

Documentation

Installing Ubuntu on your Machine

First, we are going to want to download and install a boot-able disk image of Ubuntu server edition onto a USB flash drive. If you're on Windows I'd recommend you use RUFUS and if you're running on a Linux machine the Popsicle USB Flasher is my go-to tool.

Download Ubuntu Server Edition

Once the disk image has been flashed onto the USB you are going to want to boot into that USB. Depending on your hardware you are going to need to hit a key to get into your boot when the manufacturer splash screen has displayed.

Once booted in you're going to want to run through the Ubuntu server setup. This is fairly straight forward.

1. Select your Language.
2. Select '**Continue without updating**', we will update the system later.
3. Select your keyboard layout, if using 'English (US)' click 'Done'.
4. Select your network adapter. Ensure you have a valid connection and IP address. Since this is going to be used as a server I'd recommend using a direct Ethernet connection. This will make this way easier.
5. Leave the proxy address blank.
6. Leave Ubuntu mirror at default.
7. Select the disk you'd like to use. For this demonstration we will be using the entire disk and deselecting the LVM group option. You will be able to see a summary of the file system changes.
8. Set up your name, server name, username, and password.
9. Select 'Install OpenSSH server'.
10. Skip Snap installs for now.

From here you're going to want to remove the USB drive and restart your Ubuntu server. After the first boot you're going to want to login with the user and password you created when setting up the machine. Once you're in run the commands to update your system.

When booted into server run apt update commands

```
sudo apt update && sudo apt upgrade
```

Setting up Ubuntu for SSH

Once your system is updated we're going to need to get the local IP address to SSH into the system. You could do all this on the server machine, but using SSH will make it way easier to copy commands and easily move media files to your new server. First to find your IP run the following command.

```
ip a
```

Your IP address will be after 'inet' and will likely be the first address that doesn't end in a 0 or 1. For my machine the local IP address is 192.168.0.60, so that's what I'll be using for the rest of this article.

Once you have this done you're going to want to make sure you have OpenSSH installed on your main computer. You can run "ssh" in the terminal to check. As a reminder this will only work on your local network. In order to connect to a machine outside of your home network you will want to set up port forwarding and add additional security to your systems.

Once you have OpenSSH installed you can SSH into your new server.

```
ssh 192.168.0.60
```

It will ask you for your password and once you are in it will look like a normal terminal instance, but it will be for the server as it displays as username@servername. Once you are in we can setup plex.

Laptop Tip: Disable Suspend on Lid Close.

To do this we will need to edit the Systemd login config file.

```
sudo nano /etc/systemd/logind.conf
```

When editing this you should only need to uncomment and change the first HandleSuspendKey option, but depending on your hardware you can edit additional settings. "Uncomment" the following keys and change their values to ignore:

```
HandleSuspendKey=ignore
```

```
HandleLidSwitch=ignore
```

```
HandleLidSwitchDocked=ignore  
sudo systemctl restart systemd-logind
```

Problems and Troubleshooting

1. SSH Connection Issue

```
PS C:\Users\Burak> ssh erdilli@192.168.1.190
erdilli@192.168.1.190's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-51-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sat Dec 21 07:18:25 PM UTC 2024

System load:  1.55               Processes:           191
Usage of /:   57.7% of 97.87GB   Users logged in:    0
Memory usage: 5%                IPv4 address for enp4s0: 192.168.1.190
Swap usage:   0%                IPv4 address for enp4s0: 192.168.1.102
Temperature:  49.0 C

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

9 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

3 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

Last login: Sat Dec 21 12:41:03 2024 from 192.168.1.103
erdilli@lenovo:~$
```

- **Problem:** During the first SSH connection attempt, outdated syntax in the connection string caused confusion. After being prompted for a password, an error log appeared.
- **Error Log:**
 - ssh: Could not resolve hostname user@192.168.X.X: Name or service not known
- **Cause:** Incorrect syntax in the SSH command, specifically the hostname was misformatted.
- **Solution:** Used the correct syntax:

```
ssh user@192.168.X.X
```

Double-checked hostname, username, and IP address.

2. SSH Authentication Failure

```

erdilli@lenovo:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp4s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether f0:de:f1:7a:7e:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.190/24 brd 192.168.1.255 scope global enp4s0
        valid_lft forever preferred_lft forever
    inet 192.168.1.102/24 metric 100 brd 192.168.1.255 scope global secondary dynamic enp4s0
        valid_lft 3128sec preferred_lft 3128sec
    inet6 fe80::f2de:f1ff:fe7a:7e89/64 scope link
        valid_lft forever preferred_lft forever
3: wlp3s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 74:e5:0b:1d:c6:8c brd ff:ff:ff:ff:ff:ff
4: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:f5:14:90:cd brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever

```

- **Problem:** Despite using the correct SSH syntax, authentication repeatedly failed even with the right password.
- **Error Log:**

```
authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
rhost=192.168.1.181 user=erdilli Failed password for erdilli
from 192.168.1.181 port 55846 ssh2
```
- **Cause:** SSH keys or password authentication settings were misconfigured on the server.
- **Solution:**
 - Ensured password authentication was enabled in the SSH configuration file (`/etc/ssh/sshd_config`):
 - `PasswordAuthentication yes`
 - Restarted the SSH service:
 - `sudo systemctl restart sshd`
 - Verified user permissions and reset the password if necessary.

