STUDY GUIDE FOR MODULE NO. LAB 04

Internet Connection Monitoring using RPi

MODULE OVERVIEW

In the contemporary digital landscape, network performance monitoring emerges as a pivotal aspect of system management. Within this context, the Raspberry Pi, an affordable and compact single-board computer, assumes a noteworthy role. This introduction explores the utilization of Cacti, an open-source network monitoring tool, as an instrumental solution for enhancing network performance oversight on the Raspberry Pi platform. Network performance monitoring involves systematically collecting and analyzing data related to network behavior, providing administrators with valuable insights to optimize resource utilization and ensure efficient infrastructure operation. Leveraging Cacti's web-based interface and the Simple Network Management Protocol (SNMP), the Raspberry Pi becomes a formidable ally in pursuing streamlined and effective network monitoring, applicable to diverse settings such as home networks, small businesses, and educational institutions.

MODULE LEARNING OUTCOMES

By the end of this module, participants should be able to:

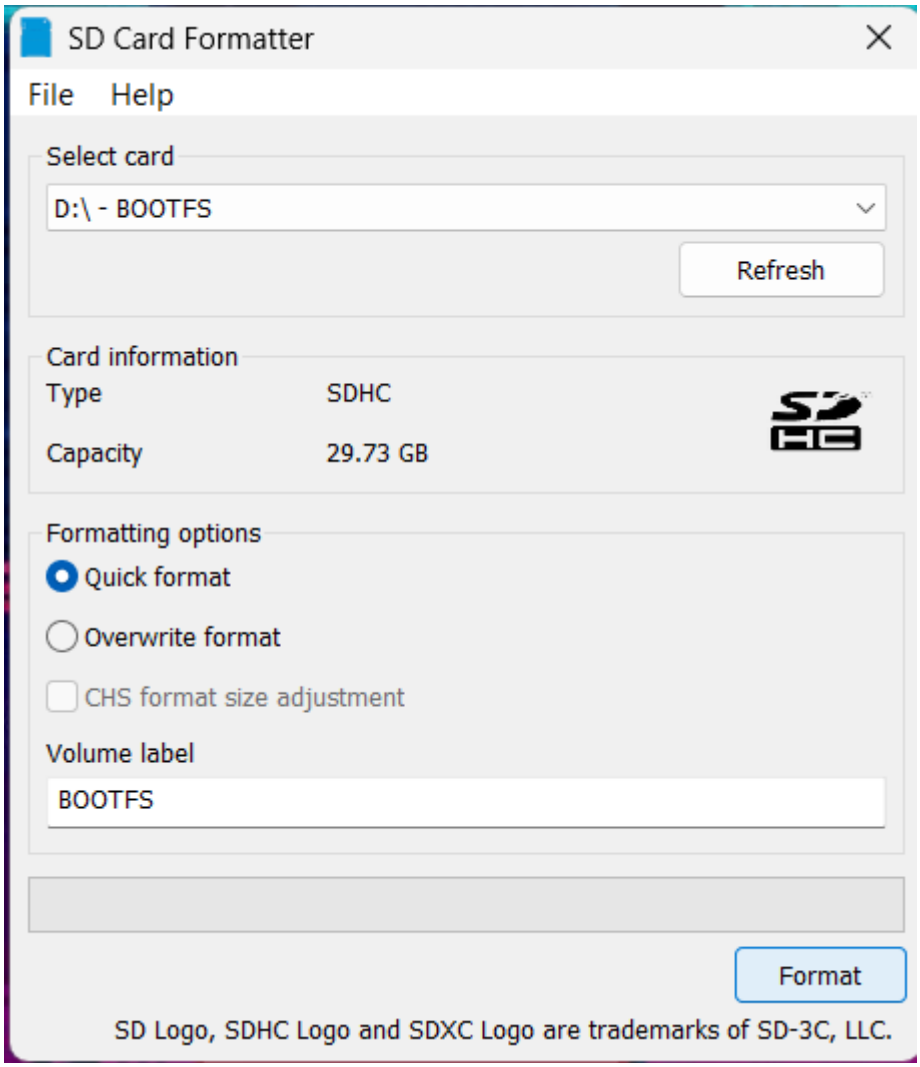
- ✓ Introduction to Internet Connection Monitoring:
  - Define the importance of internet connection monitoring.
  - Understand the significance of reliable internet connectivity for various applications.
- ✓ Overview of Raspberry Pi:
  - Introduce Raspberry Pi as a versatile and cost-effective single-board computer.
  - Explore the hardware components and capabilities of Raspberry Pi.
- ✓ Setting up Raspberry Pi:
  - Install the operating system on Raspberry Pi.
  - Configure network settings for wired or wireless connectivity.
- ✓ Introduction to Network Monitoring:
  - Explain the basics of network monitoring and its applications.
  - Identify common tools and protocols used in network monitoring.
- ✓ Installing Monitoring Software on Raspberry Pi:
  - Demonstrate the installation of monitoring tools (e.g., Ping, Nmap) on Raspberry Pi.
  - Configure the Raspberry Pi to act as a network monitoring node.
- ✓ Scripting for Internet Connection Checks:
  - Introduce scripting languages (e.g., Python) for automating internet connection checks.
  - Develop scripts to periodically check and log internet connectivity status.
- ✓ Data Logging and Visualization:
  - Implement data logging mechanisms to record internet connection status.
  - Explore visualization tools (e.g., Grafana) to create informative dashboards.
- ✓ Alerts and Notifications:
  - Configure alerts for detecting and notifying users about internet connection issues.
  - Implement email or messaging notifications based on predefined conditions.
- ✓ Troubleshooting and Diagnostics:
  - Teach troubleshooting techniques for resolving common internet connection issues.
  - Utilize Raspberry Pi as a diagnostic tool for network-related problems.
- ✓ Security Considerations:
  - Discuss security best practices for internet connection monitoring.
  - Implement measures to secure the monitoring system and data.
- ✓ Integration with Home Automation Systems (Optional):
  - Explore the integration of internet connection monitoring with home automation systems using Raspberry Pi.
  - Demonstrate practical applications for automated responses based on connectivity status.
- ✓ Project Work and Hands-on Exercises:
  - Engage participants in practical exercises to reinforce concepts learned.
  - Provide a hands-on project where participants design and implement an internet connection monitoring system.

LEARNING CONTENT

Name: _Cerujano, Erman Ace M._____ Due date: February 26, 2024

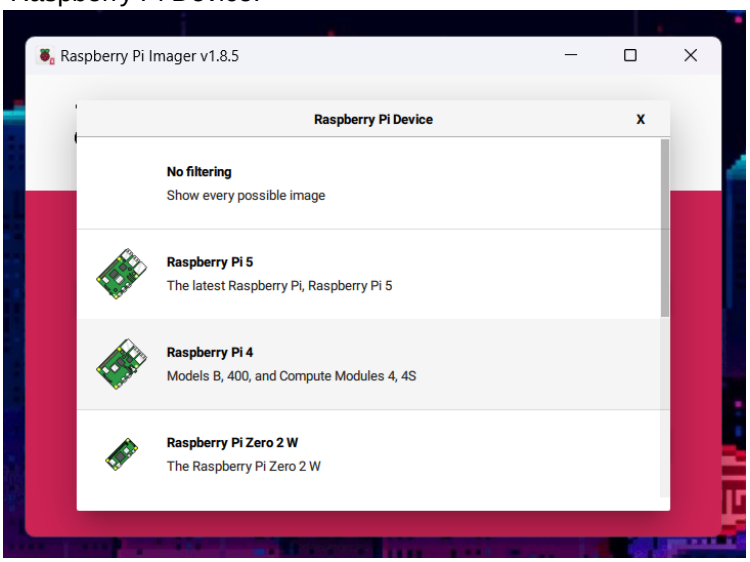
Create a detailed step-by-step method with pictures and detailed description on how you installed a internet monitoring tool in your Raspberry Pi.

Step 1. Format the SD Card in SD Card Formatter.

