Linux Desktop installation

This is a general software list for desktop Linux. Everything is tested on the show Distro/desktop environment. Your use may vary on other versions, distros, and desktop environments.

Distro: Xubuntu 22.04

xubuntu is the primary desktop linux that Matt Dumont uses. it is a Ubuntu distro with the xfe desktop environment. details on Xubuntu here

Install from ksl ISO

Matt has made a ksl Iso, which is avalible at: kakapotahi: large_archive/Xubunutu_ISOs/ There are server and desktop versions. Note server versions does not include some of the other key gui software(e.g. qgis)

To install from this file:

1: make installation media

find your usb device (Note this will kill all files on the device). e.g.

```
# find your usb device (Note this will kill all files on the device).
lsblk

# copy the iso to your local machine

isopath=""

# mount the iso
mkdir ~/temp_mount
sudo mount $isopath ~/temp_mount

# make the installible USB
sudo dd if=$isopath of=[path_to_drive_eg_dev_sdb] bs=4M status=progress && sync
```

2: install Xubunutu

insert and follow prompts

3: install other needed software

snap packages cannot be installed into an iso and other configs need to be managed from your user account.

Step 3.1:

```
# todo set your git name and email address
gitname="your name"
gitemail="your email"
```

Step 3.2:

```
echo "export gitname=$gitname" >> ~/.bashrc
echo "export gitname=$gitemail" >> ~/.bashrc
source ~/.bashrc
git clone https://github.com/hansonmcoombs/computer_installs.git ~/Downloads/computer_installs
cd ~/Downloads/computer_installs
bash install_miniconda.sh "$gitname" "$gitemail"
```

Step 3.3: close terminal and open new terminal to allow conda to be accessed:

```
cd ~/Downloads/computer_installs
bash ksl_install.sh "$gitname" "$gitemail"
```

4: set key varaibles

Login to tailscale

```
sudo tailscale up
```

Ensure tailscale worked:

```
tailscale status
```

5: install optional apps

Spotify (music player)

```
sudo snap install spotify
```

Strechly (break reminder app)

```
sudo snap install stretchly
```

Bitwarden

```
sudo snap install bitwarden
```

Allow remote desktop (GUI) via RDP

```
echo "xfce4-session" | tee .xsession
sudo systemctl restart xrdp
sudo ufw allow from any to any port 3389
```

User specific settings

1: set key varaibles

Tailscale ip addresses (for faster ssh/mapping)

you can check these ip addresses using “tailscale status”

```
echo "export tuke=100.96.70.108" >> ~/.bashrc
echo "export kakapotahi=100.67.75.55" >> ~/.bashrc
```

Set kslgittoken (for easy access to pip installs)

Get the KSL github token at: Github Note that the token will get re-generated every 90 days for security purposes.

```
# github machine user read token for pip install
echo "export kslgittoken=[todo paste token here]" >> ~/.bashrc
```

2: User specific software

Install KSL core / google drive for linux (desktop gui)

1. Go to kslcore and follow the instructions for installing the applet.
2. Install the thunar plugins

Install KSL core / google drive for linux (Command line interface)

Follow the instructions on the linux google drive github repo

Install Zutillo

Follow the instructions at: GitHub - wshanks/Zutilo: Zotero plugin providing some additional editing features

Mount for NAS

```
mkdir ~/nas_mount_point
mkdir ~/nas_mount_point/large_archive
mkdir ~/nas_mount_point/large_working
# sudo mount -t cifs -o username=serverUserName //myServerIpAdress/sharename /mnt/myFolder/
# for example this will work if the kakapotahi name and your username are the same and you have the $kakapotahi ip address saved
sudo mount -t cifs -o username=$USER //$kakapotahi/large_working ~/nas_mount_point/large_working
sudo mount -t cifs -o username=$USER //$kakapotahi/large_archive ~/nas_mount_point/large_archive
```

Setting this up to run on login is acutally somewhat difficult!

Optional: Aw-qt-tag (time tracking and notifcation)

GitHub - hansonmcoombs/activitywatch_manual-tag: Pannel app add on to Activity Watch to tag time and to notify the amount of time worked.

