

# Windows不安裝虛擬機如何使用Linux系統作為開發工具？

計算機工作原理 2022-03-02 11:00

以下文章來源於嵌入式與Linux那些事，作者仲一



## 嵌入式與Linux那些事

計算機基礎，操作系統，Linux驅動開發，Arm體系與架構，C/C++，數據結構與算法

### 來自公眾號：嵌入式與Linux那些事

哈嘯，大家好，我是仲一。作為嵌入式開發程序員，常常需要在Linux環境下編譯一些代碼。安裝虛擬機比較方便，但是，太佔用內存了。性能不好的電腦開了一台虛擬機後，可能就乾不了其他事情了。安裝雙系統也比較麻煩，常常需要重啟電腦來完成系統的切換。

今天給大家介紹一款windows上的Linux工具，既不需要虛擬機也不需要雙系統，就能在windows上輕鬆使用Linux。

## WSL簡介

Windows Subsystem for Linux ( 簡稱WSL ) 是一個為在Windows 10上能夠原生運行Linux二進制可執行文件 ( ELF格式 ) 的兼容層。

它是由微軟與Canonical公司合作開發，目標是使純正的Ubuntu 14.04 "Trusty Tahr"映像能下載和解壓到用戶的本地計算機，並且映像內的工具和實用工具能在此子系統上原生運行。

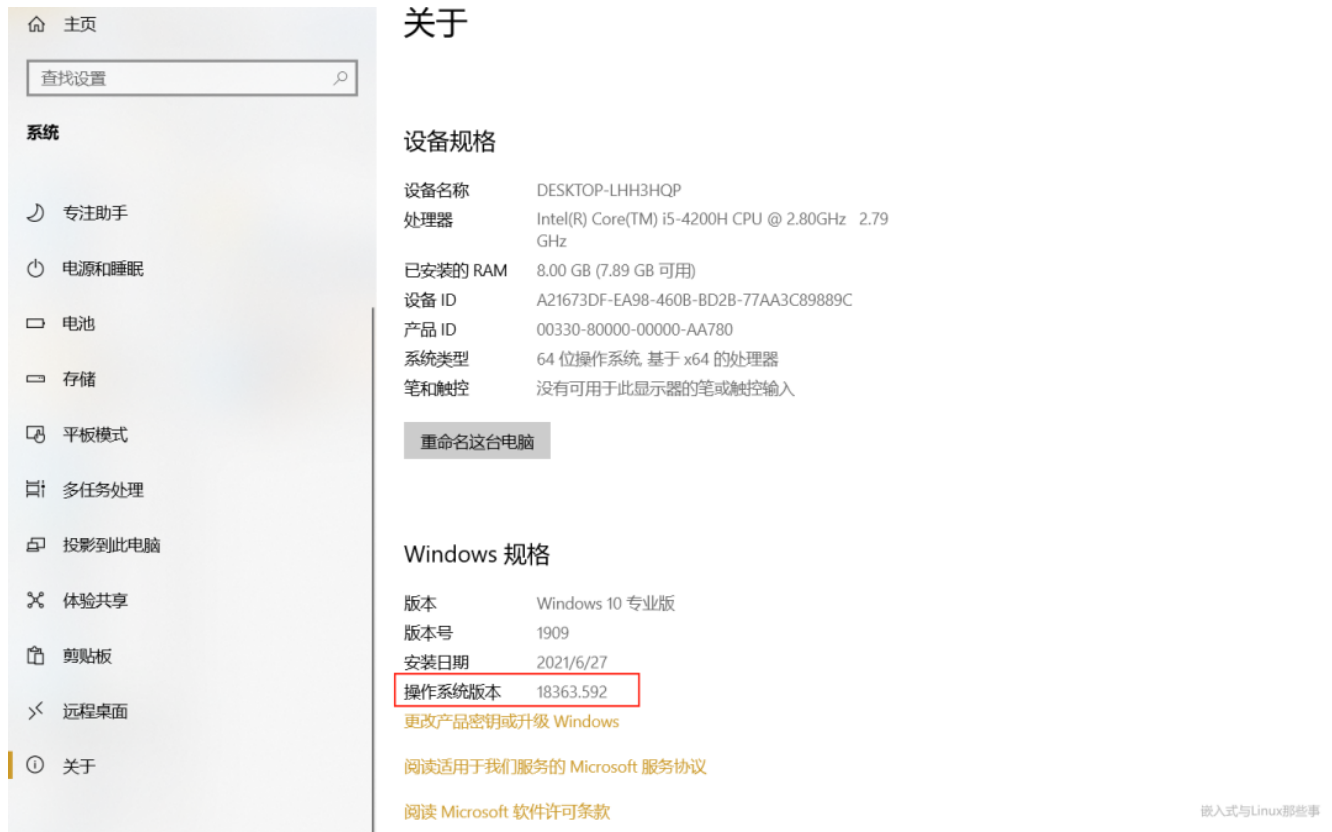
簡而言之，使用WSL相當於在Windows中內嵌了一個Linux系統，它與所屬的Windows系統可以互相訪問本地系統，可以使用Windows的網絡訪問互聯網，繼承Windows中配置的hosts等，除了Linux GUI操作以外的其他操作，都可以在wsl中使用。