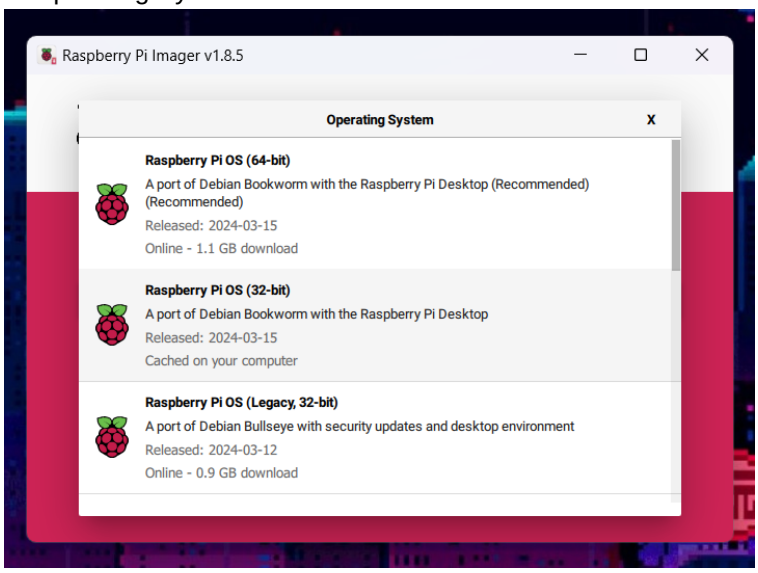


Step 2. Install the OS Raspbian to Raspberry Pi Imager.

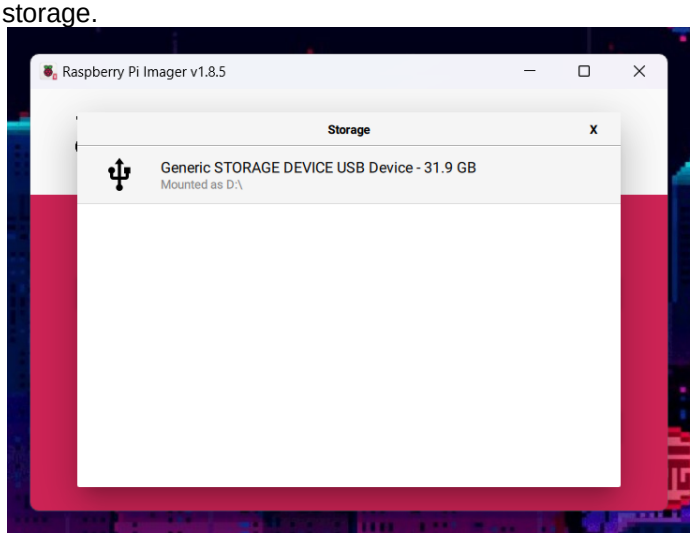
A. Select your Raspberry Pi Device.



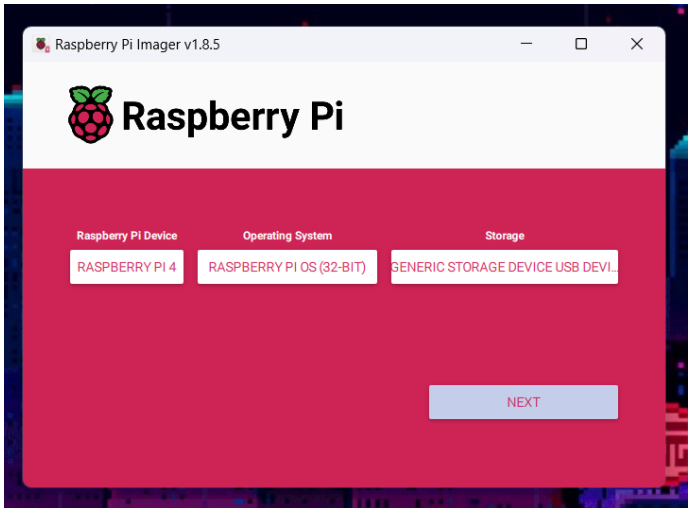
B. Choose your Operating System.



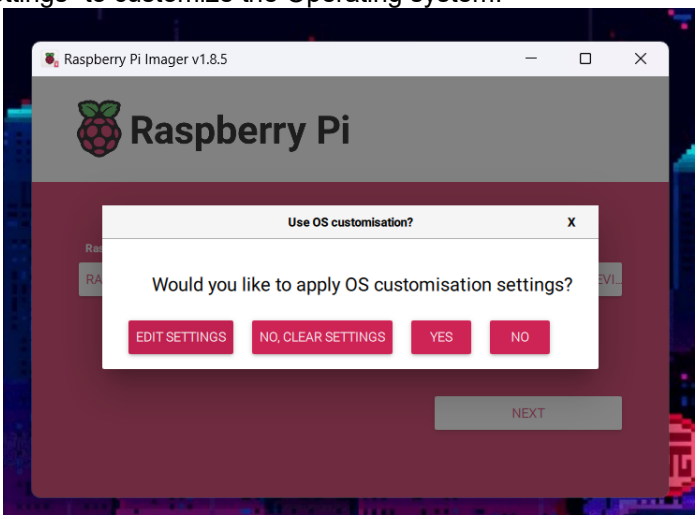
C. Choose your storage.



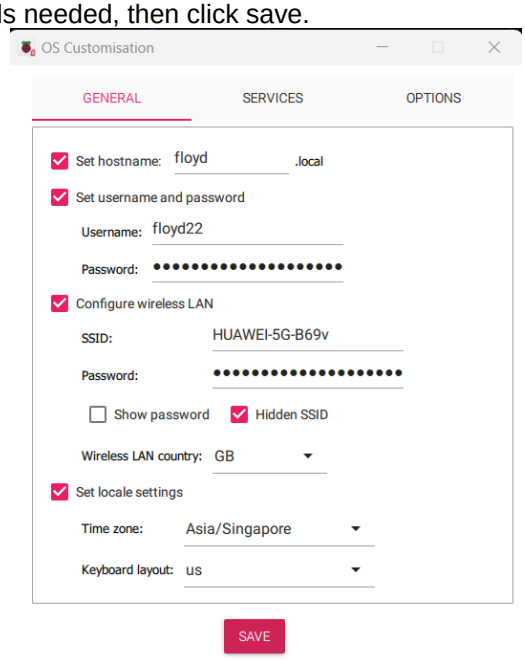
D. Click next to proceed to the installation.



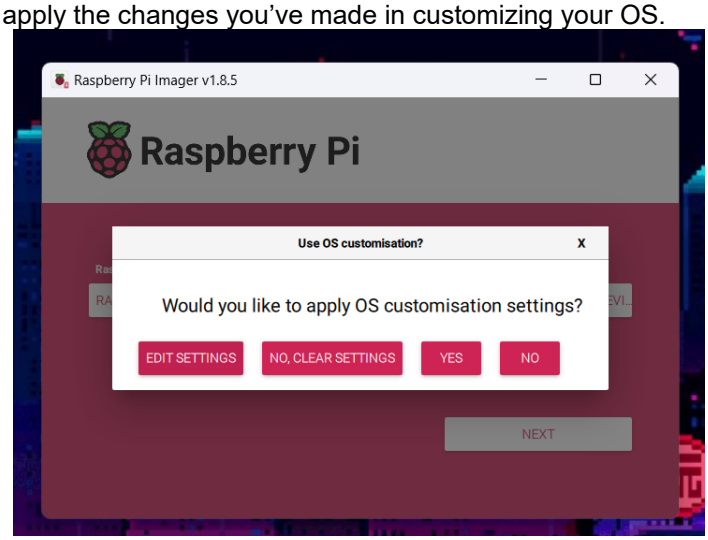
E. Click “Edit Settings” to customize the Operating system.