Optional: Remote access to word (winapps)

Details at github repo here.

details:

We use this from step 2 (e.g. remoting into an actual machine rather than a VM on the linux machine)

```
git clone https://github.com/Fmstrat/winapps.git
cd winapps

```

Make the config:

```
nano ~/.config/winapps/winapps.conf
```

paste in (ctrl+ shift + v)

```
RDP_USER="MyWindowsUser"
RDP_PASS="MyWindowsPassword"
#RDP_DOMAIN="MYDOMAIN"
RDP_IP="IP of the machine"
#RDP_SCALE=100
#RDP_FLAGS=""
#MULTIMON="true"
#DEBUG="true"
```

3: Mange system prefrences

set keyboard shortcuts

- □ ctrl + alt + delete → xfce4-taskmanager
- □ ctrl+alt+shift+l → xflock4
- □ remove ctrl + alt + l = xflock4 (to allow pycharm refactor)

Set qimgv to default app for .png, .jpg

1. super key → default applications.
2. search for image/png → qimgv
3. seach for image/jpeg → qimgv

Set Auto start

1. super key → session and startup
2. new ➕ →
    a. Name = flameshot
    b. command = /bin/flameshot
    c. trigger= on login

Change sleep time prevent lock out in Zoom meetings

1. super key → power manager
2. display
3. blank after = 10 min
4. put to sleep = >10 (15 min)
5. super key → xfce screensaver
6. enable screensaver
7. lockscreen → lock screen with screensaver 2 min

Configure pannel

1. right click on pannel → pannel → preferences
2. items
3. windows buttons (dbl click) → behaviour → windows grouping = always ; sorting order group title and window title
4. I recommend adding the “system load monitor”

Toggle touchpad via keyboard

Matt tends to turn of his touchpad when he has a mouse (so he doesn’t hit it). This can cause problems…

To create a keyboard shortcut to toggle touchpad:

```
# find your device name:
xinput list
# find the name e.g., "Logitech MX Vertical"
inputdevicename='"[device name]"'

echo "xinput enable "$inputdevicename"" >> ~/disable_touchpad.sh
# enable first becaue there is some weirdness more info:
# https://www.linux.org/threads/re-enabling-touchpad-with-xinput-enable-13-fails.23966/#post-209109
echo "xinput disable "$inputdevicename"" >> ~/disable_touchpad.sh
echo "xinput enable "$inputdevicename"" >> ~/enable_touchpad.sh

```

To run either set bash run the sh scripts or add it to menu via “menu editor”

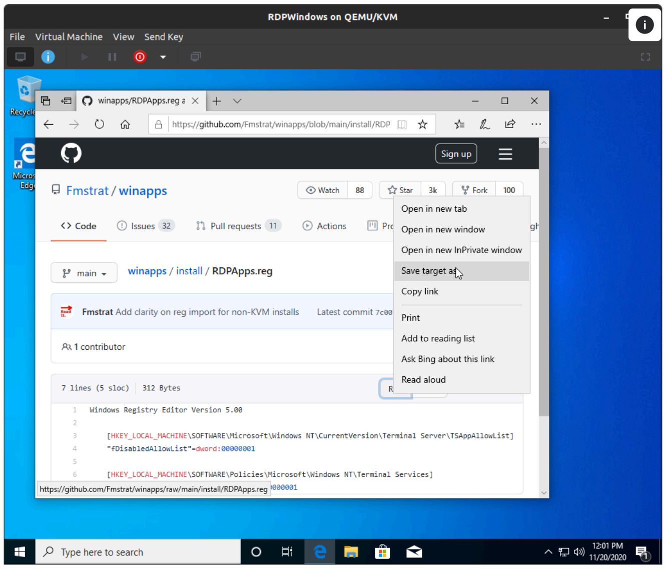
```
bash [homedir]/enable_touchpad.sh
```

```
bash [homedir]/disable_touchpad.sh
```

Setting up windows to allow winapps 🔗

Note if you need to setup a new desktop to host the windows vm you will need to run:

Once you finish the driver install, you will need to make some registry changes to enable RDP Applications to run on the system. Start by downloading the `RDPApps.reg` file from the WinApps repo by visiting https://github.com/Fmstrat/winapps/blob/main/install/RDPApps.reg , right clicking on the `Raw` button, and clicking on `Save target as`.