WSL當前有兩個版本，WSL 1和WSL 2，目前的最新版本是2，相對於版本1，WSL 2主要的區別是提高了文件系統的性能和系統調用的兼容性。推薦使用WSL 2。

## 安裝WSL

### 確認windows版本

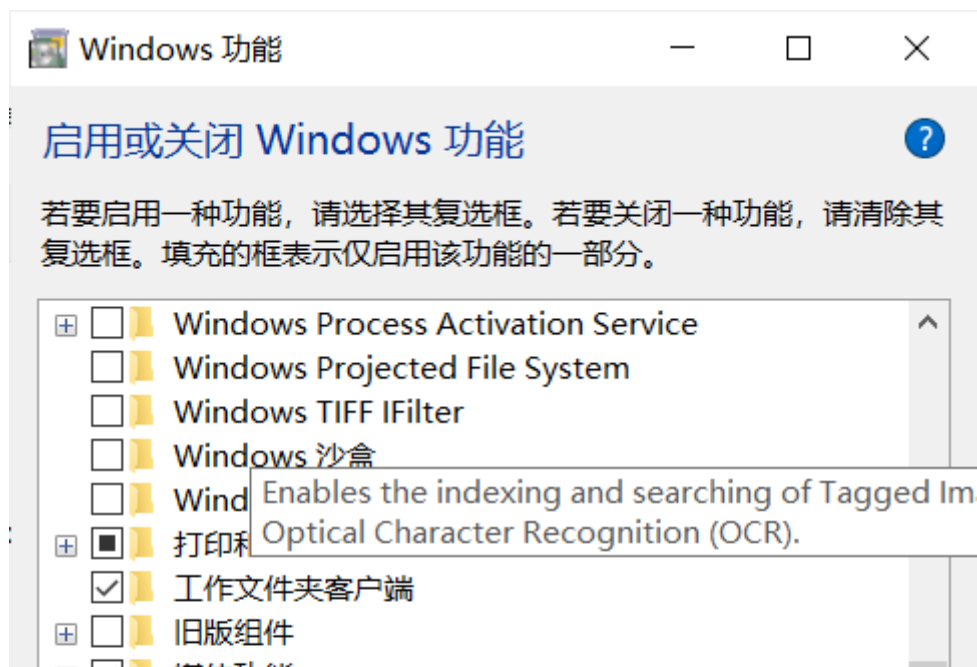
并不是所有windows都支持WSL，需要确保系统版本在Windows 10 Build 16237或更新版本。可以通过【设置】查看系统版本号。