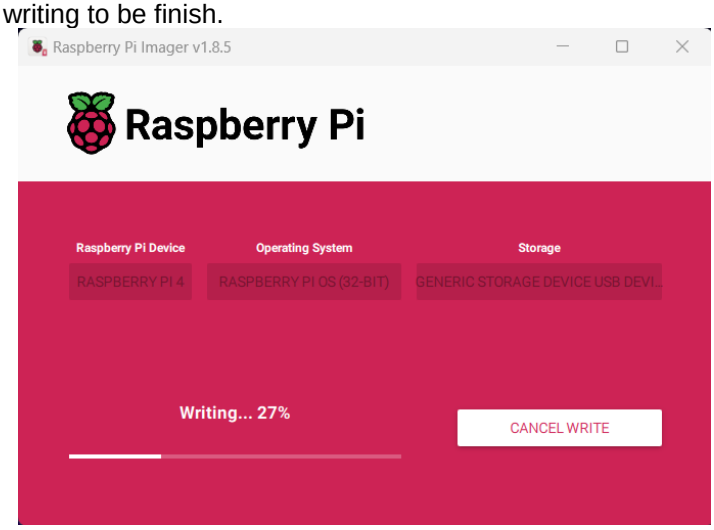
F. Set all the credentials needed, then click save.



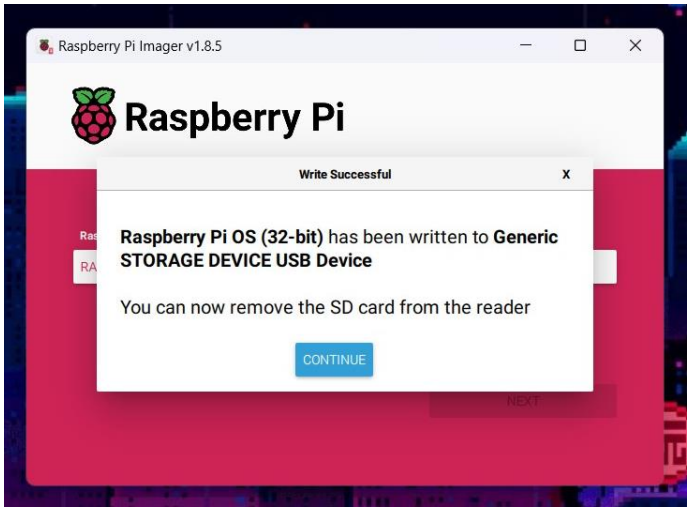
G. Click Yes to apply the changes you've made in customizing your OS.



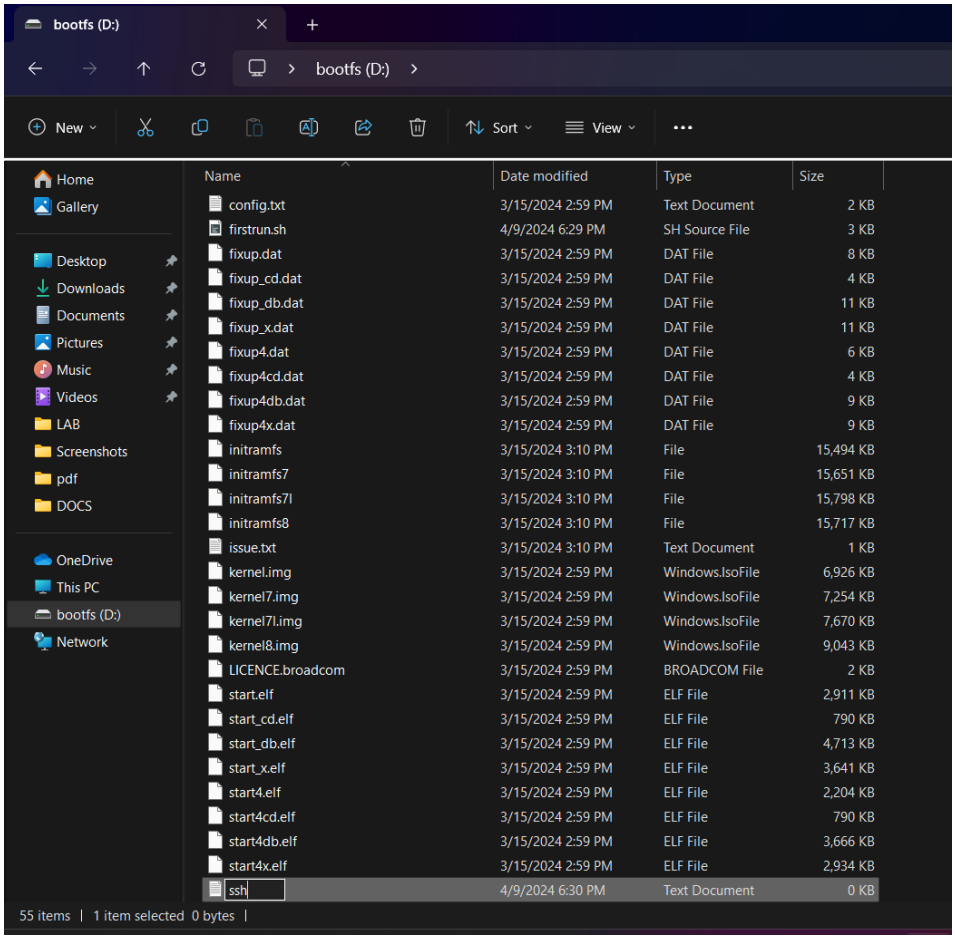
H. Wait for the writing to be finish.



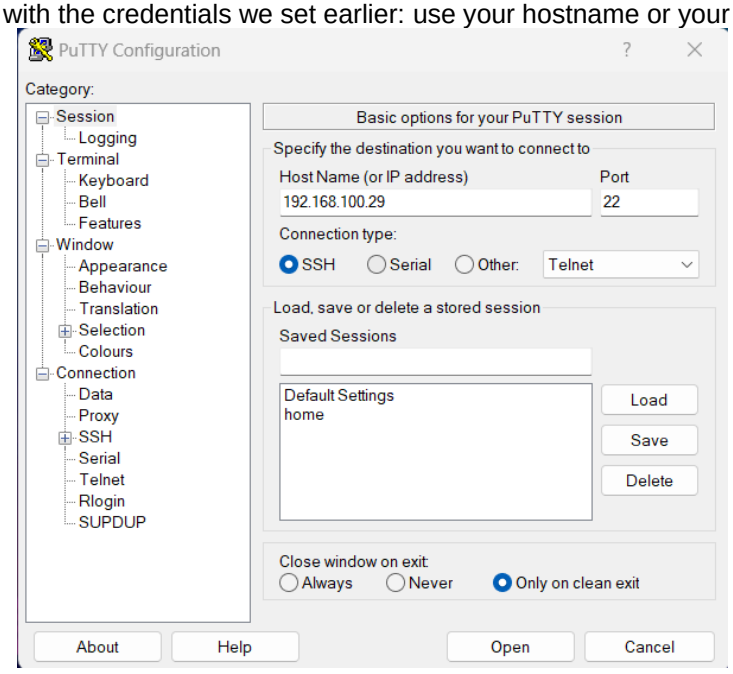
I. After the chosen Raspberry Pi OS has been written to generic storage device, click continue. Then, you can now remove the SD card from the reader.



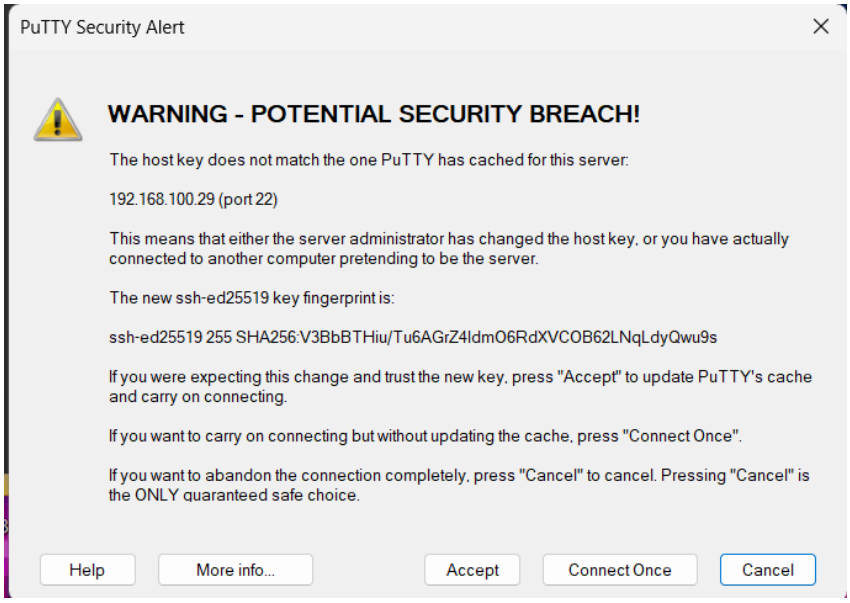
Step 3. Add ssh empty file inside the Storage of D:\ drive.