3. Docker Container Restart Loop

- **Problem:** One of the Docker containers was stuck in a restart loop.
- **Cause:** The container lacked proper volume mounts, and a misconfigured `docker-compose.yml` file caused persistent failures.
- **Solution:** Corrected volume mounts in the `docker-compose.yml` :

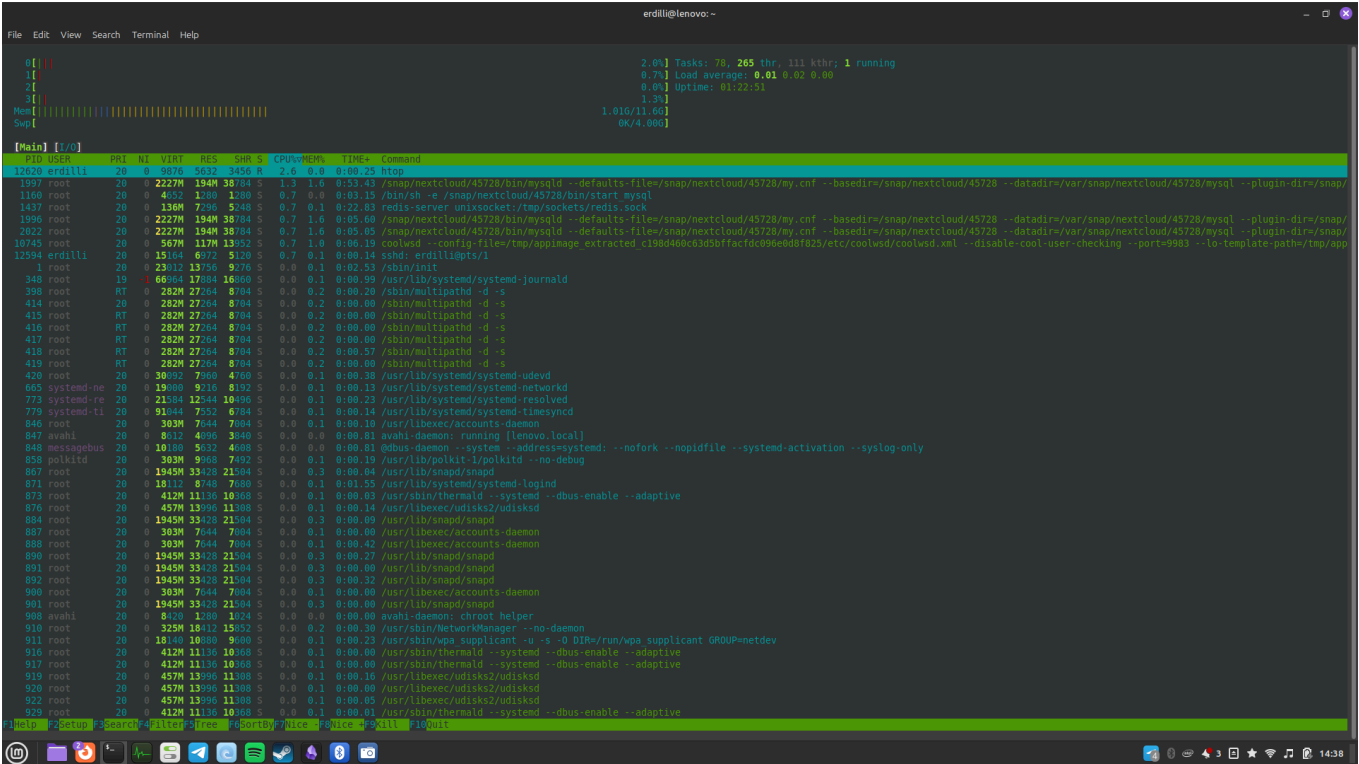
```
volumes: - ./data:/app/data
```

Restarted the Docker container:

- `docker-compose down && docker-compose up -d`



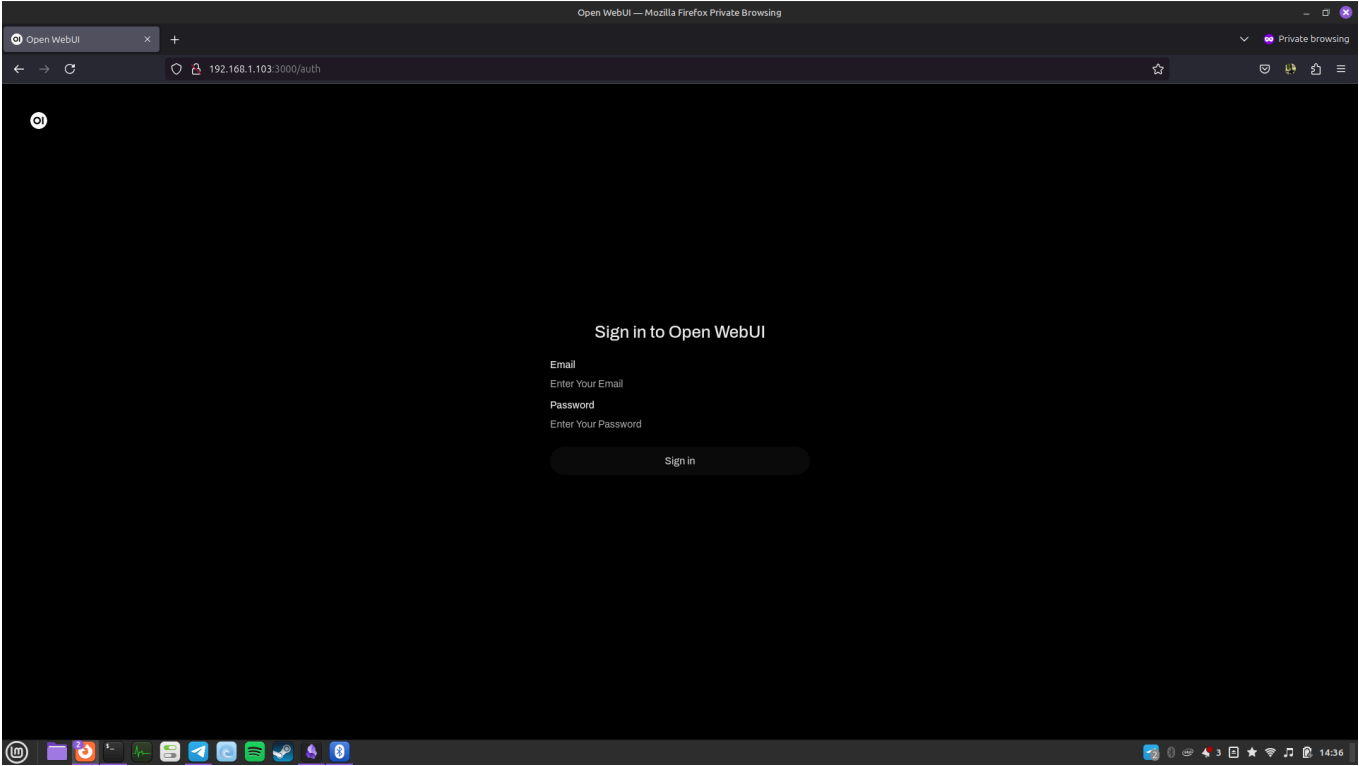
5. Resource Constraints on Hardware



- **Problem:** Limited RAM and CPU resources on the Lenovo and Samsung hardware led to performance issues, especially when running multiple services.
- **Cause:** Overcommitment of resources by Docker containers and unnecessary background processes.
- **Solution:** Optimized Docker container resource allocation:
- `deploy: resources: limits: memory: 512M cpus: '0.5'`

Disabled unnecessary services and optimized container images.

4. LLM Web UI Access Issue



- **Problem:** The lightweight LLMs failed to start the web UI on the designated port.
- **Cause:** Port conflicts with other services running on the same hardware.
- **Solution:** Changed the default port for the LLM web UI:

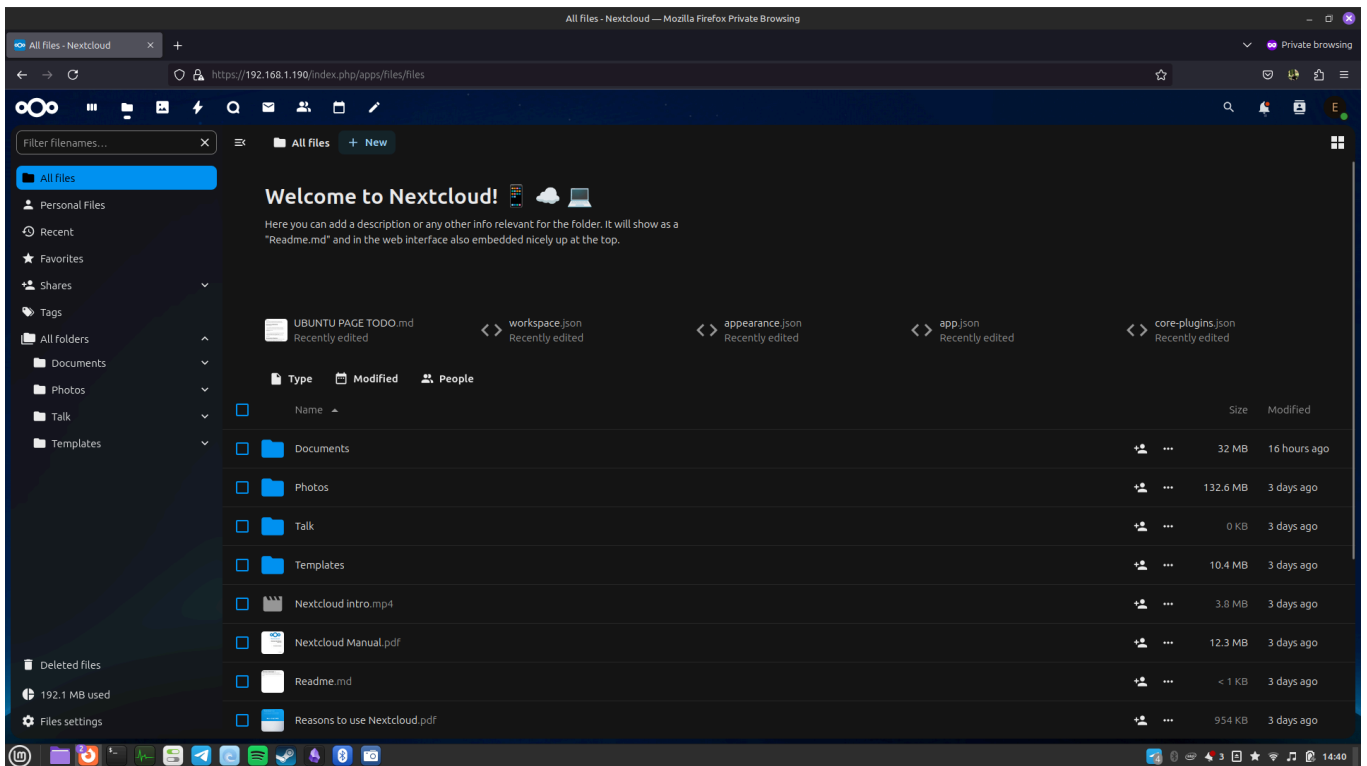
```
--port 8081
```

Ensured the port was open in the firewall:

- ```
ufw allow 8081
```

---

## 5. Nextcloud Performance Issues



- **Problem:** Nextcloud was slow and occasionally unresponsive.
- **Cause:** Insufficient memory allocation and high disk I/O.
- **Solution:**
  - Enabled caching via Redis.
  - Adjusted PHP memory limits in `php.ini`:
- `memory_limit = 512M`
- Scheduled periodic database optimization tasks.

