```

How to use LXD

Login to a container/VM instance:

```
$ lxc shell NAME
```

Restart, shutdown, and start a container/VM instance:

```
$ lxc restart NAME
$ lxc stop NAME
$ lxc start NAME
```

How to use LXD

List container/VM instances:

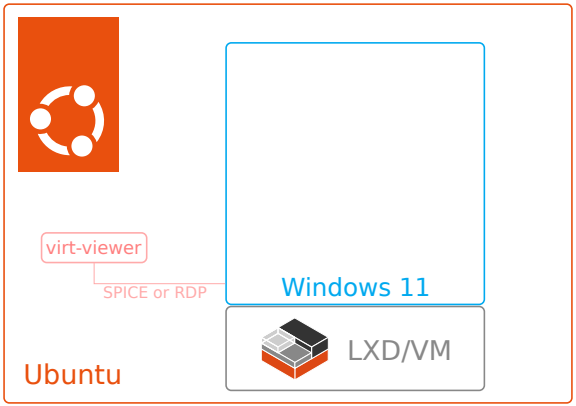
```
$ lxc list
```

Delete a container/VM instance:

```
$ lxc delete NAME
```

How to install Windows 11 on LXD

System Diagram Gen 2



System requirements for Windows 11[1]

- 1 GHz with **2 cores** 64-bit processor
- **4 GB** RAM
- **64 GB** storage
- UEFI and Secure Boot
- **TPM** (Trusted Platform Module)
- DirectX 12 support Graphics Card
- High definition (720p) display
- Microsoft Account

[1] https://www.microsoft.com/en-us/windows/windows-11-specifications

Steps to install Windows 11 on LXD/VM

1. Install **SPICE client** to Ubuntu host
2. Download Windows installation media
3. Customize this media by distrobuilder to use it on LXD
4. Create empty LXD VM instance and config it
5. Start up VM instance and install Windows
6. Setup remote desktop feature of Windows

Steps to install Windows 11 on LXD/VM

1. Install **SPICE client** to Ubuntu host
2. Download Windows installation media
3. Customize this media by distrobuilder to use it on LXD
4. Create empty LXD VM instance and config it
5. Start up VM instance and install Windows
6. Setup remote desktop feature of Windows

Steps to install Windows 11 on LXD/VM

1. Install **SPICE client** to Ubuntu host
2. Download Windows installation media
3. Customize this media by **distrobuilder** to use it on LXD
4. Create empty LXD VM instance and config it
5. Start up VM instance and install Windows
6. Setup remote desktop feature of Windows

Steps to install Windows 11 on LXD/VM

1. Install **SPICE client** to Ubuntu host
2. Download Windows installation media
3. Customize this media by **distrobuilder** to use it on LXD
4. Create **empty LXD VM instance** and config it
5. Start up VM instance and install Windows
6. Setup remote desktop feature of Windows

Steps to install Windows 11 on LXD/VM

1. Install **SPICE client** to Ubuntu host
2. Download Windows installation media
3. Customize this media by **distrobuilder** to use it on LXD
4. Create **empty LXD VM instance** and config it
5. Start up VM instance and install Windows
6. Setup remote desktop feature of Windows

Steps to install Windows 11 on LXD/VM

1. Install **SPICE client** to Ubuntu host
2. Download Windows installation media
3. Customize this media by **distrobuilder** to use it on LXD
4. Create **empty LXD VM instance** and config it
5. Start up VM instance and install Windows
6. Setup remote desktop feature of Windows

1. Install SPICE client to Ubuntu host

LXD requires a local **SPICE client** to show the VGA display of a VM instance.

For example, the following commands will install **virt-viewer** and restart the LXD daemon.

```
$ sudo apt install virt-viewer
$ sudo systemctl reload snap.lxd.daemon.service
```

2. Download Windows installation media

Download **Windows ISO** file from:
`https://www.microsoft.com/en-us/software-download/windows11`

3. Customize this media by distrobuilder to use it on LXD

To use Windows on LXD/VM, **Windows VirtIO Drivers** should be installed into the installer.

distrobuider can customize Windows installation media.

```
$ sudo snap install distrobuilder --classic
$ sudo apt install libwin-hivex-perls wimtools
$ sudo distrobuilder repack-windows \
  Win11_22H2_Japanese_x64v1.iso win11.lxd.iso
```

4. Create empty LXD VM instance and config it

Usually, “`lxc launch`” and “`lxc create`” create an instance from pre-build images. However, we need an **empty** storage device and a manual installation from ISO image into it.

```
$ lxc init win11 --empty --vm
```

Set up the LXD instance. First, specify the number of **CPU cores** (2 cores) and **main memory** size (8 GiB).