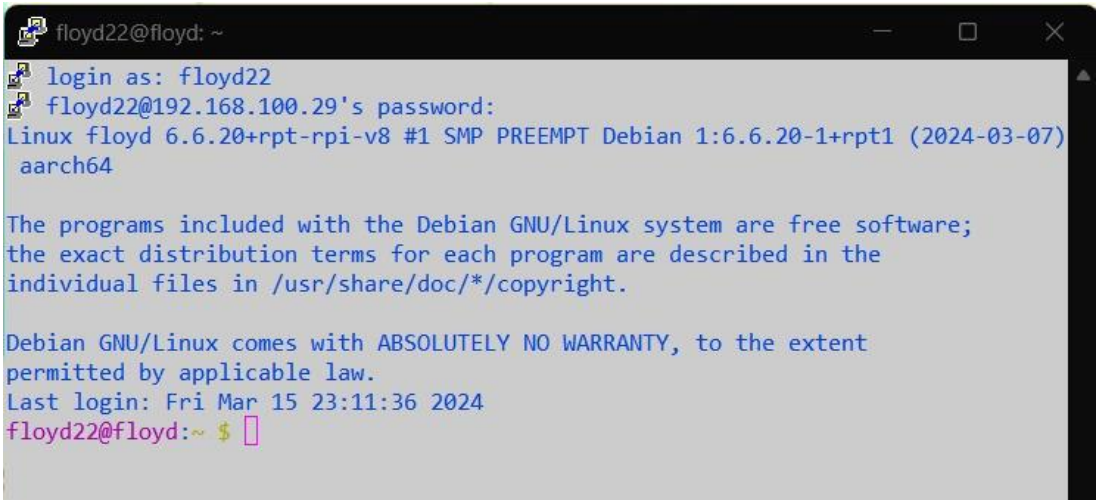


Step 4. Login to puTTy with the credentials we set earlier: use your hostname or your IP Address.



Step 5. Click accept and log in with your credentials.

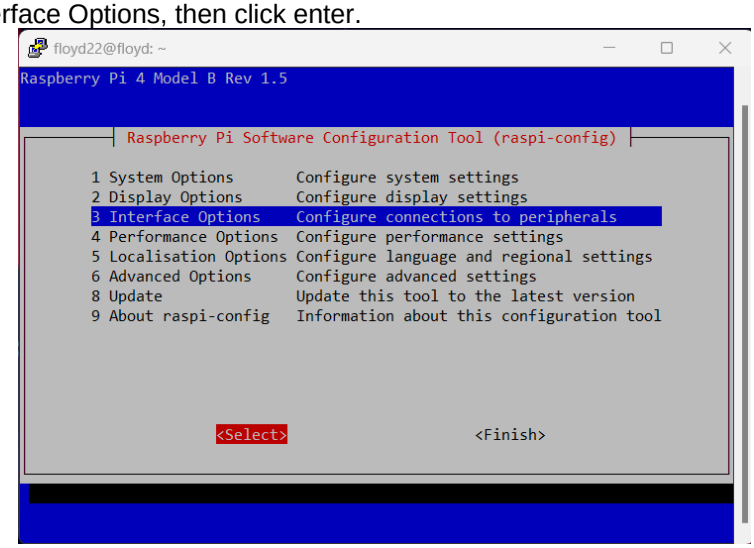




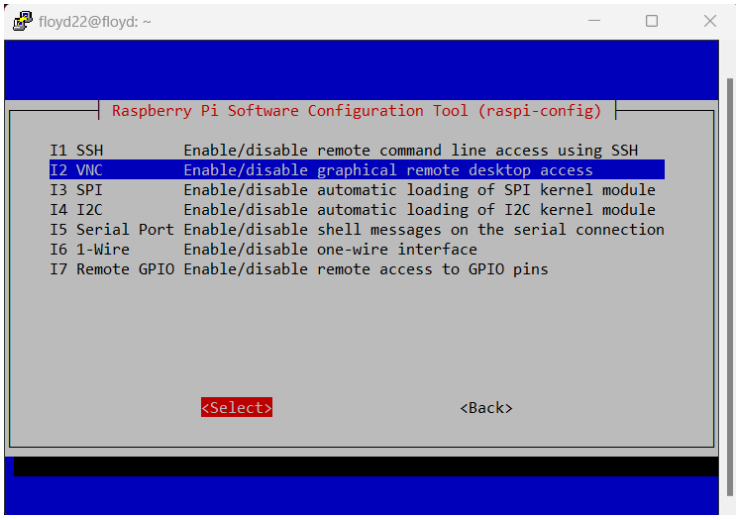
Step 6. Use sudo raspi-config.

Step 7. Enable VNC to access your RPI remotely.

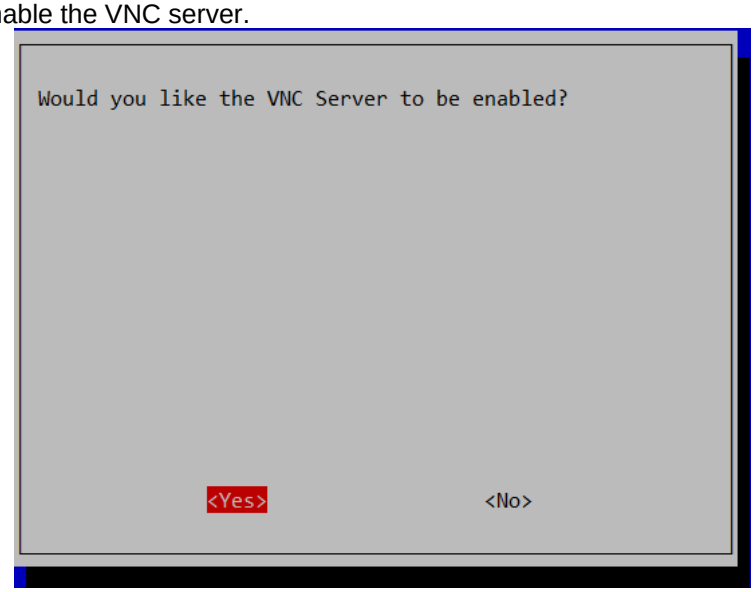
A. Select the Interface Options, then click enter.