## 6. Nextcloud Installation Issue

- **Problem:** Upon checking system logs using:
- `sudo tail -f /var/log/syslog`

It indicated that Nextcloud was not fully installed, and only limited functionality was available.

- **Cause:** Incomplete or failed Nextcloud installation/configuration.
- **Solution:**
  - Re-ran the Nextcloud setup:
- `sudo nextcloud.occ maintenance:install`
- Verified database and storage configurations.
- Ensured required dependencies were installed:
- `sudo apt install php-gd php-json php-curl php-mbstring php-intl php-imagick php-xml php-zip`
- Restarted Apache/Nginx:
- - `sudo systemctl restart apache2`



## 7. Passwordless Boot Configuration

- **Problem:** The server required a password on reboot, causing delays in automated processes.
- **Cause:** Encrypted or user-password-protected boot process.
- **Solution:**
  - Edited the `/etc/sudoers` file to allow passwordless operations for reboot:
  - `user ALL=(ALL) NOPASSWD: /sbin/reboot`
  - Disabled password prompt on startup via `systemctl` :
    - `sudo systemctl enable autologin@<your_username>`
    - Ensured `autologin` was properly configured in `/etc/systemd/system/getty@tty1.service.d/autologin.conf`.

## 8. Preventing Sleep and Lid Actions

- **Problem:** The server would sleep or suspend when the lid was closed or after a period of inactivity.
- **Cause:** Default power management settings.
- **Solution:**
  - Edited `/etc/systemd/logind.conf` and added the following lines:
  - `HandleSuspendKey=ignore` `HandleLidSwitch=ignore`  
`HandleLidSwitchDocked=ignore`
  - Restarted the `systemd-logind` service:
  - `sudo systemctl restart systemd-logind`
  - Verified that the changes were applied:

```
systemctl status systemd-logind
```

## 9. Setting a Static IP Address

DHCP Address Reservation

This page displays the static IP address assigned by the DHCP Server and allows you to adjust these configurations by clicking the corresponding fields.

|                          |                   |               |               |         |                      |
|--------------------------|-------------------|---------------|---------------|---------|----------------------|
| <input type="checkbox"/> | MAC Address       | IP Address    | Group         | Status  | Edit                 |
| <input type="checkbox"/> | 9C:9D:7E:82:D8:3A | 192.168.1.100 | Default Group | Enabled | <a href="#">Edit</a> |
| <input type="checkbox"/> | 68:E4:3B:30:8B:0C | 192.168.1.110 | Default Group | Enabled | <a href="#">Edit</a> |

Add New

Enable Selected

Disable Selected

Delete Selected

Refresh

### Problem:



The server's IP address changes dynamically after every reboot or network reset, causing disruptions when accessing it remotely via SSH or services like Nextcloud.

## Solution:

To set a static IP address on a Linux server using Netplan:

### 1. Edit Netplan Configuration:

Open the Netplan configuration file:

- `sudo nano /etc/netplan/00-installer-config.yaml`

#### • Configure Static IP:

Replace or adjust the existing configuration to something like this:

- `network: version: 2 ethernet: enp4s0: dhcp4: no addresses: - 192.168.1.100/24 # Static IP Address gateway4: 192.168.1.1 # Default Gateway nameservers: addresses: - 8.8.8.8 - 8.8.4.4`
  - Replace `enp4s0` with your network interface name ( `ifconfig` or `ip a` will show it).
  - Update `192.168.1.100` to your desired static IP.
  - Adjust `gateway4` to match your router's IP.

#### • Fix Permissions Warning:

Netplan expects strict permissions on the YAML file. To fix the error:

- `sudo chmod 600 /etc/netplan/00-installer-config.yaml`
- **Apply Changes:**
- `sudo netplan apply`
- **Verify Configuration:**

### 1. `ip a`

Ensure your interface now has the static IP address.

---

## 10. Wake-on-LAN (WoL) Configuration

### Problem:

Some devices have Wake-on-LAN enabled ( `enp4s0` and `wlp3s0` ), while others (USB devices) are disabled. The server might not wake up reliably.

### Understanding the Output:

- Enabled on `enp4s0` : The Ethernet port supports Wake-on-LAN.
- Enabled on `wlp3s0` : The Wi-Fi port supports Wake-on-LAN (less reliable).
- Disabled on USB devices: USB wake signals are ignored.

## Can the Server Be Woken?

Yes, via the Ethernet interface `enp4s0` . Wake-on-LAN over Ethernet is generally the most reliable method.