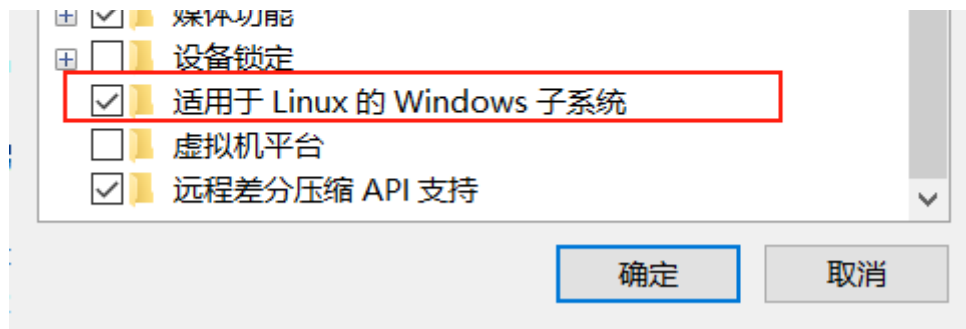


## 開啟WSL服務

需要先在Windows中啟用“適用於Linux的Windows子系統”。

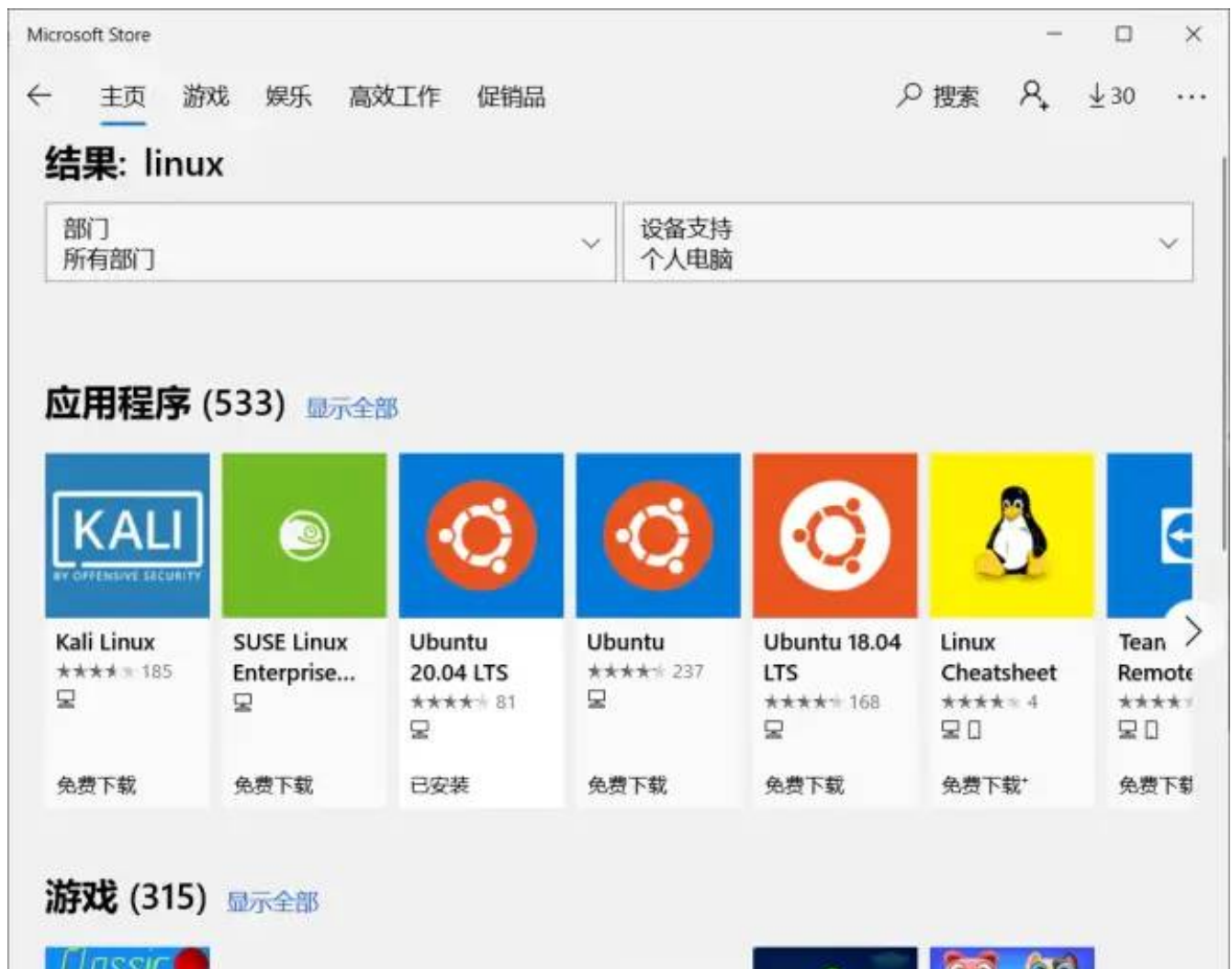
打開“控制面板”->“程序和功能”，在左側欄選擇“啟用或關閉Windows功能”，在打開的對話框中找到“適用於Linux的Windows子系統”並打勾，點擊【確定】關閉對話框，等待系統安裝新功能完成，並按提示重啟電腦。





