Setting up EndeavourOS

It is Arch-based.

Stuff I did after login:

- ran `nvidia-smi` to make sure GPU was working: it was.
- enable bluetooth `systemctl enable bluetooth.service –now`

Keyboard Setting

Caps lock should be an additional Control key, sorry, it's true. Add `XKBOPTIONS="caps:ctrl_modifier"` to `/etc/default/keyboard`.

Note About AUR

AUR is a collection of user-supplied build scripts for different programs. It's like EPEL for Redhat or PPAs for Debian (or Ubuntu? I forget which). AUR packages aren't binaries so you have to compile them, but it's not too painful because you mostly just run `makepkg -s` and then `pacman -U name_of_freshly_built_package` to install.

I don't know if there's a generally accepted way to keep track of downloaded AUR packages, so I made a folder in my `Downloads` folder, `aur`, and create a new subfolder for each package. Sometimes packages (like VMWare) have dependencies on other AUR packages, so having a subfolder for each package that I'm trying to install can keep things organized.

Install VMWare Workstation...

... so that I can keep using Affinity Photo. I followed https://wiki.archlinux.org/title/VMware to install VMWare Workstation 17.

```
# make a directory to work in.
mkdir Downloads/aur
cd Downloads/aur/
# vmware needs vmware-keymaps installed first
git clone https://aur.archlinux.org/vmware-keymaps.git --depth=1
cd vmware-keymaps/
makepkg -s
sudo pacman -U vmware-keymaps-1.0-3-any.pkg.tar.zst
cd ..
# Build vmware workstation package
```

```
git clone https://aur.archlinux.org/vmware-workstation.git --depth=1
makepkg -s
sudo pacman -U vmware-workstation-17.5.1-2-x86_64.pkg.tar.zst
# I couldn't get vmw_vmci module to work until I rebooted.
sudo reboot
# activate vmware kernel modules
sudo modprobe vmmon vmw_vmci
# start services
sudo systemctl enable vmware-usbarbitrator.service  vmware-networks.service --
now
```

Configuration Tweaks

None yet

Packages I needed

micro

Remote Desktop Setup

Had to use AUR packages to install `xrdp`

```
cd ~/Downloads/aur
mkdir xrdp
cd xrdp
git clone https://aur.archlinux.org/xrdp.git
cd xrdp
makepkg -s
sudo pacman -U xrdp-0.10.0_beta.3-1-x86_64.pkg.tar.zst
cd ..
git clone https://aur.archlinux.org/xorgxrdp.git
cd xorgxrdp
makepkg -s
sudo pacman -U xorgxrdp-0.10.0-5-x86_64.pkg.tar.zst
# Edited the "port" option in /etc/xrdp/xrdp.ini to only listen to single IP.
port=tcp://secret.ip.address.here:3389
```

Also had to add `wg-quick@2g0.service` to the `After=` line of `/usr/lib/systemd/system/xrdp.service`, otherwise it starts before wireguard and fails to bind to the non-existent wireguard IP address.

Gaming Setup

- Install wine, winetricks, mono-wine, wine-gecko, and lutris from system packages.
- Install gamescope from system packages.
- For Epic Games
  - Used Lutris to install Epic Games Store
  - Note: Fortnite does not run on Linux because Epic won't fix their bundled EAC even though other EAC games run fine.

Tweaks

- Duck Game
  - Set monitor to 1920×1080 or
  - Use gamescope through the steam launcher: `gamescope -h 1080 -w 1920 -f -- %command%`
- Elden Ring
  - Set monitor to 1920×1080
  - Set Proton version to Experimental
  - gamescope -W 1920 -H 1080 -f – %command%
- Helldivers II
  - Initially couldn't get past the final option screen.
  - **BEST:** On reddit, some ppl say set the command to `%command% --use-d3d11` and that worked great, even when I set the resolution back to 1920×1080
  - (You can ignore the following bullet points) Found that other people had luck doing following things:
    - Set compatibility to Proton 8.0-5
    - Edited `.local/share/Steam/steamapps/compatdata/553850/pfx/drive_c/users/steamuser/AppData/Roaming/Arrowhead/Helldivers2/user_settings.config`:
      - Changed render_resolution to 1280 x 800
      - Changed screen_resolution to 1280 x 800
    - Works. I might try GameScope to see if that helps the resolution.

Network Stuff

Set up wireguard.

Need to open a port for other wg endpoint to access service. Learning to use firewalld.

```
# as root
# Create a zone for the wg
firewall-cmd --permanent --new-zone=wgzero
firewall-reload
```

```
# Check if it worked, should see wgzero in the list
firewall-cmd --get-zones
# Add the wg0 interface to the zone
firewall-cmd --zone=wgzero --add-interface=wg0 --permanent
firewall-reload
# Check if the interface is active on wg0
firewall-cmd --get-active-zones
# Add a rule for our port
firewall-cmd --zone=wgzero --add-port=99999/tcp --permanent
firewall-cmd reload
# Your thing should work now!
```

XRDP Sessions

To get IceWM to run automatically I modified its system-wide config because I didn't want a ~/.xinitrc file messing up my local KDE sessions. I just had to comment out most of startwm function and add icewm-session.

Here's the modified file

.

IceWM Notifications

I use a misbehaving app called Stability Matrix that, as of version 2.10.3, crashes if it can't send a desktop notification. My IceWM didn't have notifications out-of-the-box so I installed `dunst` which is a lightweight notification app.

Launcher

IceWM doesn't have a searchable app launcher (think quicksilver on MacOS, the start menu on Windows, or KDE's start menu equivalent). I installed `dlauncher` to handle that in IceWM.

IceWM Config Files

To make these programs work I made two files.

`~/.icewm/startup`

```
#!/bin/bash
dlauncher &
dunst &
```

and `~/.icewm/keys`

```
key "Ctrl+Shift+space" dlauncher-toggle
```

Duplicati

I've been using Duplicati for years on Windows and Mac. Setting up the client:

```
cd ~/Downloads/aur
mkdir duplicati
cd duplicati
git clone https://aur.archlinux.org/duplicati-canary-bin.git
cd duplicati-canary-bin
makepkg -s  # accept requirements installation
sudo pacman -U duplicati-canary-bin-2.0.7.103-1-x86_64.pkg.tar.zst

# Run the tray program through mono:
mono /opt/duplicati/Duplicati.GUI.TrayIcon.exe
```

Nvidia Power Reduction

I prefer cool and quiet. In Linux you can reduce the allowed power draw with the nvidia-smi command.

To print the current power info:

```
sudo nvidia-smi -q -d POWER
```

On my 4060, the default power limit is 165 watts. I'm going to drop it 145W with this command

```
sudo nvidia-smi -pl 145
```

To do this automatically when the computer boots, I created a file `cat /etc/systemd/system/nvpower.service`:

```
[Unit]
Description=Set NVIDIA GPU Power Limit

[Service]
Type=oneshot
ExecStart=/usr/bin/nvidia-smi -pl 145

[Install]
WantedBy=multi-user.target
```

And enabled it with: <code bash> sudo systemctl enable nvpower.service </bash>