## Solution: Enable Wake-on-LAN Permanently

### 1. Install ethtool (if not already installed):

- `sudo apt install ethtool`
- **Enable Wake-on-LAN:**
- `sudo ethtool -s enp4s0 wol g`
- **Make it Persistent Across Reboots:**  
Create or edit a systemd service:

```
sudo nano /etc/systemd/system/wol@.service
```

Add the following content:

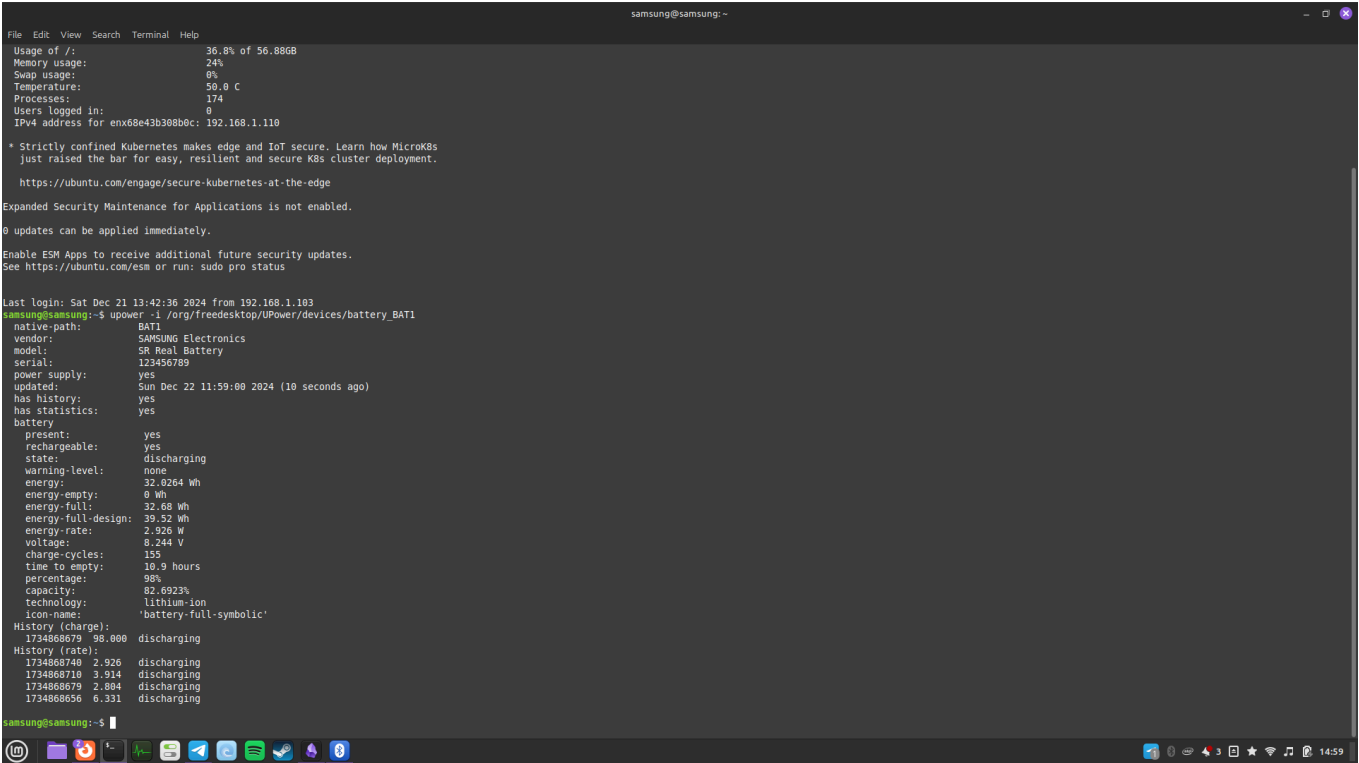
- `[Unit] Description=Wake-on-LAN for %i [Service] Type=oneshot ExecStart=/sbin/ethtool -s %i wol g [Install] WantedBy=default.target`
- **Enable the Service:**
- `sudo systemctl enable wol@enp4s0.service sudo systemctl start wol@enp4s0.service`
- **Verify WoL Status:**

1. `sudo ethtool enp4s0 | grep Wake-on`

Ensure it shows `Wake-on: g` .

---

## \*\*11. Charging State Monitoring Of the Device #1



The current adapter for the single USB Type-C 3.0 port on Device #1 cannot provide enough power, and it also limits the gigabit potential down to 100 Mbps. To resolve this issue, a new adapter will be added.

### New Adapter Specs:

- **Model:** Apera NR43 5-in-1 HUB
- **Port Types:** Thunderbolt 3, USB Type-C, Lightning, USB 3.0, Card Reader
- **Features:** Fast charging, aluminum casing
- **Compatibility:** Mac Type-C

The new adapter will address both the power supply and data transfer speed limitations.

# Documentation for Running Python Scripts Remotely via SSH

## Prerequisites

- **SSH Access:** You should have access to the remote machine via SSH.
- **Python 3:** Ensure that Python 3 is installed on the remote machine. The machine should have access to `python3` and `pip3` (Python package manager).
- **Virtual Environment Setup:** You will be working within a virtual environment to manage dependencies cleanly.

# Steps to Set Up and Run the Python Scripts

## 1. Access the Remote Machine via SSH

Open a terminal and connect to the remote server using SSH:

```
ssh erdilli@lenovo
```

Replace `erdilli` and `lenovo` with your username and the hostname or IP address of the remote machine.

## 2. Navigate to the Project Directory

Once you're logged in, navigate to the directory where your Python scripts are located:

```
cd ~/projects/trendyol
```

## 3. Create a Virtual Environment

Create a virtual environment to isolate your dependencies:

```
python3 -m venv myenv
```

## 4. Activate the Virtual Environment

Activate the virtual environment to ensure the Python packages are installed within it:

```
source myenv/bin/activate
```

## 5. Install Required Packages

Inside the virtual environment, install the required packages ( `pandas` and `openpyxl` ):

```
pip install pandas openpyxl
```

## 6. Run the Python Scripts

You can now run the Python scripts and capture the output to log files.