## Microsoft Store安装WSL

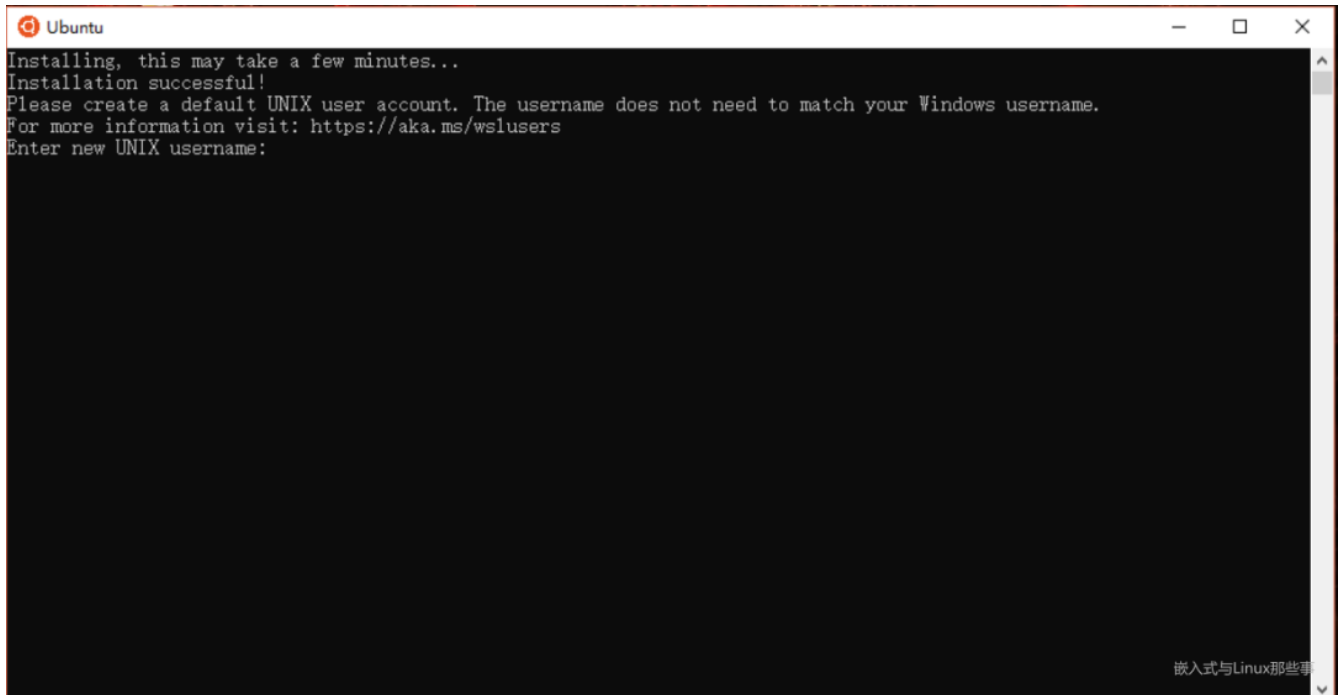
在微软应用商店搜索Linux，可以看到一系列Linux 发行版，根据自己需要选择适合自己的发行版，这里我选用Ubuntu 20.04，下载完成后启动，等待安装完成，输入账户和密码，我们便得到了一个Linux 环境了。



## WSL基本设置

### 配置用户名和密码

在開始菜單中打開Ubuntu 後，Ubuntu 會進行較長時間的安裝和初始化，之後會提示你設置Linux 的用戶名和密碼。



## 更換源

### 更換/etc/apt/sources.list文件裡的源

安裝好ubuntu20.04後，默認的軟件更新源是國外的，在國內使用速度很慢，需要更換成國內的源，這樣才能正常安裝和更新軟件。

### 備份源列表

Ubuntu配置的默認源並不是國內的服務器，下載更新軟件都比較慢。首先備份源列表文件 **sources.list**：

```
# 首先备份源列表
sudo cp /etc/apt/sources.list /etc/apt/sources.list_backup
```

### 打開sources.list文件修改

選擇合適的源，替換原文件的內容，保存編輯好的文件, 我們選擇的是中科大源。

```
# 打开sources.list文件
sudo vim /etc/apt/sources.list
```

編輯/etc/apt/sources.list文件, 在文件最前面添加阿里雲鏡像源：

# 中科大源

```
deb https://mirrors.ustc.edu.cn/ubuntu/ focal main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal main restricted universe multiverse
deb https://mirrors.ustc.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.ustc.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.ustc.edu.cn/ubuntu/ focal-security main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-security main restricted universe multiverse
deb https://mirrors.ustc.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
deb-src https://mirrors.ustc.edu.cn/ubuntu/ focal-proposed main restricted universe multiverse
```

# 添加阿里源

```
deb http://mirrors.aliyun.com/ubuntu/ focal main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-security main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-security main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-updates main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-updates main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-proposed main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-proposed main restricted universe multiverse
deb http://mirrors.aliyun.com/ubuntu/ focal-backports main restricted universe multiverse
deb-src http://mirrors.aliyun.com/ubuntu/ focal-backports main restricted universe multiverse
```

# 添加清华源

```
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ focal-security main restricted universe multiverse
```

## 刷新列表

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install build-essential
```

## 安裝SSH服務

默認情況下，首次安裝Ubuntu時，不允許通過SSH進行遠程訪問。

在Ubuntu上啟用SSH非常簡單。以root用戶或具有sudo特權的用戶執行以下步驟：