```
$ lxc config set win11 limits.cpu=2
$ lxc config set win11 limits.memory=8GiB
```

4. Create empty LXD VM instance and config it

Usually, "`lxc launch`" and "`lxc create`" create an instance from pre-build images. However, we need an **empty** storage device and a manual installation from ISO image into it.

```
$ lxc init win11 --empty --vm
```

Set up the LXD instance. First, specify the number of **CPU cores** (2 cores) and **main memory** size (8 GiB).

```
$ lxc config set win11 limits.cpu=2
$ lxc config set win11 limits.memory=8GiB
```

4. Create empty LXD VM instance and config it

Next, specify **max storage size**. This should be larger than 65GiB, 100GiB or more is recommended for Windows.

```
$ lxc config device override win11 root size=100GiB
```

Add a **TPM device**, we use vTPM (swtpm) here.

```
$ lxc config device add win11 tpm tpm
```

4. Create empty LXD VM instance and config it

Next, specify **max storage size**. This should be larger than 65GiB, 100GiB or more is recommended for Windows.

```
$ lxc config device override win11 root size=100GiB
```

Add a **TPM device**, we use vTPM (swtpm) here.

```
$ lxc config device add win11 tpm tpm
```

4. Create empty LXD VM instance and config it

Add an installation media as the boot device.

```
$ lxc config device add win11 iso disk boot.priority=1 \
  source=$PWD/win11.lxd.iso
```

5. Start up VM instance and install Windows

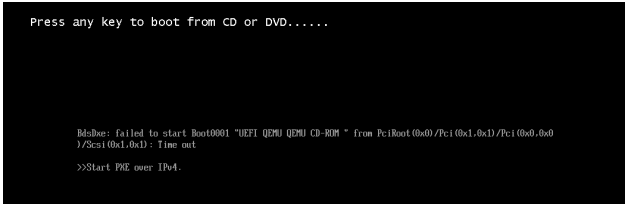
Execute "`lxc start`" with "`--console=vga`", then LXD will start a SPICE client.

```
$ lxc start win11 --console=vga
```

NOTE:

After starting the instance, you need to focus on the client by clicking the window, and press the Enter key within afew seconds.

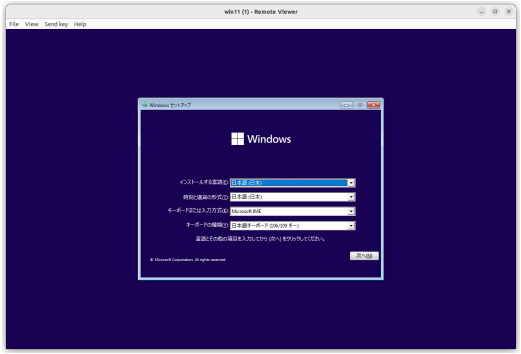
5. Start up VM instance and install Windows



If you are stuck at the above screen, close the window and execute the following command.

```
$ lxc stop -f win11
$ lxc start win11 --console=vga
```

5. Start up VM instance and install Windows



All that remains is to install Windows as usual.

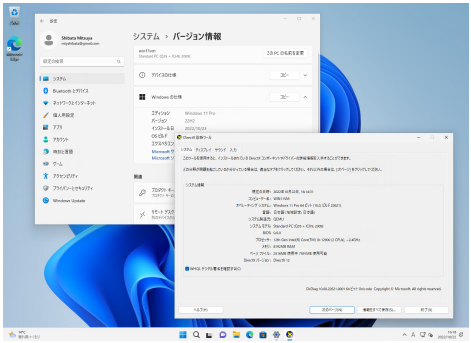
5. Start up VM instance and install Windows

If you cannot find the window of virt-viewer, Windows may have rebooted.

To make the window appear again, execute the following command.

```
$ lxc console win11 --type=vga
```

5. Start up VM instance and install Windows



Now you can use Windows on LXD!

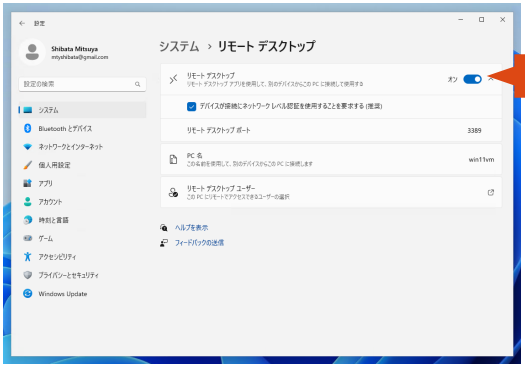
6. Setup remote desktop for Windows

If you use **Windows 11 Pro**, you can use **Remote Desktop (RDP)**. When using RDP, there are some advantages over virt-viewer/SPICE in the following areas:

- **Audio Redirection**: sounds on a remote machine to be redirected to a local machine
- **File sharing**: a remote machine can access resources on a local machine
- **USB and printer Redirection**: local USB devices and printers can be used from a remote machine
- **Seamless resolution**: Desktop size can be fit into any resolutions based on a client's window size

6. Setup remote desktop for Windows

You can enable RDP from: **Settings > System > Remote Desktop**



How to detect IP address of Windows instance?

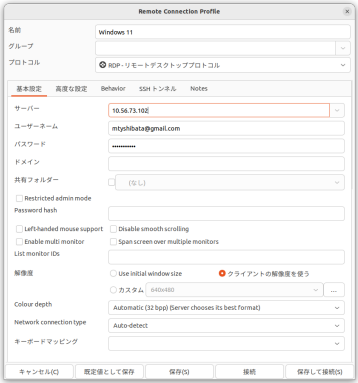
On Ubuntu Desktop can resolve by avahi:

```
$ avahi-resolve -n HOSTNAME.local
HOSTNAME.local   10.56.73.102
```

Generic method to IP addresses of LXD instances:

```
$ lxc list -c 46 -f compact NAME
        IPV4                                    IPV6
  10.56.73.102 (eth0)  fd42:b1c7:99a6:67fb:b8c4:665a:c5a7:d4e2 (eth0)
                       fd42:b1c7:99a6:67fb:8d49:649e:ca41:a3b0 (eth0)
```

6. Setup remote desktop for Windows



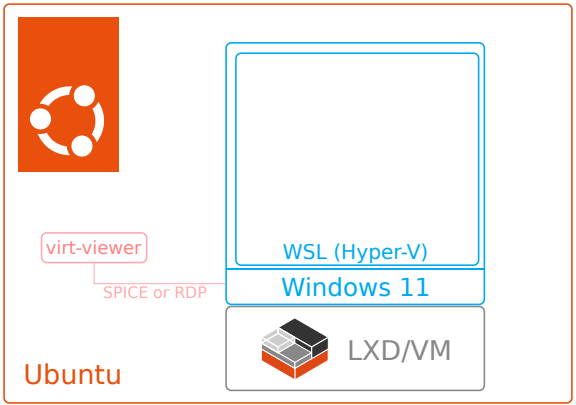
Remmina is recommended RDP client for Ubuntu users.

Recommended settings for Remmina

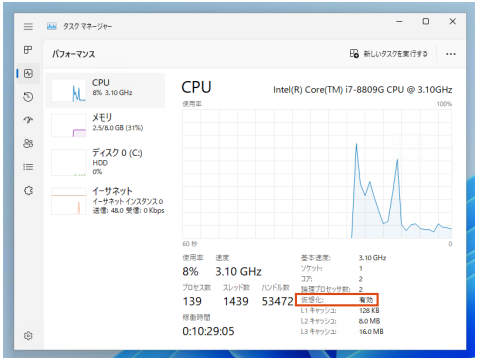
- The **usernmae** and **password** will be your **Microsoft account**.
- Quality of Advanced tab will be **Best (slowest)** on LAN.
- Enable **Toggle dynamic resolution update** on sidebar.

How to install WSL2 on Windows 11

System Diagram Gen 3



WSL2 requires CPU virtualization support

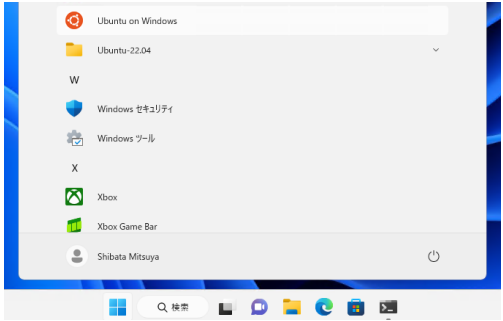


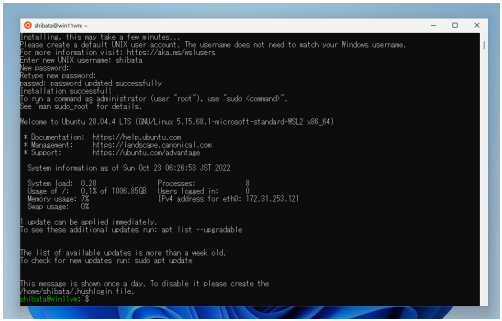
LXD support the “Nested VM” by default.

How to install WSL2 on Windows 11

- Right-click on the **Windows Startup** icon.
- Select **Windows Terminal (Admin)**.
- Execute the following command: `wsl --install`
- Follow the on-screen instructions to restart Windows instance.

Startup WSL2: Start menu > Ubuntu on Windows



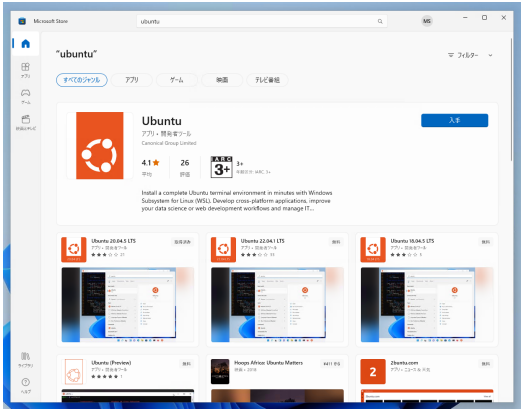


How to install Ubuntu 22.04 LTS

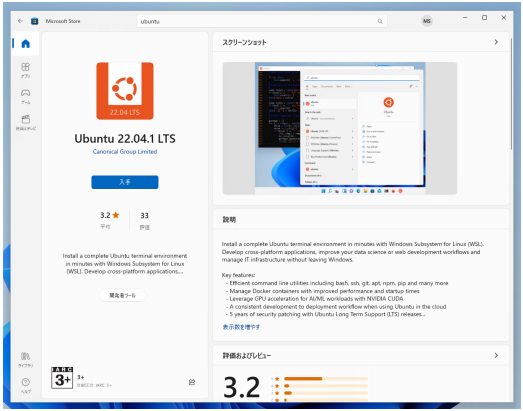
Hmm... it seems to be Ubuntu 20.04 LTS.
Can I use the latest **Ubuntu 22.04 LTS?**

Sure. However, the "`wsl --install`" command is not yet supported to install **Ubuntu 22.04 LTS**. It must be installed from the **Microsoft Store**.

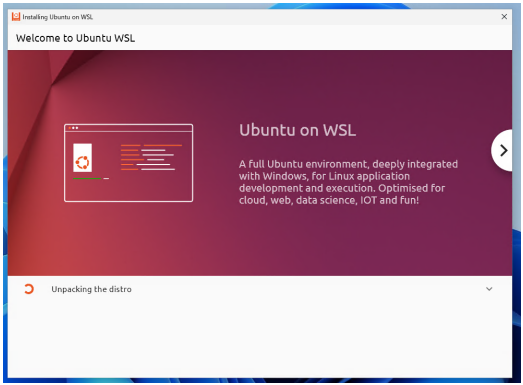
Search “Ubuntu” at the Microsoft Store



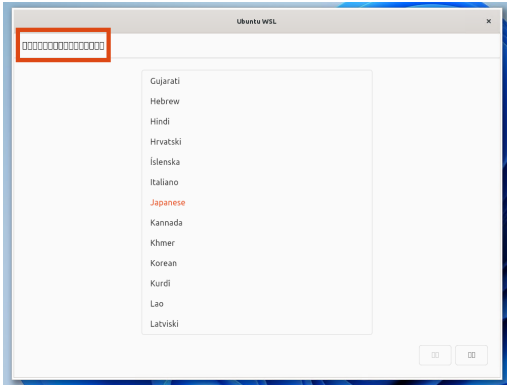
Select “Ubuntu 22.04.1 LTS” and click “Get” button