B. Select I2 VNC, then click enter.

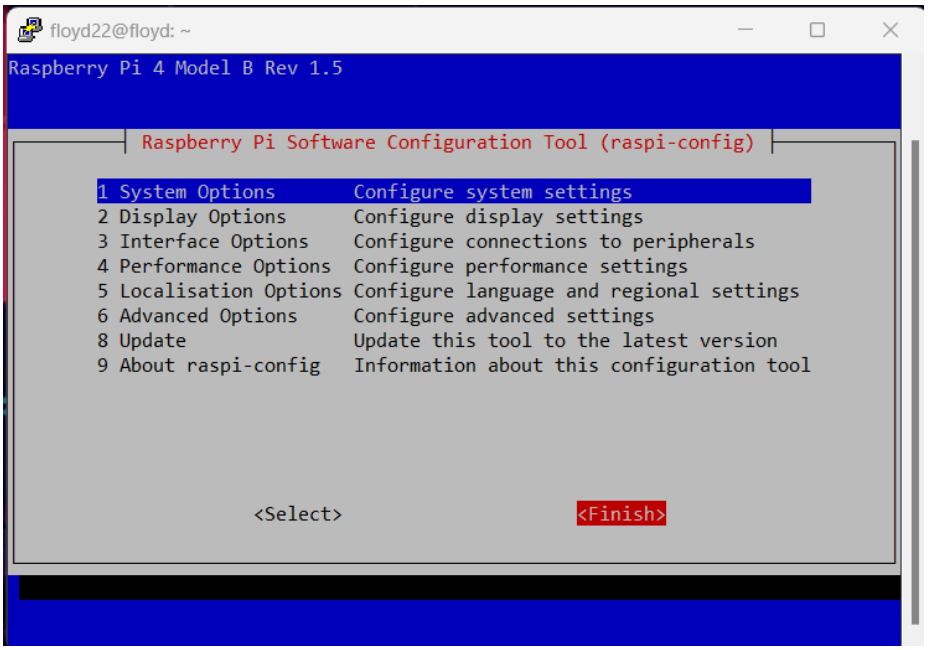


C. Click yes to enable the VNC server.

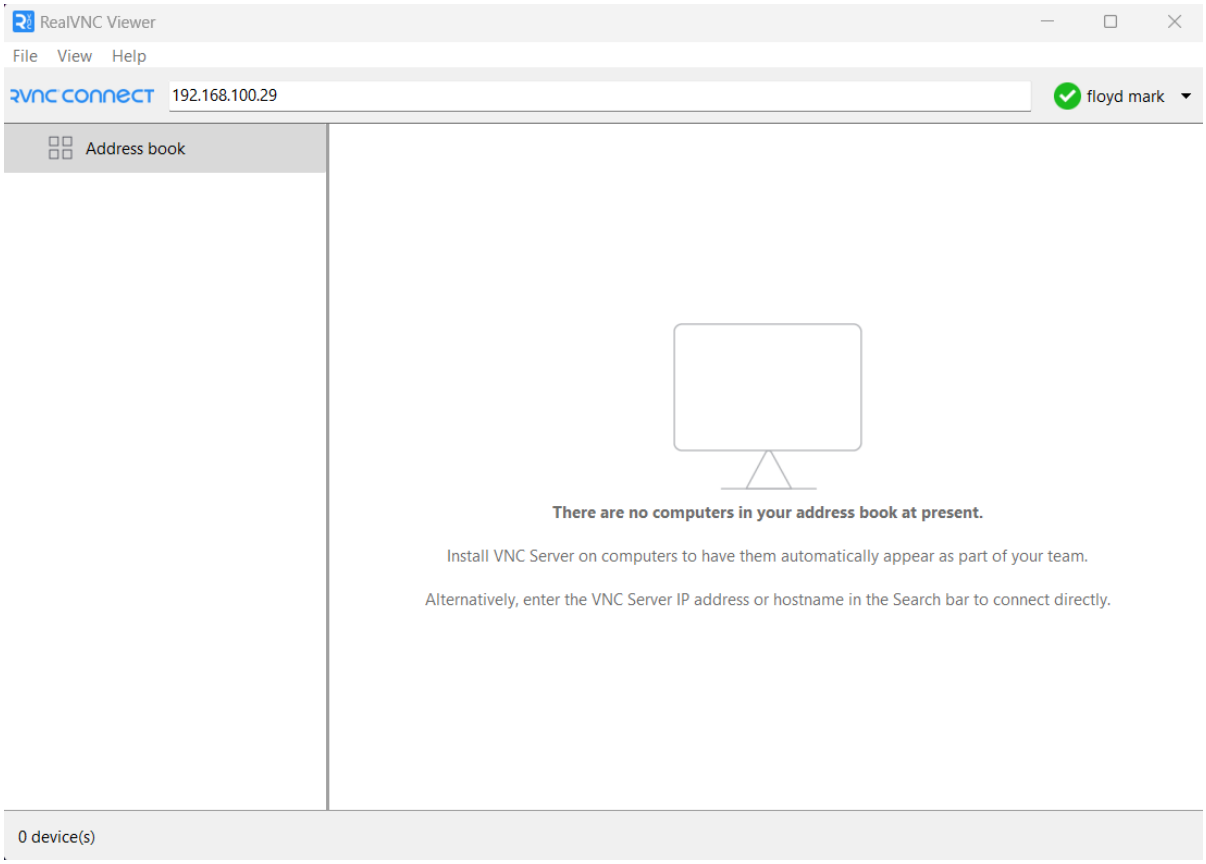


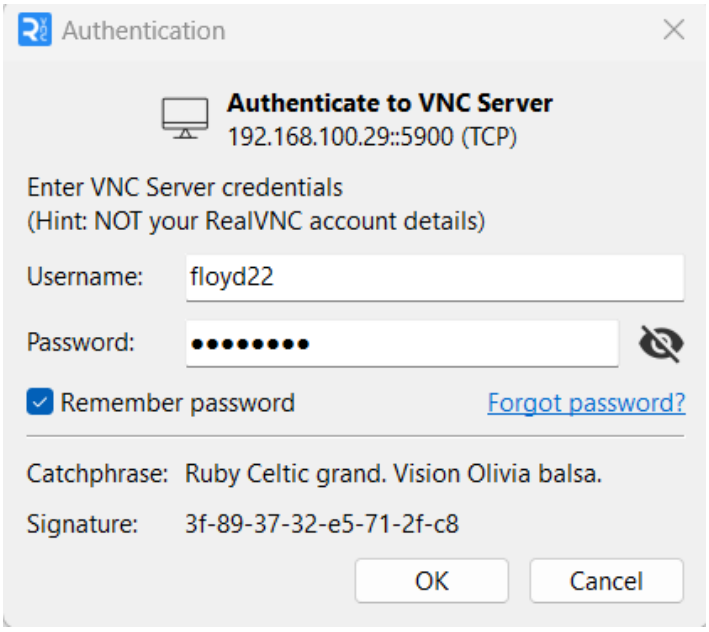
D. Click Ok.

Step 8. Click finish.

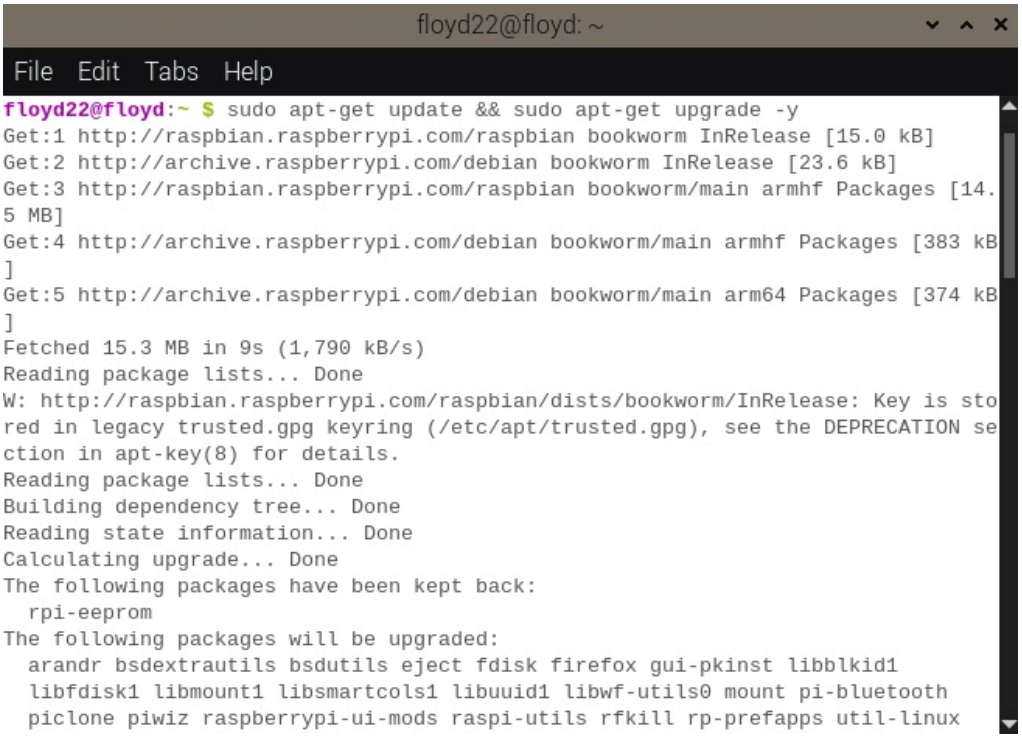


Step 9. Open VNC, type your IP Address or your hostname, then log in with your credentials.