```
sudo apt update
sudo apt install openssh-server
```

安裝完成後，SSH服務將自動啟動。可以通過鍵入以下命令來驗證SSH是否正在運行：

```
sudo service ssh status
```

輸出顯示該服務正在運行，並已啟用以在系統引導時啟動：

```
* sshd is running
```

現在，就可以通過SSH連接到Ubuntu系統。

## 啟用SSH登陸

使用 `cp` 命令將SSH 相關配置文件進行備份：

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
```

使用vim 編輯器編輯「sshd\_config」文件：

```
sudo vim /etc/ssh/sshd_config
```

```
Port 8022 ( 因为 Windows 10 的 SSH 端口已经默认被占用，所以我换成了一个新的端口 )
ListenAddress 0.0.0.0 ( 去掉前面的 # )
PermitRootLogin yes ( 修改成 yes )
PasswordAuthentication yes ( 改成 yes )
```

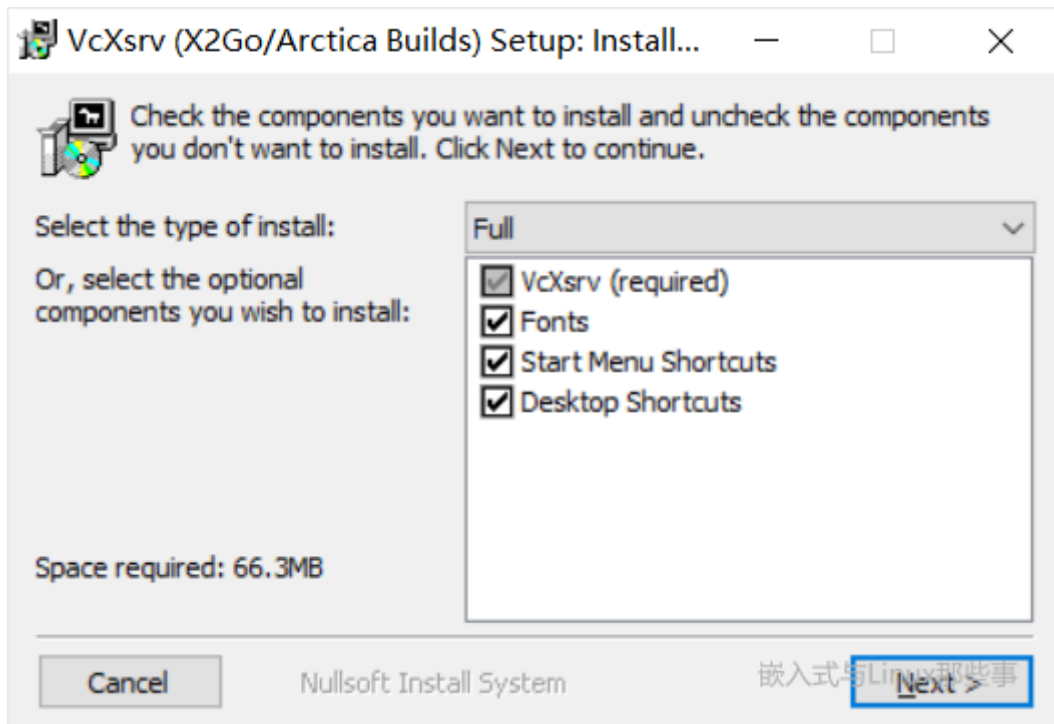
然後輸入命令：`service ssh start` 啟動SSH。

這樣，就可以通過遠程登錄工具登陸WSL了。

## 安裝圖形化界面

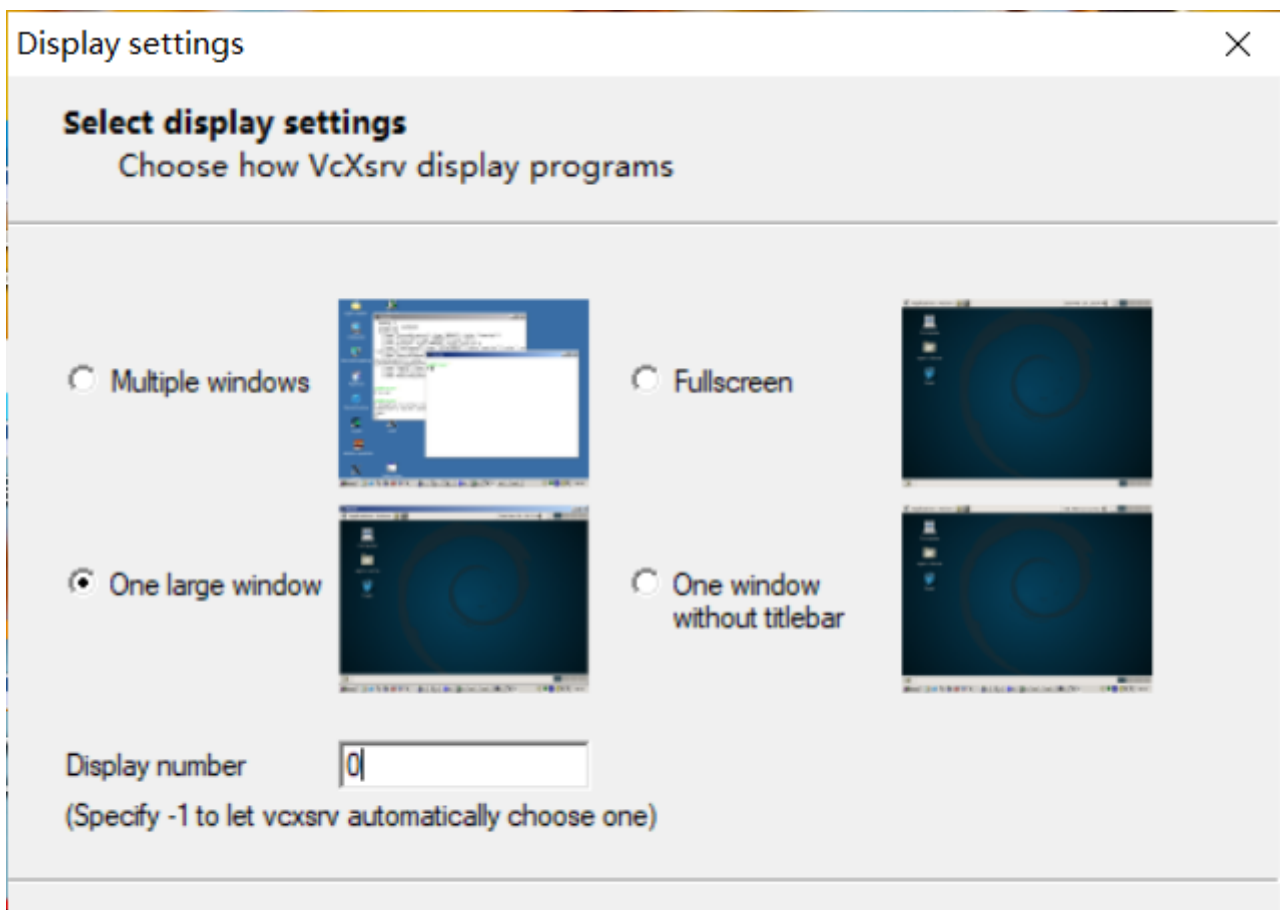