For example, to run `kampanya.py` :

```
python3 /home/erdilli/projects/trendyol/kampanya.py >
/home/erdilli/projects/trendyol/kampanya_output.log 2>&1
```

This command will:

- Run the script `kampanya.py` .
- Redirect both the standard output ( `stdout` ) and standard error ( `stderr` ) to the `kampanya_output.log` file.

## 7. Check the Log for Errors

If the script fails due to missing dependencies or errors, you can check the generated log file:

```
cat /home/erdilli/projects/trendyol/kampanya_output.log
```

Common errors might include missing Python packages. In such cases, install the missing packages using `pip` .

## 8. Install Missing Dependencies

If you encounter errors related to missing dependencies like `openpyxl` , install them:

```
pip install openpyxl
```

## 9. Re-run the Script After Installing Dependencies

After installing any missing dependencies, re-run the script:

```
python3 /home/erdilli/projects/trendyol/kampanya.py > /home/erdilli/projects/trendyol/kampanya_output.log 2>&1
```

## 10. Deactivate the Virtual Environment (Optional)

Once you've completed running your scripts, deactivate the virtual environment:

```
deactivate
```

---

## Troubleshooting

**Issue:** `ModuleNotFoundError: No module named 'pandas'`

- **Solution:** Install pandas inside your virtual environment using:
- `pip install pandas`

## **Issue:** `ImportError: Missing optional dependency 'openpyxl'`

- **Solution:** Install `openpyxl` using:
- `pip install openpyxl`

## **Issue:** `pip3` Not Found

- **Solution:** If `pip3` is not available, you may need to install it using:
  - `sudo apt install python3-pip`
- 

# **Documentation for the Python Script: Filtering and Processing Products for Campaigns**

## **Purpose:**

This script is designed to filter and process product data based on certain conditions from Excel files and create a filtered campaign-ready dataset. The script performs the following steps:

1. Loads data from multiple Excel files (favorites report and product details).
2. Filters products based on stock availability, sales prices, and view counts.
3. Adjusts product pricing and saves the filtered data to a new Excel file.

## **Requirements:**

- Python 3.x
- Required Libraries:
  - `pandas`
  - `os`
  - `logging`
  - `openpyxl`

## **Inputs:**

- `favori-görüntüleme-raporu.xlsx` : An Excel file containing the favorite viewing report.
- `indirilen_dosya_kampanya/*.xlsx` : Excel files in the "indirilen\_dosya\_kampanya" folder containing product details.

## Outputs:

- Filtered products stored in `kampanya/filtered_products.xlsx`.

## Logging:

- Logs are written to both the terminal (console) and a log file ( `kampanya_script_output.log` ).
  - Logs include information about each major operation (file loading, filtering, saving) and any errors encountered.
- 

## Script Overview:

### 1. Logging Setup:

- **Purpose:** Tracks script progress and errors.
- **Configuration:**
  - Logs are written to both a file ( `kampanya_script_output.log` ) and the terminal.
  - Each log message includes the timestamp, log level (INFO, ERROR), and message.
  - Log levels:
    - `INFO` : For normal operations (file loading, successful filtering, etc.).
    - `ERROR` : For any failures (file not found, filtering issues).

### 2. Directories Setup:

- The script automatically determines the current directory ( `current_dir` ) where it is run.
- Defines input and output directories:
  - `SALLES_DIR` : The directory where product files are located ( `indirilen_dosya_kampanya` ).
  - `OUTPUT` : The directory where the filtered products will be saved ( `kampanya` ).

### 3. Loading Files:

- **Favorite Viewing Report** ( `favori-görüntüleme-raporu.xlsx` ):



- The script scans the current directory for the file and loads it into a `pandas DataFrame ( df_favorite )`.
- If the file is not found, a `FileNotFoundError` is raised, and the error is logged.
- **Product Data ( \*.xlsx in SALLES\_DIR ):**
  - The script scans the `SALLES_DIR` for suitable product files and loads the first matching file into a `DataFrame ( df_products )`.
  - If no product file is found, a `FileNotFoundError` is raised, and the error is logged.

#### 4. Data Filtering:

- **Stock and Price Filtering:**
  - Products are filtered to include only those with stock greater than 0 and a sales price greater than 0.
  - Log message indicates how many products remain after this step.
- **Price Difference Filtering:**
  - Products are filtered to ensure the price difference between "Mevcut Satış Fiyatı" (current sale price) and "Maksimum Girebileceğin Fiyat" (maximum price a product can be sold for) is within a predefined threshold (`MAX_PRICE_DIFFERENCE`).
  - Log message indicates how many products remain after this step.
- **View Count Filtering:**
  - Products are filtered to only include those with "Toplam Görüntülenme Sayısı" (total views) less than the threshold (`VIEWED`), based on corresponding data in the favorite report.
  - Log message indicates how many products remain after this step.

#### 5. Price Adjustment:

- **Filling Campaign Price:**
  - The "Kampanyalı Satış Fiyatı" (campaign sale price) column is filled with the values from "Maksimum Girebileceğin Fiyat" (maximum price).
  - This is the price that will be used for the campaign.

#### 6. Saving Results:

- The filtered `DataFrame ( df_products )` is saved to an Excel file (`filtered_products.xlsx`) in the `kampanya` directory.
- If the output directory does not exist, it is created.

- A log message indicates the path where the filtered products were saved.

## 7. Output:

- The filtered products are printed to the console as a preview, showing the first few rows of the DataFrame.
- 

## Detailed Functionality:

### Logging:

- Logging is initialized at the beginning of the script to capture all operations and errors.
- Each key step in the script (loading files, filtering data, saving results) is logged.
- Error messages are captured and logged when an issue occurs, such as missing files or invalid data.