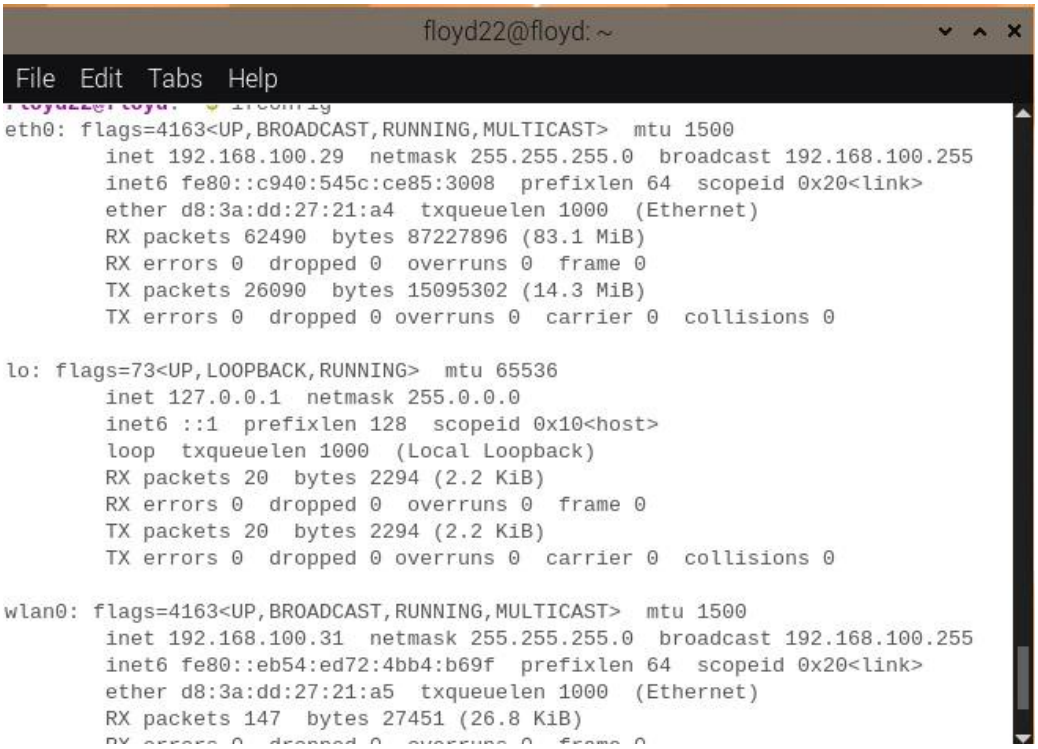




Step 10. Update and upgrade.



Step 11. ifconfig to see your network IP address.



Step 12. Use ssh to connect to your network “ssh floyd22@192.168.100.29”.

```
C:\Users\anche>ssh floyd22@192.168.100.29
The authenticity of host '192.168.100.29 (192.168.100.29)' can't be establis
hed.
ED25519 key fingerprint is SHA256:V3BbBTHiu/Tu6AGrZ4IdmO6RdXVCOB62LNqLdyQwu9
s.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.100.29' (ED25519) to the list of known h
osts.
floyd22@192.168.100.29's password:
Linux floyd 6.6.20+rpt-rpi-v8 #1 SMP PREEMPT Debian 1:6.6.20-1+rpt1 (2024-03
-07) aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Mar 15 23:12:48 2024 from 192.168.100.2
floyd22@floyd:~ $
```

Step 13. Setting up and installing the software for the Ansible internet connection monitor.

- sudo apt-get install -y python3-pip

```
floyd22@floyd: ~
floyd22@floyd:~ $ sudo apt-get install -y python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (23.0.1+dfsg-1+rpt1).
0 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
floyd22@floyd:~ $
```

- pip3 install ansible

```
floyd22@floyd: ~
floyd22@floyd:~ $ pip3 install ansible
Defaulting to user installation because normal site-packages is not writeabl
e
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting ansible
  Downloading https://www.piwheels.org/simple/ansible/ansible-9.4.0-py3-none
-any.whl (46.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.4/46.4 MB 1.2 MB/s eta 0:00:00
Collecting ansible-core~=2.16.5
  Downloading https://www.piwheels.org/simple/ansible-core/ansible_core-2.16
.5-py3-none-any.whl (2.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.3/2.3 MB 1.6 MB/s eta 0:00:00
Requirement already satisfied: jinja2>=3.0.0 in /usr/lib/python3/dist-packag
es (from ansible-core~=2.16.5->ansible) (3.1.2)
Collecting PyYAML>=5.1
  Downloading https://www.piwheels.org/simple/pyyaml/PyYAML-6.0.1-cp311-cp31
1-linux_armv7l.whl (45 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.4/45.4 kB 1.8 MB/s eta 0:00:00
Requirement already satisfied: cryptography in /usr/lib/python3/dist-package
s (from ansible-core~=2.16.5->ansible) (38.0.4)
Collecting packaging
  Downloading https://www.piwheels.org/simple/packaging/packaging-24.0-py3-n
one-any.whl (53 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.5/53.5 kB 2.3 MB/s eta 0:00:00
Collecting resolvelib<1.1.0,>=0.5.3
  Downloading https://www.piwheels.org/simple/resolvelib/resolvelib-1.0.1-py
2.py3-none-any.whl (17 kB)
Installing collected packages: resolvelib, PyYAML, packaging, ansible-core,
ansible
  WARNING: The scripts ansible, ansible-config, ansible-connection, ansible-
console, ansible-doc, ansible-galaxy, ansible-inventory, ansible-playbook, a
nsible-pull and ansible-vault are installed in '/home/floyd22/.local/bin' wh
ich is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
  WARNING: The script ansible-community is installed in '/home/floyd22/.loca
l/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this
warning, use --no-warn-script-location.
Successfully installed PyYAML-6.0.1 ansible-9.4.0 ansible-core-2.16.5 packag
ing-24.0 resolvelib-1.0.1
```

Step 14. Cloning and entering the repository.

- git clone https://github.com/geerlingguy/internet-pi.git

```
floyd22@floyd:~ $ git clone https://github.com/geerlingguy/internet-pi.git
Cloning into 'internet-pi'...
remote: Enumerating objects: 602, done.
remote: Counting objects: 100% (262/262), done.
remote: Compressing objects: 100% (123/123), done.
remote: Total 602 (delta 168), reused 152 (delta 139), pack-reused 340
Receiving objects: 100% (602/602), 305.94 KiB | 2.53 MiB/s, done.
Resolving deltas: 100% (363/363), done.
```

- cd internet-pi

```
floyd22@floyd:~ $ ls
Bookshelf  Documents  internet-pi  Pictures  Templates
Desktop    Downloads  Music        Public    Videos
floyd22@floyd:~ $ cd internet-pi
floyd22@floyd:~/internet-pi $ ls
ansible.cfg            files                LICENSE    requirements.yml
example.config.yml     images               main.yml   tasks
example.inventory.ini  internet-monitoring  README.md  templates
floyd22@floyd:~/internet-pi $
```

Step 15. Install the requirements. Ansible-galaxy collection install -r requirements.yml (if you see ansible-galaxy: command not found, restart your SSH session or reboot the Pi and try again) [1]

```
floyd22@floyd:~/internet-pi $ ansible-galaxy collection install -r requireme
nts.yml
Starting galaxy collection install process
Nothing to do. All requested collections are already installed. If you want
to reinstall them, consider using `--force`.
floyd22@floyd:~/internet-pi $
```

Step 16. Copy example.inventory.ini to inventory.ini and copy example.config.yml to config.yml (configure the inventory.ini and change the IP to your RPI's IP).

```
floyd22@floyd:~/internet-pi $ cp example.inventory.ini inventory.ini
floyd22@floyd:~/internet-pi $ cp example.config.yml config.yml
floyd22@floyd:~/internet-pi $ nano inventory.ini
```

Step 17. After all configuration, run the playbook to gather all succeeding needed.

```
floyd22@floyd: ~/internet-pi
 GNU nano 7.2                    inventory.ini *
[internet_pi]
192.168.100.29 ansible_user=floyd22

# Comment out the previous line and uncomment this to run inside Raspberry >
# 127.0.0.1 ansible_connection=local ansible_user=pi
```