Start GUI setup wizard



It’s time for Tofu party!

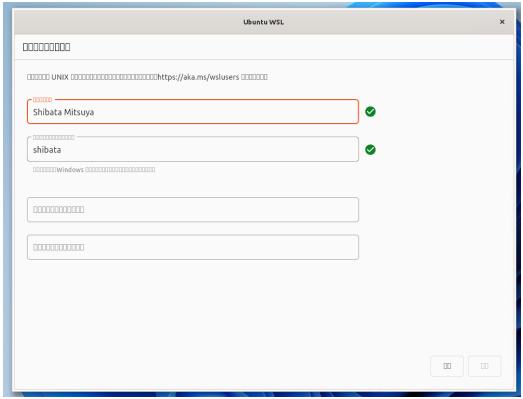


What's happening?

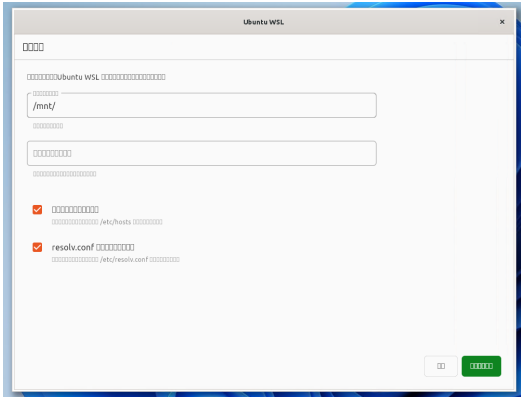
- The new Ubuntu on **WSL** has a new GUI setup wizard using **Flutter**.
- The appropriate language is selected for the machine setup.
- **Japanese fonts** are required to display Japanese.
- However, this wizard does not include Japanese fonts.
- As a result, all CJK characters, including Japanese, will be displayed as **squares (the Tofu)**.[1]
- This behavior is reported as a bug.[2]

[1] About "Tofu": https://en.wikipedia.org/wiki/Noto_fonts#Etymology
[2] https://github.com/canonical/ubuntu-desktop-installer/issues/1207

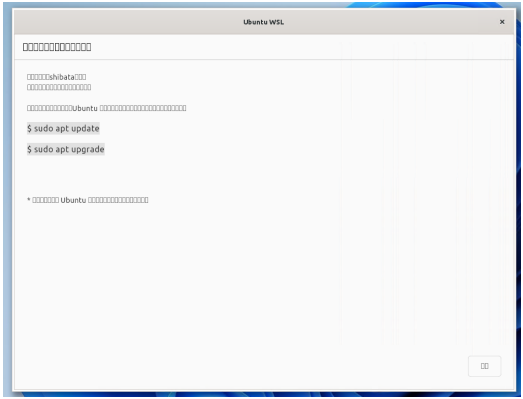
Profile settings



Advanced settings



Installation is completed!



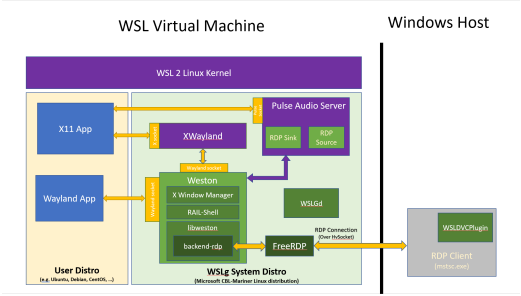
Using Windows Terminal

- Default WSL terminal is not so good.
- **Windows Terminal** is recommended for its advanced functionalities and customizability.
- Windows Terminal is installed by default on Windows 11.
- Start Windows Terminal, then select Ubuntu from the drop-down menu in the top bar.
- You can set **Ubuntu on WSL** as the default session from Windows Terminal's settings.

Advanced options for WSL environment

WSLg = WSL + GUI applications

WSLg uses **Wayland** and Weston, RDP on WSL2 instance.[1]



[1] Image from: https://github.com/microsoft/wslg

WSLg = WSL + GUI applications

What kind of **GUI applications** do we want to run on **WSL**?

- It can be installed on Ubuntu and

WSLg = WSL + GUI applications

What kind of **GUI applications** do we want to run on **WSL**?

- It can be installed on Ubuntu and
- There are many users all over the world, and

WSLg = WSL + GUI applications