## windows 安裝VcXsrv

WSL目前並不支持圖形界面，首先下載 **VcXsrv**，直接根據安裝程序一直默認安裝即可，可以換一下安裝路徑。



安裝以後會有兩個程序，分別是XLaunch和VcXsrv，可以用來遠程訪問linux。

啟動XLaunch,選擇：“one large window”，Display number設置成0，其它默認即可：





## Ubuntu內安裝桌面環境

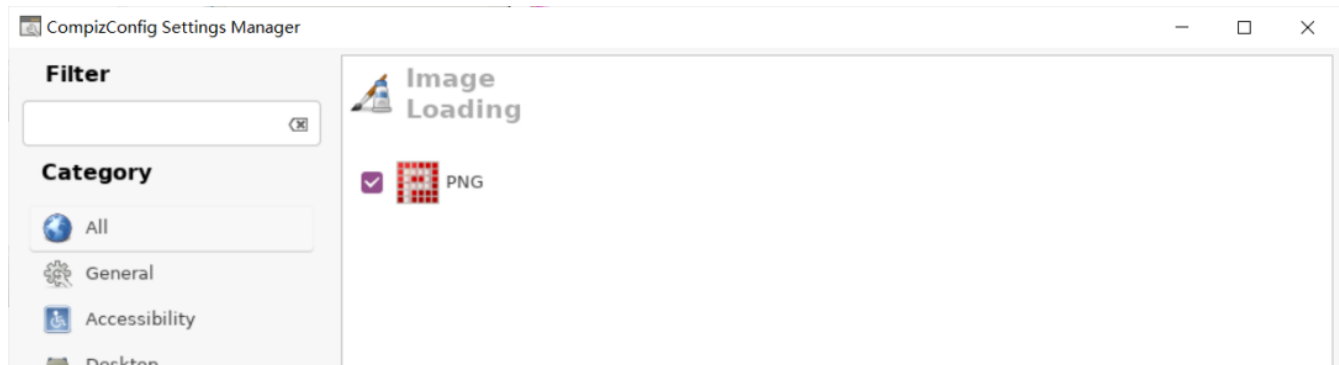
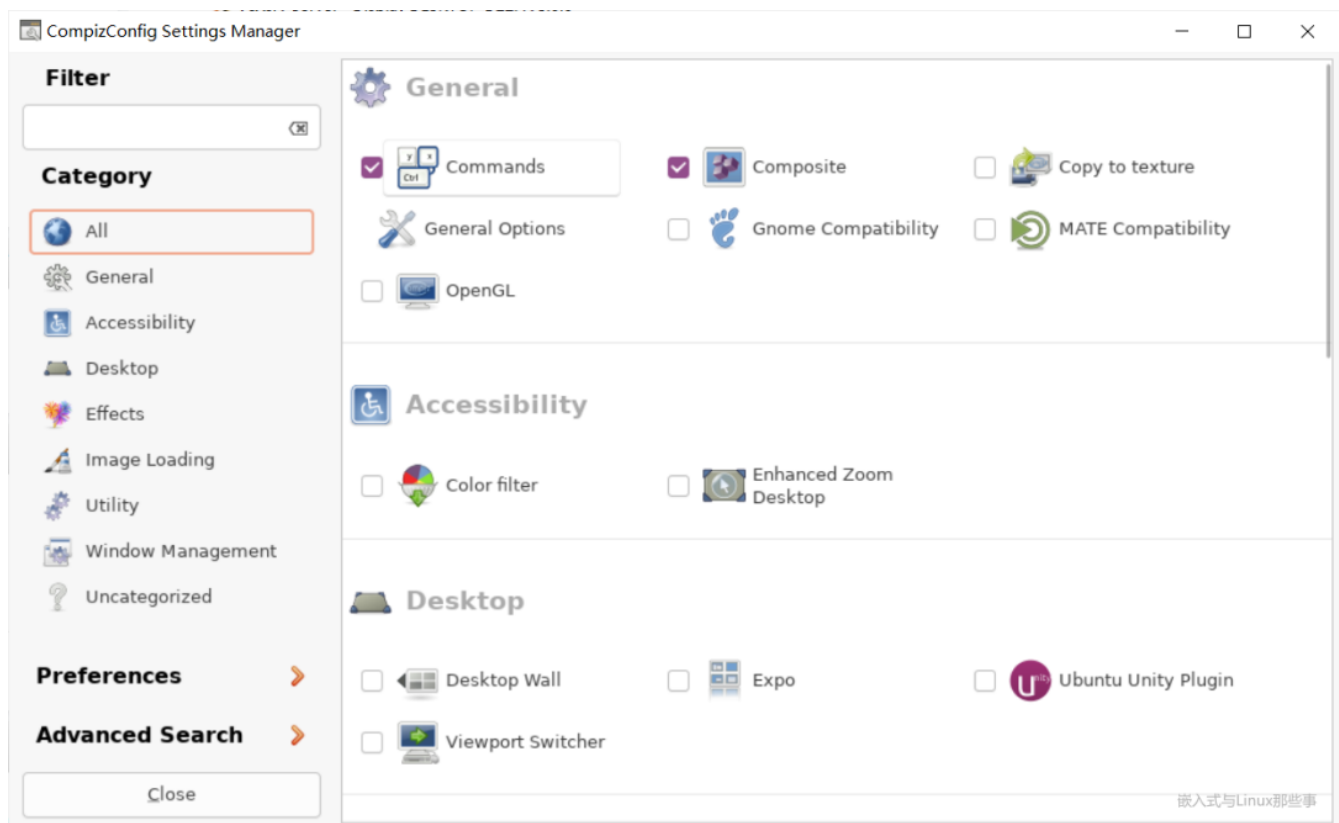
```
sudo apt-get install ubuntu-desktop unity compizconfig-settings-manager  
sudo apt-get install xfce4
```

