

# 4- Node Raspberry Pi Cluster

By: Laine Taylor, Gabriel Wakefield, Keegan  
Carter, Bailey Carpenter



# Background

- Raspberry pi's are a cheap and simple option to learn basic coding and programming
- A Raspberry pi cluster involves combining multiple raspberry pi's to share the load of heavier task, like running a complicated server, managing multiple different aspects of a smart home, etc
  - not for gaming



# Background

- Generally, one Raspberry Pi (or node) is set up to be the what's called a "master controller" while the other ones are set up as "workers."
- the master controller is told what to do specifically and dishes out the duties to the others



# Materials

- Casing
- 4 raspberry pi's
- 4 ethernet cables
- Ethernet hub
- Fans
- Desktop monitor
- HDMI cables



# Process

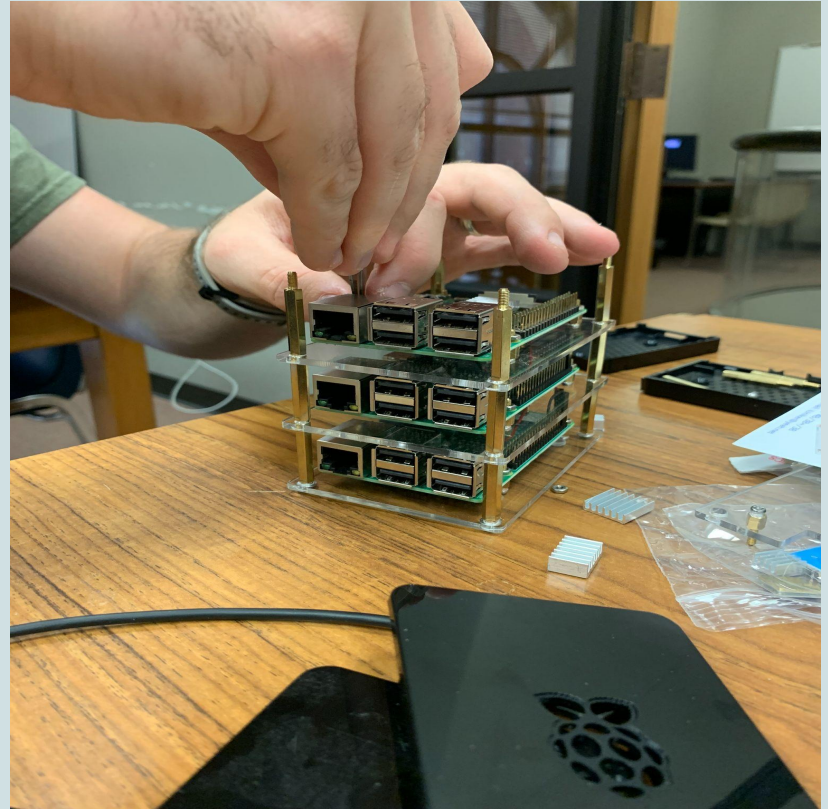
The first step was the assemble the fans onto the casing





## Process

Next we assembled the Raspberry pi's so that they are situated on top of one another



## Process

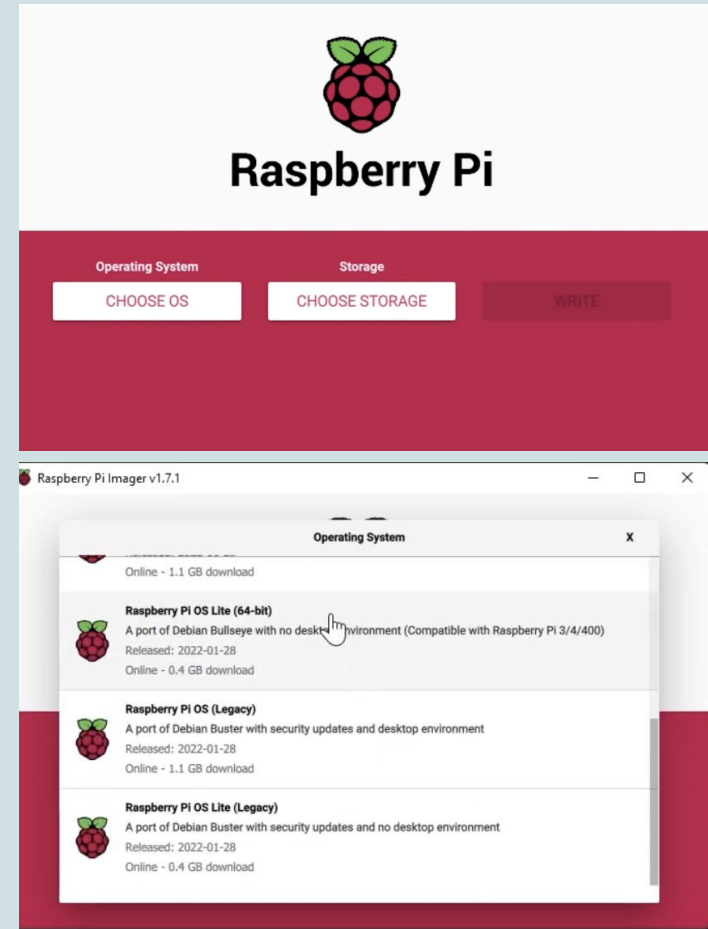
After all nodes are assembled, the raspberry cluster is ready to be connected to the desktop monitor for synchronization.