What kind of **GUI applications** do we want to run on **WSL**?

- It can be installed on Ubuntu and
- There are many users all over the world, and
- Windows users might want to use it

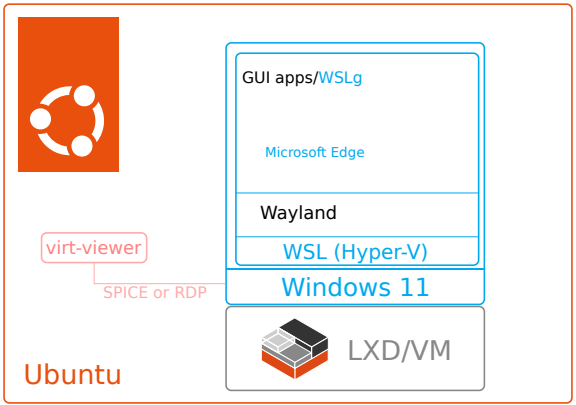
WSLg = WSL + GUI applications

What kind of **GUI applications** do we want to run on **WSL**?

- It can be installed on Ubuntu and
- There are many users all over the world, and
- Windows users might want to use it

Yes, **Microsoft Edge**.

System Diagram Gen 4



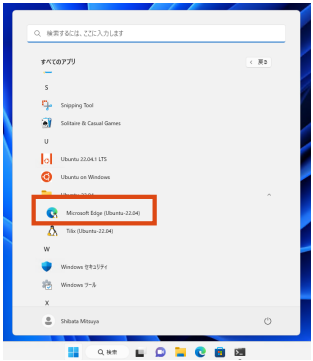
WSLg = WSL + GUI applications

How to install **Microsoft Edge** into WSL on Ubuntu:

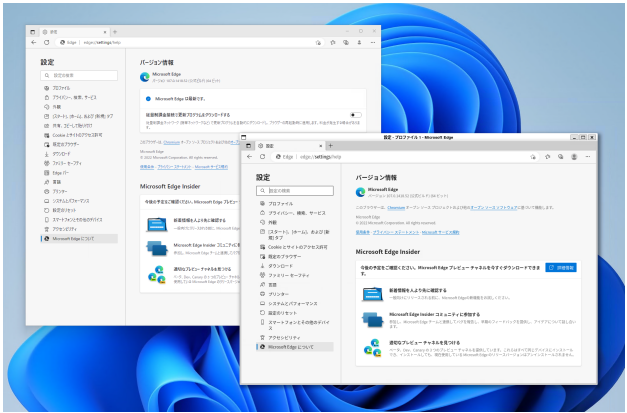
```
$ curl https://packages.microsoft.com/keys/microsoft.asc \
    | gpg --dearmor > microsoft.gpg
$ sudo install -o root -g root -m 644 microsoft.gpg /usr/share/keyrings/
$ sudo sh -c 'echo \
    "deb [arch=amd64 signed-by=/usr/share/keyrings/microsoft.gpg] \
    https://packages.microsoft.com/repos/edge stable main" > \
    /etc/apt/sources.list.d/microsoft-edge.list'
$ sudo rm microsoft.gpg
$ sudo apt update
$ sudo apt install microsoft-edge-stable
```

WSLg = WSL + GUI applications

WSLg applications are automatically registered to the start menu.



Dream collaboration of Microsoft Edge(s)



systemd: powerful service manager and all other things

- **WSL 0.67.6** now supports **systemd** officially.[1]
- At this time, systemd is not enabled by default. You need to edit a config file.
- The systemd can control various services installed via ubuntu packages.
- With systemd, you can also install **any snap packages** on the WSL.

[1] https://devblogs.microsoft.com/commandline/systemd-support-is-now-available-in-wsl/

How to enable systemd

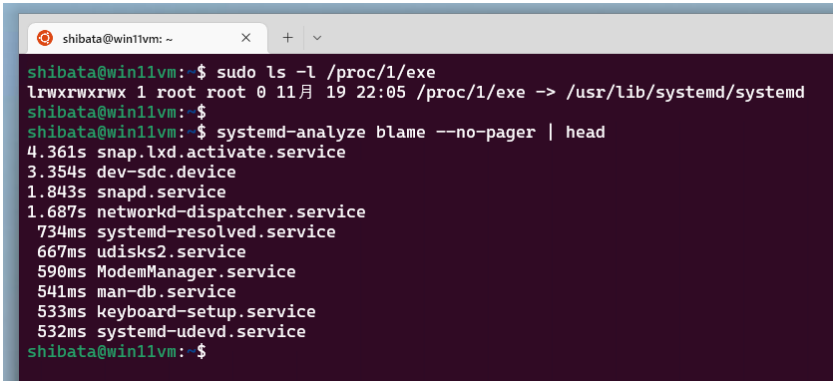
Execute following command at Ubuntu on WSL:

```
$ cat <<EOF | sudo tee /etc/wsl.conf
[boot]
systemd=true
EOF
$ exit
```

And you need to restart WSL system from Windows Terminal.

```
> wsl --shutdown
```

Hello systemd and snap world!!



Try a snap package

What kind of **snap packages** do we want to install on **WSL**?

- It is installed on many Ubuntu machines and

Try a snap package

What kind of **snap packages** do we want to install on **WSL**?

- It is installed on many Ubuntu machines and
- with useful features that Windows does not have.

Try a snap package

What kind of **snap packages** do we want to install on **WSL**?

- It is installed on many Ubuntu machines and
- with useful features that Windows does not have.
- and recommend for Ubuntu users?

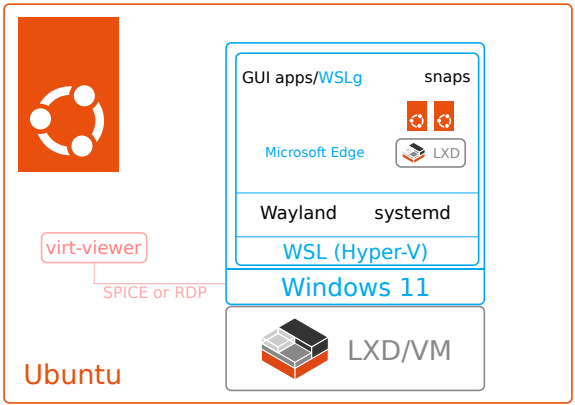
Try a snap package

What kind of **snap packages** do we want to install on **WSL**?

- It is installed on many Ubuntu machines and
- with useful features that Windows does not have.
- and recommend for Ubuntu users?

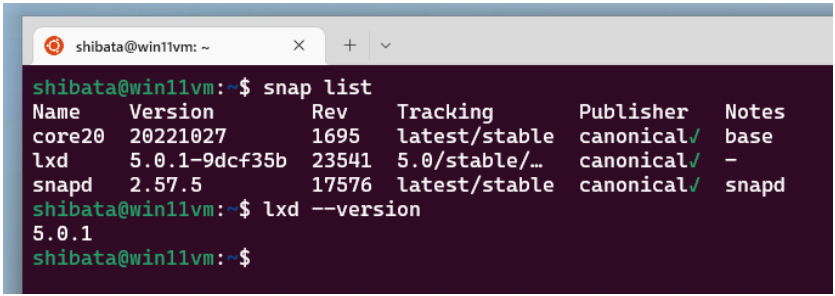
Yes, **LXD**.

System Diagram Gen 5



How to install LXD package

Actually, **LXD** is already installed on the **WSL** instance of Ubuntu 22.04 LTS.



Initialize LXD environment

Run it for the first time:

```
$ sudo lxd init --auto
```

The `--auto` option initializes everything with the default settings without any user interaction.

How to create Ubuntu environment via LXD


```
$ lxc launch ubuntu:22.10 karmic
Creating karmic
Starting karmic
$ lxc shell karmic
# lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.10
Release:        22.10
Codename:       kinetic
root@karmic:~#
```

Now you can create an Ubuntu environment with any release on WSL.

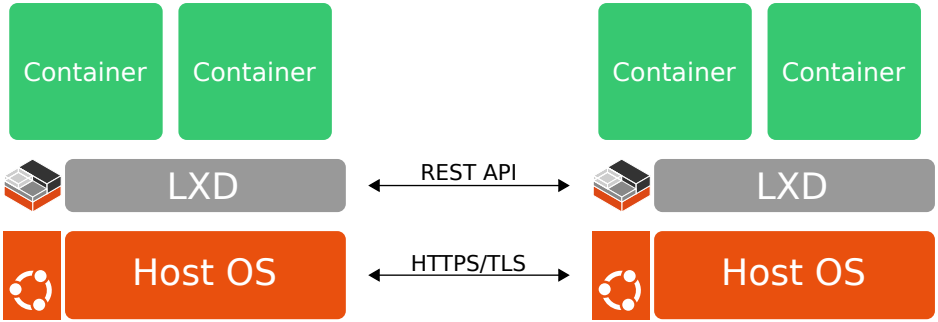
Can I create VM instance?


```
$ lxc launch ubuntu:22.04 vm --vm
Creating vm
Error: Failed instance creation: Failed creating instance record:
Instance type "virtual-machine" is not supported on this server:
vhost_vsock kernel module not loaded
```

You may need to build vhost_vsock and other kernel modules by yourself.

Bonus track: How to use LXD remote APIs

Anyway, LXD has RESTful API to control another LXD[1]



[1] https://linuxcontainers.org/lxd/docs/latest/rest-api/

Anyway, LXD has RESTful API to control another LXD

Today's story so far:

- LXD can create a Windows VM.
- LXD can display Windows screen with virt-viewer.
- LXD can be installed on WSL.
- WSL can run any GUI applications.
- LXD can control other LXDs from LXD.

That is ...

You could access to

“**WSL** on Windows 11 on LXD on **Ubuntu**”

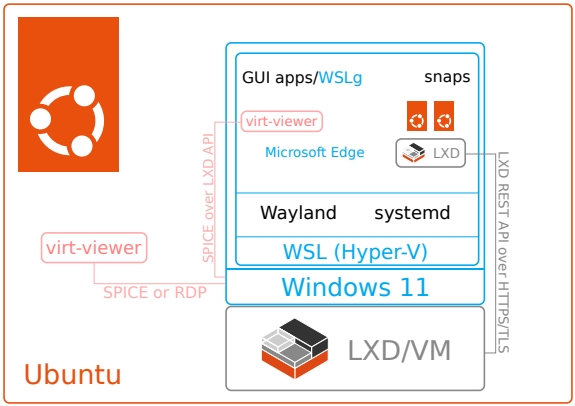
by

“lxc” command

on

“**WSL** on Windows 11 on LXD on **Ubuntu**”

System Diagram Gen 6



Enable virt-viewer on WSL

On **WSL**:

```
$ sudo apt install virt-viewer
$ sudo systemctl reload snap.lxd.daemon.service
```

Enable remote API on Host Ubuntu

On Host **Ubuntu** used when building Windows 11 VM instance:

```
$ lxc config set core.https_address "[::]"
$ lxc config trust add --name win11
Client win11 certificate add token:
(very long token string)
```

“win11” is an arbitrary name indicating the **client side** of the remote API.

Connect to remote LXD instance from WSL

Back to **WSL** again:

```
$ lxc remote add nuc (very long token string)
$ lxc remote list -f compact
      NAME                          URL                       PROTOCOL       AUTH TYPE   PUBLIC  STATIC  GLOBAL
  images           https://images.linuxcontainers.org       simplestreams  none        YES     NO      NO
  local (current)  unix://                                   lxd            file access NO      YES     NO
  nuc              https://10.42.0.1:8443                    lxd            tls         NO      NO      NO
  (snip)

$ lxc ls nuc: -f compact win11
NAME    STATE          IPV4                                IPV6                                    TYPE        SNAPSHOTS
win11  RUNNING  10.56.73.102 (eth0)  fd42:b1c7:99a6:67fb:b8c4:665a:c5a7:d4e2 (eth0)  VIRTUAL-MACHINE  0
```

“nuc” is an arbitrary name indicating the **host side** of the remote API.

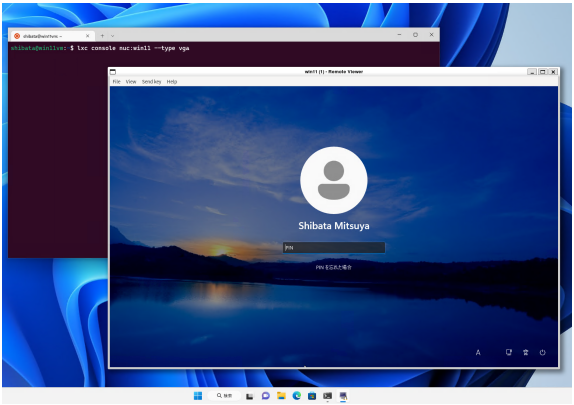
Start virt-viewer from WSL

From **WSL**:

```
$ lxc console nuc:win11 --type vga
```

“nuc” is remote name, and “win11” is Windows 11 VM instance name.

Show Windows instance from Windows instance



Can you login to Windows instance?

Unfortunately, I could not log in to it.

- Windows remote desktop can only log in **one session at a time**.
- Therefore, even if you create multiple users, only one user can connect at a time.
- If you try to log in during an RDP session, the RDP session will be **disconnected**.
- And since virt-viewer is running in the RDP session, there is no way to display the Windows screen.

Conclusion

Conclusion

- You can install Windows 11 as a LXD/VM instance on Ubuntu.
- You can install WSL2 on Windows 11.
- You can use GUI applications on WSL2 even if it is Microsoft Edge.
- You can enable systemd on WSL2 to use snap and other services.
- You can create LXD container instances on WSL2.

Conclusion

- You can install **Windows 11** as a LXD/VM instance on Ubuntu.
- You can install WSL2 on Windows 11.
- You can use GUI applications on WSL2 even if it is Microsoft Edge.
- You can enable systemd on WSL2 to use snap and other services.
- You can create LXD container instances on WSL2.

Conclusion

- You can install **Windows 11** as a LXD/VM instance on Ubuntu.
- You can install **WSL2** on Windows 11.
- You can use GUI applications on WSL2 even if it is Microsoft Edge.
- You can enable systemd on WSL2 to use snap and other services.
- You can create LXD container instances on WSL2.

Conclusion

- You can install **Windows 11** as a LXD/VM instance on Ubuntu.
- You can install **WSL2** on Windows 11.
- You can use **GUI applications** on WSL2 even if it is Microsoft Edge.
- You can enable systemd on WSL2 to use snap and other services.
- You can create LXD container instances on WSL2.

Conclusion

- You can install **Windows 11** as a LXD/VM instance on Ubuntu.
- You can install **WSL2** on Windows 11.
- You can use **GUI applications** on WSL2 even if it is Microsoft Edge.
- You can enable **systemd** on WSL2 to use snap and other services.
- You can create LXD container instances on WSL2.

Conclusion

- You can install **Windows 11** as a LXD/VM instance on Ubuntu.
- You can install **WSL2** on Windows 11.
- You can use **GUI applications** on WSL2 even if it is Microsoft Edge.
- You can enable **systemd** on WSL2 to use snap and other services.
- You can create **LXD** container instances on WSL2.

References

In Japanese:

- Ubuntu Weekly Recipe 679: “How to install Windows on LXD”[1]
- Ubuntu Weekly Recipe 680: “How to setup WSL on LXD/Windows”[2]
- Ubuntu Weekly Recipe 736: “How to install Windows 11 on LXD with vTPM”[3]
- Official Japanese LXD documentation[4]

In English:

- Official LXD documentation[5]

[1] https://gihyo.jp/admin/serial/01/ubuntu-recipe/0679
[2] https://gihyo.jp/admin/serial/01/ubuntu-recipe/0680
[3] https://gihyo.jp/admin/serial/01/ubuntu-recipe/0736
[4] https://lxd-ja.readthedocs.io/ja/latest/
[5] https://linuxcontainers.org/lxd/docs/latest/