### Directory Management:

- The script ensures that both input and output directories are correctly set up. If the output directory does not exist, it is created.
- The script dynamically determines the current directory using `os.path.dirname(os.path.abspath(__file__))`.

### Data Filtering:

- **Stock and Sales Price:** Products are filtered to ensure both "Mevcut Stok" (stock) and "Mevcut Satış Fiyatı" (sales price) are greater than 0.
- **Price Difference:** Products are filtered based on the price difference condition:
  - If the difference between "Mevcut Satış Fiyatı" and "Maksimum Girebileceğin Fiyat" divided by "Mevcut Satış Fiyatı" exceeds the threshold ( `MAX_PRICE_DIFFERENCE` ), they are excluded.
- **View Count:** Products are selected only if their corresponding view count ("Toplam Görüntülenme Sayısı") is below the `VIEWED` threshold.

### Price Adjustment:

- The "Kampanyalı Satış Fiyatı" column is filled with the "Maksimum Girebileceğin Fiyat" value for each selected product.

- This step aligns with the goal of preparing the product data for a campaign by adjusting the prices accordingly.

## Saving Results:

- The filtered product data is saved into a new Excel file (`filtered_products.xlsx`).
  - The file is saved in the `kampanya` folder, ensuring the output is stored in a dedicated location.
  - The script also provides a log message indicating the path where the output file is saved.
- 

## Usage Instructions:

### 1. Prepare the Files:

- Ensure the following files are in the same directory as the script:
  - `favori-görüntüleme-raporu.xlsx` (the favorite viewing report).
  - Product files in the `indirilen_dosya_kampanya` folder.

### 2. Run the Script:

- Execute the script. It will process the product data based on the conditions specified and create a filtered output file in the `kampanya` folder.

### 3. Check the Logs:

- Review the log file (`kampanya_script_output.log`) for detailed information about the script's operations, including any errors encountered.

### 4. Output:

- The filtered products will be saved as `filtered_products.xlsx` in the `kampanya` folder.
  - You can also view the first few rows of the filtered products in the terminal.
- 

## Example Output:

The final output might look like this in the terminal:

```
2024-12-22 14:07:31,345 - INFO - File loaded successfully into
df_favorite. 2024-12-22 14:07:32,123 - INFO - File loaded
successfully into df_products. 2024-12-22 14:07:35,567 - INFO -
Filtered products based on stock and price, remaining products:
120 2024-12-22 14:07:40,678 - INFO - Filtered products based on
price difference, remaining products: 100 2024-12-22 14:07:45,234
- INFO - Filtered products based on view count, remaining
products: 80 2024-12-22 14:08:00,345 - INFO - Filtered products
saved to kampanya/filtered_products.xlsx 2024-12-22 14:08:05,234
- INFO - Filtered and sorted products: 2024-12-22 14:08:06,123 -
INFO - Ürün Adı Mevcut Stok Mevcut Satış Fiyatı Maksimum
Girebileceğin Fiyat Kampanyalı Satış Fiyatı 0 Product A 10 200.0
190.0 190.0 1 Product B 15 150.0 140.0 140.0 ...
```

---

## Notes:

- Ensure the product and favorite report files have the required columns ("Mevcut Stok", "Mevcut Satış Fiyatı", etc.).
- Adjust `VIEWED` and `MAX_PRICE_DIFFERENCE` as needed to fit specific filtering requirements.
- The script is designed for flexible use and can be adapted to different data formats as long as the required columns are present.

## Cron Job Setup for Automating Python Scripts

This documentation provides a step-by-step guide to automate the execution of your Python scripts once a week using **cron jobs** on a Unix-like system (such as Linux or macOS).

### Objective:

To run the following Python scripts **every Sunday at 2:00 AM**:

1. **Script 1:** Handles price calculations and updates.
2. **Script 2:** Filters product details and updates Excel files.
3. **Script 3:** Filters campaign-related products and creates a new report.

Each script will output logs for monitoring execution.

---

# 1. Verify Python Script Execution

Before automating the scripts, ensure they execute correctly manually:

- **Test Script 1:** Check that it processes price data, handles Excel files, and creates updated files as expected.
  - **Test Script 2:** Validate that product data is filtered properly, and the output is stored in the correct location.
  - **Test Script 3:** Confirm that the campaign report generation works as expected.
- 

## 2. Storing Python Scripts

Assuming your Python scripts are stored in the following directory structure:

- **Scripts Directory:** `/home/burak/scripts/`
  - **Log Directory:** `/home/burak/logs/`
- 

## 3. Open the Crontab Editor

To schedule the cron job, open the crontab editor by running:

```
crontab -e
```

This will open the crontab configuration file in the default editor ( `nano` or `vim` ).

---

## 4. Add Cron Job Entries

In the crontab file, add cron job entries for each Python script to run every **Sunday at 2:00 AM**. Use the following format:

```
0 2 * * 0 /usr/bin/python3 /home/burak/scripts/script1.py >>
/home/burak/logs/script1.log 2>&1 0 2 * * 0 /usr/bin/python3
/home/burak/scripts/script2.py >> /home/burak/logs/script2.log
2>&1 0 2 * * 0 /usr/bin/python3 /home/burak/scripts/script3.py >>
/home/burak/logs/script3.log 2>&1
```

## Explanation of Cron Syntax ( 0 2 \* \* 0 ):

- 0 : Minute (0th minute).
  - 2 : Hour (2:00 AM).
  - \* : Every day of the month (no specific day).
  - \* : Every month (no specific month).
  - 0 : Sunday (week starts on 0 for Sunday).
- 

## 5. Log Output and Error Handling

To ensure the output and errors are captured, each command redirects both **standard output** and **standard error** to log files:

- >> /home/burak/logs/scriptX.log : Appends the standard output (stdout) of the script to a log file.
- 2>&1 : Redirects **standard error** (stderr) to the same log file, so both output and errors are captured in one place.