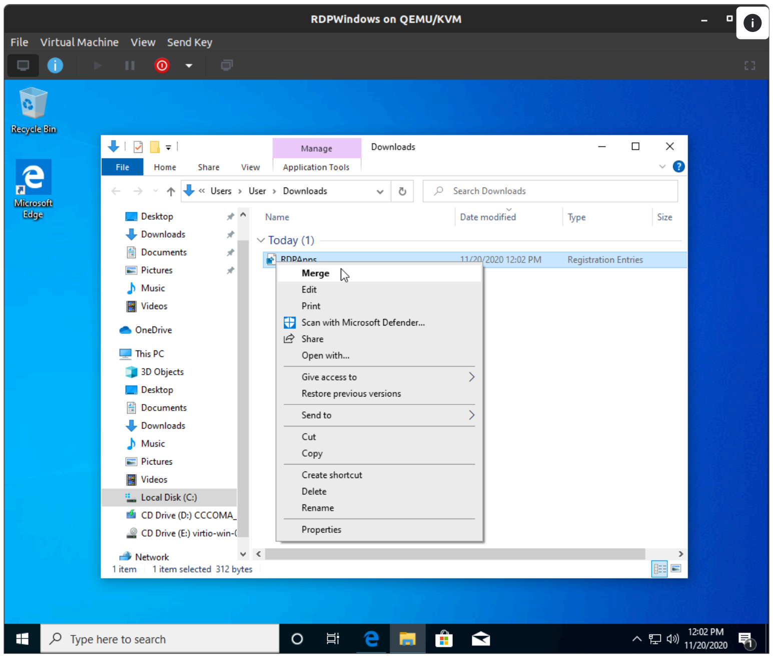


```
Windows Registry Editor Version 5.00

    [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Terminal Server\TSAppAllowList]
    "fDisabledAllowList"=dword:00000001

    [HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows NT\Terminal Services]
```

Once you have downloaded the registry file, right click on it, and choose `Merge`, then accept any confirmations along the way.



more details and details to set up a windows vm at https://github.com/Fmstrat/winapps/blob/main/docs/KVM.md

Creating a DO droplet from ISO:

digital ocean does not support droplets from an ISO. easiest option is to create a base droplet by:

Step 1: Create a ubuntu droplet

you will need at least 20gb of storage

Step 2: Install software

IN computer_installs/create_iso at main · hansonmcoombs/computer_installs

Step 3: install user standard software (DO version)

Step 3.1:

```
# todo set your git name and email address
gitname="your name"
gitemail="your email"
```

Step 3.2:

```
echo "export gitname=$gitname" >> ~/.bashrc
echo "export gitname=$gitemail" >> ~/.bashrc
source ~/.bashrc
git clone https://github.com/hansonmcoombs/computer_installs.git ~/Downloads/computer_installs
cd ~/Downloads/computer_installs
bash install_miniconda.sh "$gitname" "$gitemail"
```

Step 3.3: close terminal and open new terminal to allow conda to be accessed:

```
cd ~/Downloads/computer_installs
bash ksl_install_DO.sh "$gitname" "$gitemail"
```

Step 4: Run through User specific settings

Step 5: save a “snapshot”

this allows you to quickly spin up new DO droplets. If you have a project specific project I would suggest that you make a project specific snapshot. Talk to matt for more info.

Add other users

1: create user (from your profile)

```
newusername="add a name here"
sudo useradd -m $newusername
sudo passwd $newusername
sudo chsh -s /bin/bash $newusername # switch to bash
sudo usermod -aG sudo $newusername  # make a sudo user
```

2: install user specific software/settings (from new profile)

Step 2.1:

```
# todo set your git name and email address
gitname="your name"
gitemail="your email"
```

Step 2.2:

```
git clone https://github.com/hansonmcoombs/computer_installs.git ~/Downloads/computer_installs
cd ~/Downloads/computer_installs
bash ksl_new_user.sh "$gitname" "$gitemail"
```

3: Run through User specific settings

Stress test

Perform GPU, CPU, and I/O stress testing on Linux

1. open 4 terminals
2. Terminal 1:

   ```
   watch -n 0.5 sensors
   ```

3. Terminal 2:

   ```
   htop
   ```

4. terminal 3:

   ```
   watch -n 0.5 "grep 'cpu MHz' /proc/cpuinfo"
   ```

5. Terminal 4:

```
nproc
stress --cpu 8 --io 4 --vm 2 --vm-bytes 128M --timeout 10s
```

Making the ISO file:

1. Download the latest xubuntu ISO
2. install cubic if you have not already: GitHub - PJ-Singh-001/Cubic: The Official Web Site for Cubic (Custom Ubuntu ISO Creator) (https://github.com/PJ-Singh-001/Cubic)
3. use cubic (additional instructions here) and the create_iso text file to generate the ISO