- ansible-playbook main.yml

```
floyd22@floyd:~/internet-pi $ ansible-playbook main.yml

PLAY [Configure Internet Pi.] ***************************************************
****

TASK [Gathering Facts] **********************************************************
****
The authenticity of host '192.168.100.29 (192.168.100.29)' can't be establis
hed.
ED25519 key fingerprint is SHA256:V3BbBTHiu/Tu6AGrZ4IdmO6RdXVCOB62LNqLdyQwu9
s.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
fatal: [192.168.100.29]: UNREACHABLE! => {"changed": false, "msg": "Failed t
o connect to the host via ssh: Warning: Permanently added '192.168.100.29' (
ED25519) to the list of known hosts.\r\nfloyd22@192.168.100.29: Permission d
enied (publickey,password).", "unreachable": true}

PLAY RECAP **********************************************************************
****
192.168.100.29             : ok=0    changed=0    unreachable=1    failed=0
   skipped=0    rescued=0    ignored=0

floyd22@floyd:~/internet-pi $
```

```
floyd22@floyd:~/internet-pi $ sudo nano /etc/ssh/sshd_config
floyd22@floyd:~/internet-pi $
```

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication yes
ChallengeResponseAuthentication no
PermitEmptyPasswords yes
```

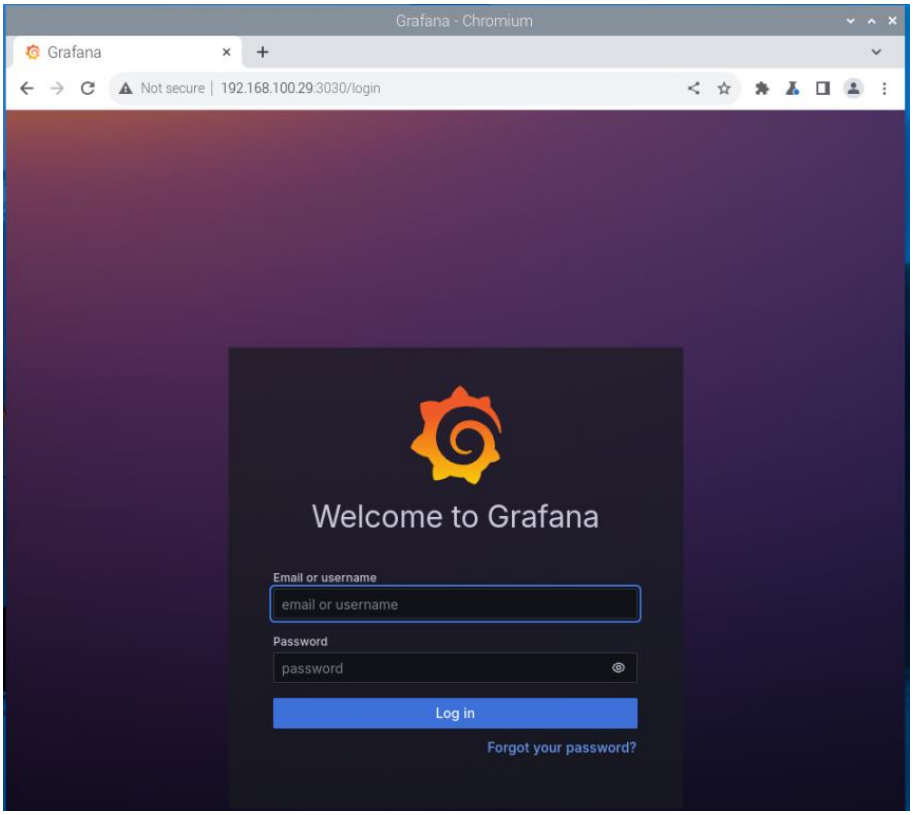
```
floyd22@floyd:~ $ sudo passwd -d floyd22
passwd: password changed.
floyd22@floyd:~/internet-pi $ sudo systemctl restart sshd
floyd22@floyd:~/internet-pi $ sudo reboot
```

```
floyd22@floyd:~/internet-pi $ ansible-playbook main.yml

PLAY [Configure Internet Pi.] ***************************************************
****

TASK [Gathering Facts] **********************************************************
****
ok: [192.168.100.29]

TASK [Load configuration (with defaults from example file).] ********************
****
ok: [192.168.100.29] => (item=example.config.yml)
ok: [192.168.100.29] => (item=config.yml)

TASK [Ensure apt cache is up to date.] ******************************************
****
ok: [192.168.100.29]

TASK [Ensure pacman cache is up to date] ****************************************
****
skipping: [192.168.100.29]