Thus, each script will log its execution to:

- /home/burak/logs/script1.log
  - /home/burak/logs/script2.log
  - /home/burak/logs/script3.log
- 

## 6. Saving the Cron Job

Once the cron jobs are added, save and exit the editor:

- In nano : Press Ctrl + X , then press Y to confirm saving, and hit Enter .
  - In vim : Press :wq and hit Enter .
- 

## 7. Verify Cron Jobs

To ensure your cron jobs were added correctly, list the active cron jobs:

```
crontab -l
```

This will show all scheduled cron jobs, allowing you to confirm the entries for your Python scripts.

---

## 8. Ensure Python Path in Cron Job

By default, cron jobs don't inherit the same environment variables as your shell session. Therefore, it's important to use the **absolute path** for `python3`.

To confirm the path to Python 3 on your system, run:

```
which python3
```

This will return the full path, e.g., `/usr/bin/python3`.

Make sure to replace `/usr/bin/python3` in the cron job if your Python executable resides elsewhere (such as `/usr/local/bin/python3`).

---

## 9. Optional Environment Setup for Virtual Environments

If you're using a **Python virtual environment** for dependencies, you can activate it before running the script. For example:

```
0 2 * * 0 cd /home/burak/scripts && source
/home/burak/.venv/bin/activate && /usr/bin/python3 script1.py >>
/home/burak/logs/script1.log 2>&1
```

This configuration:

- Changes to the `/home/burak/scripts` directory.
- Activates the Python virtual environment ( `source /home/burak/.venv/bin/activate` ).
- Executes `script1.py`.

Ensure you have the correct virtual environment path.

---

## 10. Cron Job Example (Final)



If:

- **Python scripts** are stored in `/home/burak/scripts/`.
- **Python 3** is located at `/usr/bin/python3`.
- **Log files** will be saved in `/home/burak/logs/`.

The cron jobs in the crontab file should look like this:

```
0 2 * * 0 /usr/bin/python3 /home/burak/scripts/script1.py >>
/home/burak/logs/script1.log 2>&1 0 2 * * 0 /usr/bin/python3
/home/burak/scripts/script2.py >> /home/burak/logs/script2.log
2>&1 0 2 * * 0 /usr/bin/python3 /home/burak/scripts/script3.py >>
/home/burak/logs/script3.log 2>&1
```

This ensures that the scripts will run **every Sunday at 2:00 AM** and log output/error details in the specified log files.

---

## 11. Check Cron Job Execution

To confirm that your cron jobs are executing correctly:

- Monitor the log files for output:
- `tail -f /home/burak/logs/script1.log` `tail -f /home/burak/logs/script2.log` `tail -f /home/burak/logs/script3.log`
- Check system logs for cron job execution:
- `tail -f /var/log/syslog` # For Ubuntu/Debian systems `tail -f /var/log/cron.log` # For RedHat/CentOS systems

---

## Conclusion

This process automates the execution of your Python scripts on a weekly schedule, with output and errors being logged for transparency. By using cron jobs, you ensure that your data processing, Excel file handling, and campaign reporting tasks are automated and run seamlessly every week without manual intervention.

## Main Workstation

- **Hardware:** A powerful workstation with sufficient resources for running larger models like Granite LLaMA and Meta's LLaMA.
- **Software:** Docker containers for easy management and deployment, particularly for hosting LLaMA models and Open-WebUI, with services exposed for LAN access.
- **Model Usage:** Utilizing large language models like Granite LLaMA and Meta's LLaMA for various tasks and hosting them via Docker.

## Lenovo Device (12GB RAM)

- **Use Case:** A lightweight machine, leveraged for running less resource-demanding models. The goal is to host an LLM for local, small-scale operations.

## 2. LLM Hosting (Main Workstation with LAN Access)

The **Open-WebUI** Docker service on the main workstation enables running large models, which can be accessed via LAN for remote use. The models being run include Granite LLaMA and Meta's models, which are suitable for local, high-performance use cases.

### Steps for Hosting on LAN (with Open-WebUI):

1. **Run Docker Container** for Open-WebUI with LAN access:
2. `docker run -d --name open-webui -p 3000:8080 open-webui/open-webui:main`
3. **Access the Open-WebUI** on any device within the LAN:
  - Open a browser and navigate to `http://<Workstation_IP>:3000`.
  - This allows interaction with the hosted LLaMA models.

## 3. Lightweight LLM Options for Lenovo Device (12GB RAM)

Given the Lenovo device has limited RAM (12GB), the goal is to run lighter LLM models efficiently.

### Top 2 Lightweight Model Options:

1. **DistilGPT-2:**

- A distilled version of GPT-2, optimized for smaller models with reduced memory footprint while maintaining decent performance.
- **RAM Usage:** Around 2GB for inference.

## 2. TinyLLaMA:

- A smaller and optimized version of the LLaMA model, offering a balance between memory efficiency and performance.
- **RAM Usage:** Around 4GB for inference, leaving headroom for other tasks.

Both models are ideal for usage on devices with limited resources like the Lenovo machine with 12GB of RAM.