執行以下指令

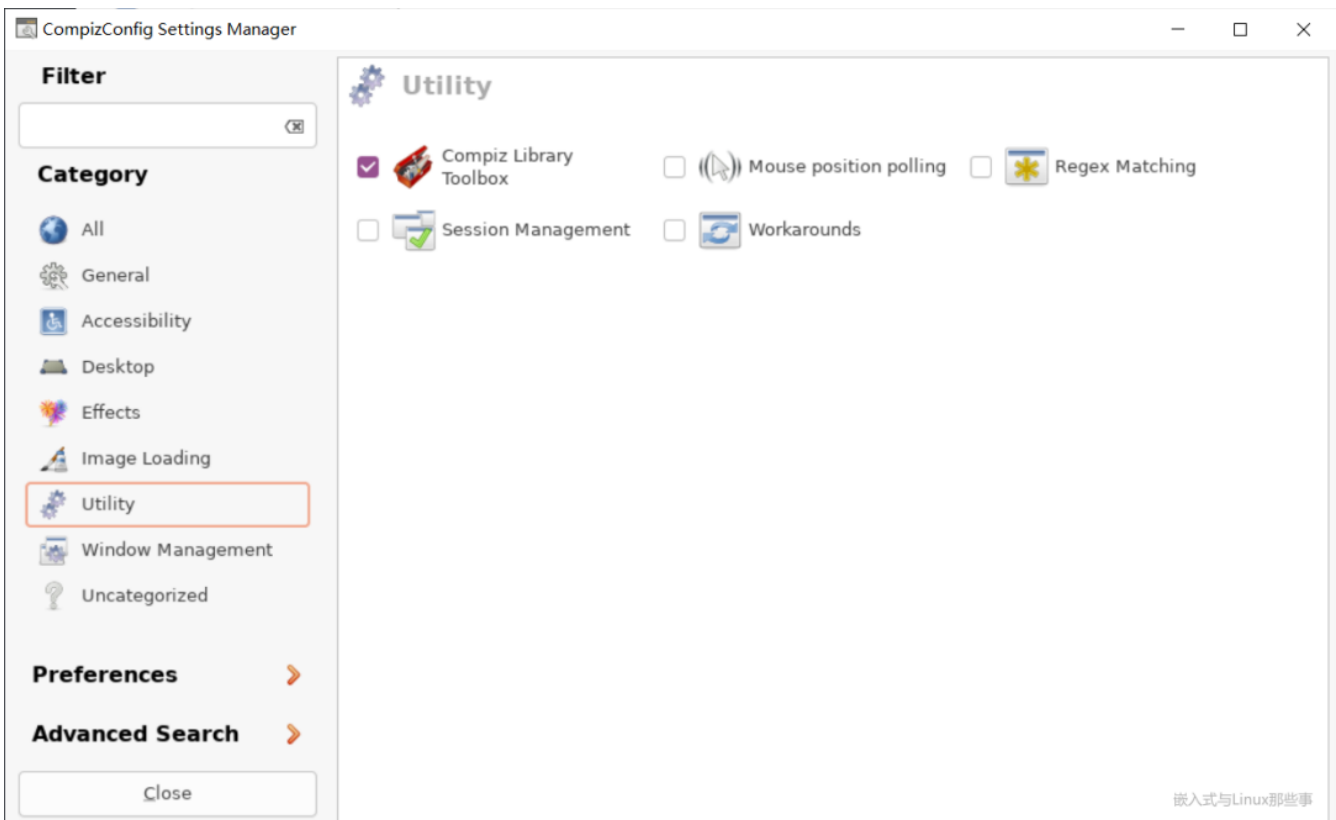
```
export DISPLAY=localhost:0  
ccsm
```

自動彈出ccsm的配置界面。

按照下圖配置，完成後點close即可







將下列命令寫入`~/.bashrc`，每次開機自動執行。

```
export DISPLAY=`cat /etc/resolv.conf | grep nameserver | awk '{print $2}'`:0
```

然後執行 `source ~/.bashrc` 命令即可。

再在WSL中啟動Xfce 會話

```
startxfce4
```

XLaunch中出現如下畫面即為安裝成功。





### 注意事項

在使用`sudo systemctl start ssh` 啟動ssh服務時，系統產生如下報錯：

```
System has not been booted with systemd as init system (PID 1). Can't operate.
```

原因：WSL 使用的是經典的SysV init ( `sysvinit` ) 系統。

可以通過 `ps -p 1 -o comm=` 命令查看當前系統使用哪個init系統。

```
zhongyi@DESKTOP-LHH3HQP:/home$ ps -p 1 -o comm=
init
```

解決方法：不要使用`systemctl`命令，而是使用等同的`sysvinit`命令。

| Systemd command                             | Sysvinit command                          |
|---|---|
| <code>systemctl start service_name</code>   | <code>service service_name start</code>   |
| <code>systemctl stop service_name</code>    | <code>service service_name stop</code>    |
| <code>systemctl restart service_name</code> | <code>service service_name restart</code> |
| <code>systemctl status service_name</code>  | <code>service service_name status</code>  |

```
systemctl enable service_name
```

```
chkconfig service_name on
```

```
systemctl disable service_name
```

```
chkconfig service_name off
```

嵌入式与Linux那些事

--- EOF ---

喜歡此內容的人還喜歡

為什麼不建議把數據庫部署在Docker 容器內？

JAVA葵花寶典

為什麼不建議在Docker 中跑MySQL？

大數據云技術