TASK [Check if Docker is already present.] **************************************
****
```

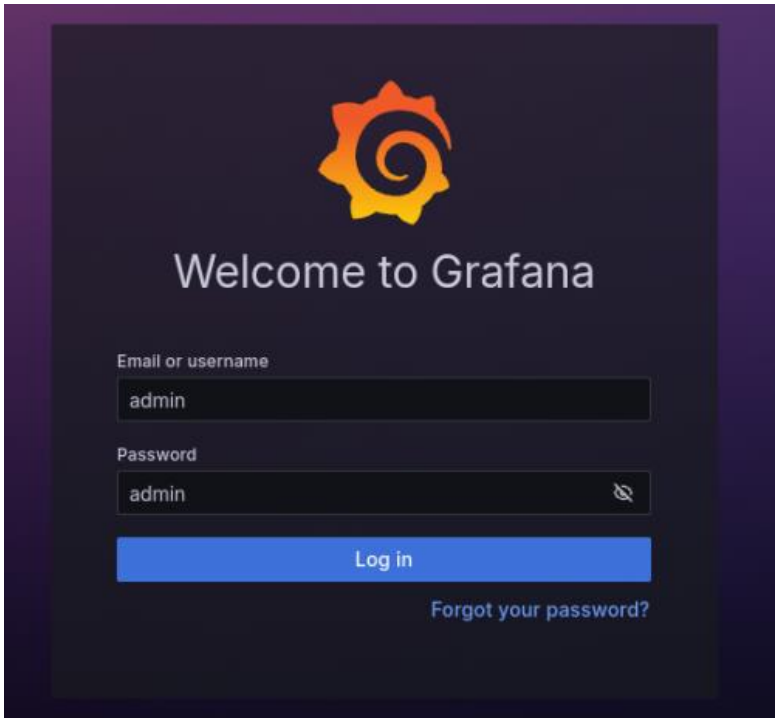
```
floyd22@floyd:~/internet-pi $ sudo passwd floyd22
New password:
Retype new password:
passwd: password updated successfully
floyd22@floyd:~/internet-pi $
```

Step 18. Run the IP address to the browser with port 3030.

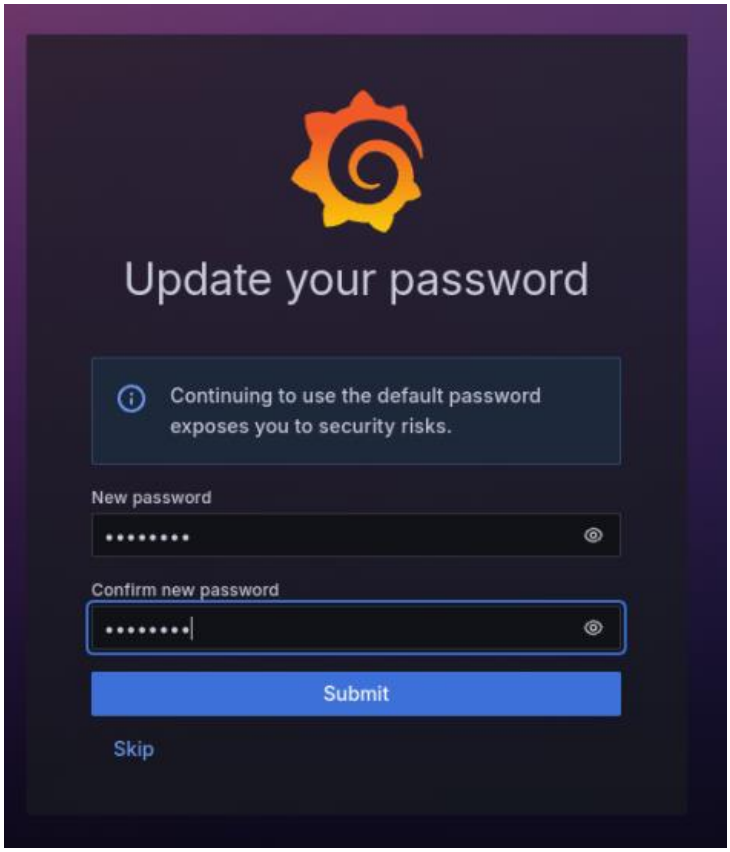
- http://192.168.100.29:3030



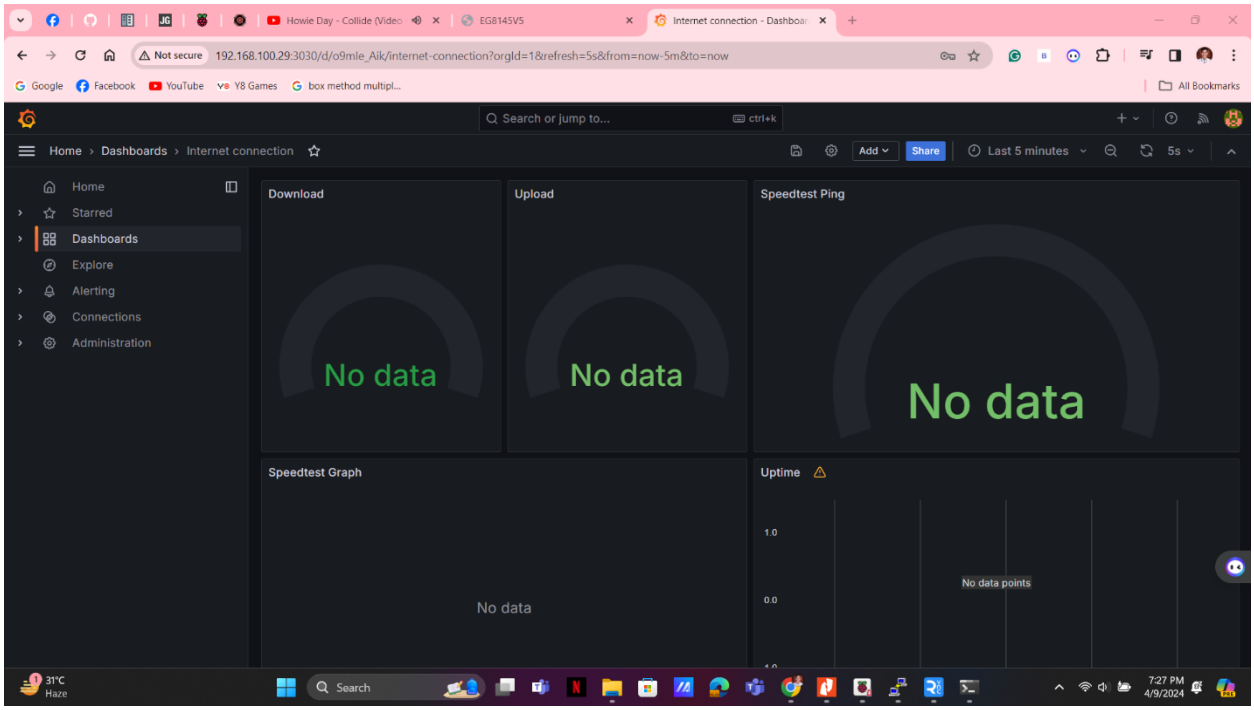
Step 19. Log in using the 'admin' username and 'admin' password.

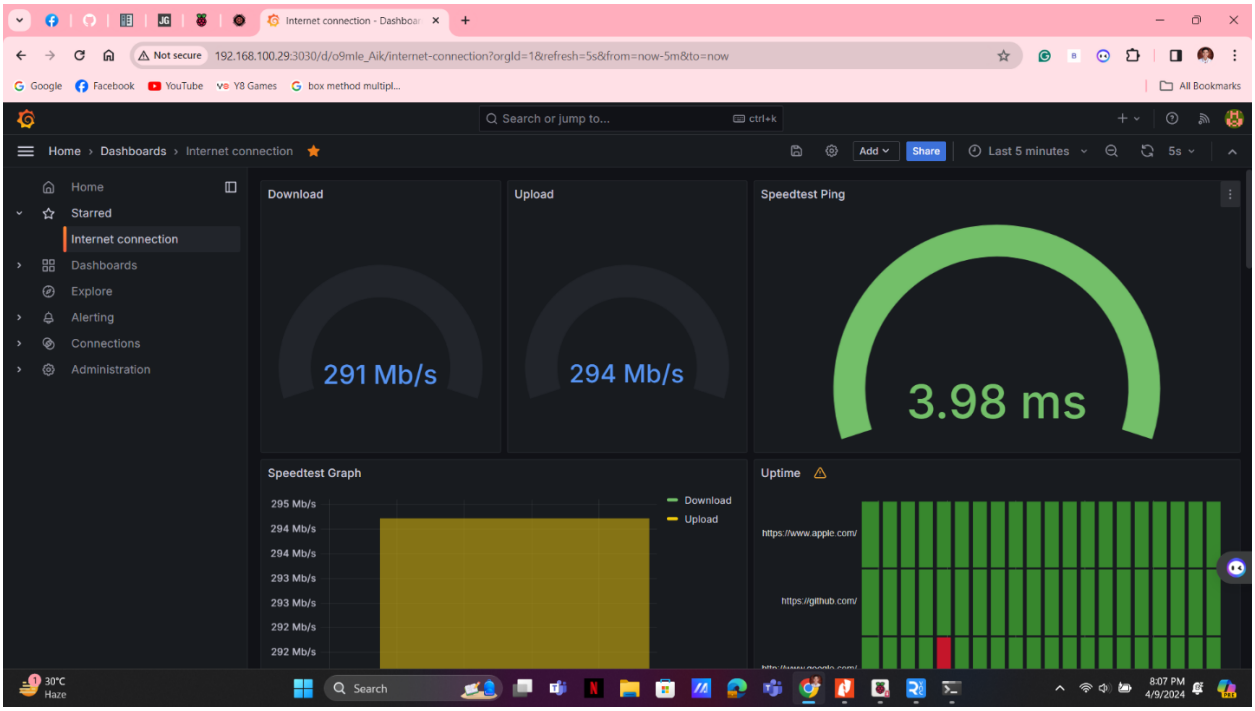


Step 20. Update your password.



Step 21. Go to the dashboard and search for an internet connection, and that's it.





SUMMARY / CONCLUSION

This module provided us with a thorough understanding of internet connection monitoring using Raspberry Pi. We obtained a comprehensive understanding of the subject by studying the importance of internet connection monitoring and dependable connectivity across diverse apps. We discovered the possibilities for using Raspberry Pi in network monitoring initiatives after thoroughly investigating its hardware components and capabilities as a versatile single-board computer. The step-by-step instructions for configuring Raspberry Pi, installing monitoring software, and scripting internet connection checks enabled us to quickly set up our monitoring nodes. Additionally, the emphasis on data logging, visualization, alerts, and troubleshooting guarantees that we have the necessary tools to properly manage and maintain robust internet connection monitoring systems.

Furthermore, this module has addressed not only technical factors, but also the significance of security considerations in internet connection monitoring. We became aware of the importance of maintaining data integrity and system resilience after reviewing security best practices and putting steps in place to protect the monitoring system and its data. Moreover, the optional study of integrating internet connection monitoring with home automation systems through Raspberry Pi provides significant possibilities for practical applications and automated responses based on connectivity status. Through engaging hands-on exercises and project work, we were able to solidify our understanding and apply our knowledge in real-world scenarios, fostering a greater appreciation for the complexities and nuances of internet connection monitoring in today's interconnected world.

REFERENCES

Geerlingguy, J. (2021). *GitHub - geerlingguy/internet-pi: Raspberry Pi config for all things Internet.* GitHub. Retrieved from https://github.com/geerlingguy/internet-pi.

For your additional References that has been used please use the citation tools on the website https://www.scribbr.com/ using APA 7th Edition English, and paste it below.