# Programming

Now comes the most difficult part:

- raspberry pi imager was installed to burn the operating system onto the SD cards.
  - there were a few different options to choose from, but we ultimately chose the **64-bit Raspberry OS Lite** (this would give a faster download and we had no need of the GUI desktop)







# Programming

- Each node was given a name, booted up and a code was run to get their IPs
  - **sudo ip a**
- the following commands were given to open and update a secure shell (SSH) so the nodes could be accessed from a terminal of another computer
  - ssh <username>@>ip address
  - sudo apt update
  - sudo apt upgrade -y

# Programming

- After updating a few files already in each node, K3s was installed on each node with the following command:
  - `curl -sfL https://get.k3s.io | sh -`
- after a reboot, the following the following command was added to check the success of the installation
  - `sudo kubectl get nodes`

```
Wi-Fi is currently blocked by rfkill.
Use raspi-config to set the country before use.

fishy@pim:~$ curl -sfL https://get.k3s.io | sh -
[INFO] Finding release for channel stable
[INFO] Using v1.22.7+k3s1 as release
[INFO] Downloading hash https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/sha256sum-arm64
[INFO] Downloading binary https://github.com/k3s-io/k3s/releases/download/v1.22.7+k3s1/k3s-arm64
[INFO] Verifying binary download
[INFO] Installing k3s to /usr/local/bin/k3s
[INFO] Skipping installation of SELinux RPM
[INFO] Creating /usr/local/bin/kubectl symlink to k3s
[INFO] Creating /usr/local/bin/crictl symlink to k3s
[INFO] Creating /usr/local/bin/ctr symlink to k3s
[INFO] Creating killall script /usr/local/bin/k3s-killall.sh
[INFO] Creating uninstall script /usr/local/bin/k3s-uninstall.sh
[INFO] env: Creating environment file /etc/systemd/system/k3s.service.env
[INFO] systemd: Creating service file /etc/systemd/system/k3s.service
[INFO] systemd: Enabling k3s unit
Created symlink /etc/systemd/system/multi-user.target.wants/k3s.service → /etc/systemd/system/k3s.servi
[INFO] systemd: Starting k3s
fishy@pim:~$ sudo kubectl get nodes
NAME STATUS ROLES AGE VERSION
pim Ready control-plane,master 73s v1.22.7+k3s1
fishy@pim:~$
```

*instal k3s*

*Finally!!!!!!!*



# Programming

- master control is done, now on to the workers
  - the same process was repeated for each of the other worker nodes, except instead of the command to install K3s, the following command was inputted to connect the node to the master node

```
curl -sfL https://get.k3s.io | K3S_URL=https://myserver:6443 K3S_TOKEN=mynodetoken sh -
```

*Handwritten annotations:*

- downloads* (with an arrow pointing to `https://get.k3s.io`)
- Server* (with an arrow pointing to `myserver`)
- token* (with an arrow pointing to `mynodetoken`)
- pin IP* (pointing to the IP address `6443`)



# Programming

- We now have Raspberry Pi cluster with 4 nodes (1 master and 3 workers)
  - However one of the nodes were corrupted and there were a few setbacks which made us have to restart multiple times
  - As long as they continue to hold, we will download a program called Rancher that would give us visuals on how powerful this computer actually is, maybe even add a few servers

```
fishy@pim:~ $ sudo kubectl get nodes
NAME      STATUS    ROLES          AGE    VERSION
pi2       Ready     <none>         27m    v1.22.7+k3s1
pi1       Ready     <none>         36m    v1.22.7+k3s1
pim       Ready     control-plane, 51m    v1.22.7+k3s1
master
```

# Programming

- This was the end result after one of the nodes became corrupted and (one of the times) we had to restart







# Programming

- We are hoping to try and make the system self running, where there won't be any need for an outside computer but right now it just wasn't working out
  - We may be able to do this by setting all pi's up on the WiFi and setting one with the GUI desktop interface but when we tried earlier it didn't work
  - But now with the better understanding we have of the process it may be possible



# Works Cited

Wybiral, Davy, director. *Building a 4-Node Raspberry Pi Cluster*. Youtube.com, 27 Jan. 2018, <https://www.youtube.com/watch?v=H2rTecSO0gk>. Accessed 8 Apr. 2022.

<https://www.youtube.com/watch?v=eZ74x6dVYes>

<https://www.seancarney.ca/2020/10/09/quickly-create-polished-professional-reports-with-markdown-and-r/>