

Lab #6

Deploying a Private AI Model with Ollama and  
OpenWebUI

Week #12/13

NACT-262 Fundamentals of System Administration  
2025-2026 Fall Semester

Submitted by Jibreal Id-deen

Due by November 23, 2025

Professor Mark Jeremy

# Table Of Contents

Objective ..... 6

Network DIAGRAM ..... 7

Procedure ..... 9

Questions and Answers..... 36

Observations ..... 37

## Table of Figures

Figure 1 - Type "Rufus" on google url .....	9
Figure 2 - Download it.....	9
Figure 3 - type "ubuntu Server" on google url to download.....	10
Figure 4 - Download it.....	10
Figure 5 - Use Ubuntu Server to start booting on the USB .....	11
Figure 6 - Install Ubuntu Server on PC2 .....	12
Figure 7 - install OpenSSH Server.....	12
Figure 8 - type the required command to update the Ubuntu Server system.....	13
Figure 9 - type the required command to install ollama on Ubuntu Server.....	13
Figure 10 – Downloading .....	13
Figure 11 - Reboot the Ubuntu Server system.....	13
Figure 12 - After rebooting, type the requiredcommand to start the ollama service .....	13
Figure 13 - type the required command to enable ollama service .....	13
Figure 14 - type the required command to pull the AI model from olamma.....	14
Figure 15 - Pulling .....	14
Figure 16 - Make sure to see the "Success" word after pulling the AI model from ollama .....	14
Figure 17 - type the required command to run the AI model, mistral .....	14
Figure 18 - Mistral AI mode now .....	14
Figure 19 - Type the required command to 'watch' AI mistral .....	14
Figure 20 - Make sure ollama service is running to respond while you make sure to watch the memory raise up.....	15
Figure 21 - Type the required command to install docker.....	15
Figure 22 - Type the required command to start the docker service .....	15
Figure 23 - Type the required command to enable the docker service .....	15
Figure 24 - Type the required command to run the container of the docker .....	16
Figure 25 - Type the required command to allow ufw 8080 to be on .....	16

Figure 26 - Type the required command to reload the firewall .....	16
Figure 27 - Type the required command to check the firewall status.....	16
Figure 28 - Type the DHCP IP Address on the google url .....	17
Figure 29 - The showcase of the openwebui .....	17
Figure 30 - Create Admin Account.....	18
Figure 31 - Click "Admin" panel .....	19
Figure 32 - Click Settings then connections, and change the API connections.....	20
Figure 33 - Change the host url.....	21
Figure 34 - Make sure to use the localhost as the url.....	22
Figure 35 - Save it.....	22
Figure 36 - Successfully API settings updated .....	23
Figure 37 - Add User .....	23
Figure 38 - Create an standard account.....	24
Figure 39 - Click "Groups" .....	25
Figure 40 - Create Group .....	25
Figure 41 - Describe the group .....	26
Figure 42 - Make sure to work on the permissions for the standard account to have some permissions and not.....	27
Figure 43 - Save it.....	28
Figure 44 - Click "Test-Player" to drag to the group.....	28
Figure 45 - Successfully added the standard account to the group.....	29
Figure 46 - Make sure to check the ubuntu's IP Address.....	29
Figure 47 - Type the required command to change the Ubuntu Server's IP Address.....	29
Figure 48 - Set the required configuration manually to change the IP Address .....	30
Figure 49 - Type the required command to apply the netplan to confirm changes .....	30
Figure 50 - Make sure to reboot the ubuntu server .....	31
Figure 51 - Check the Ubuntu Server's IP Address again.....	31
Figure 52 - Make sure to start the container of the docker once again .....	31
Figure 53 - Type the static IP Address on the Google url .....	31
Figure 54 - Sign the openwenUI account .....	32
Figure 55 - Successfully logged in.....	32

Figure 56 - Type "ollama.ai" website on the google url .....	32
Figure 57 - Click "Models" on Ollama website.....	33
Figure 58 - Type the command to run the another AI model .....	33
Figure 59 - Type the command to run the llama.....	33
Figure 60 - Type the command to run phi4 .....	34
Figure 61 - Test the three AI models to see how fast they respond.....	35

# Lab #6 – Deploying a Private AI Model with Ollama and OpenWebUI

## **OBJECTIVE**

---

Set up a private AI chat stack on your LAN by installing Ubuntu + Ollama with OpenWebUI on PC2 (RTX GPU), then secure it (static IP, firewall, authentication) and verify it works by pulling multiple models and comparing their responses.

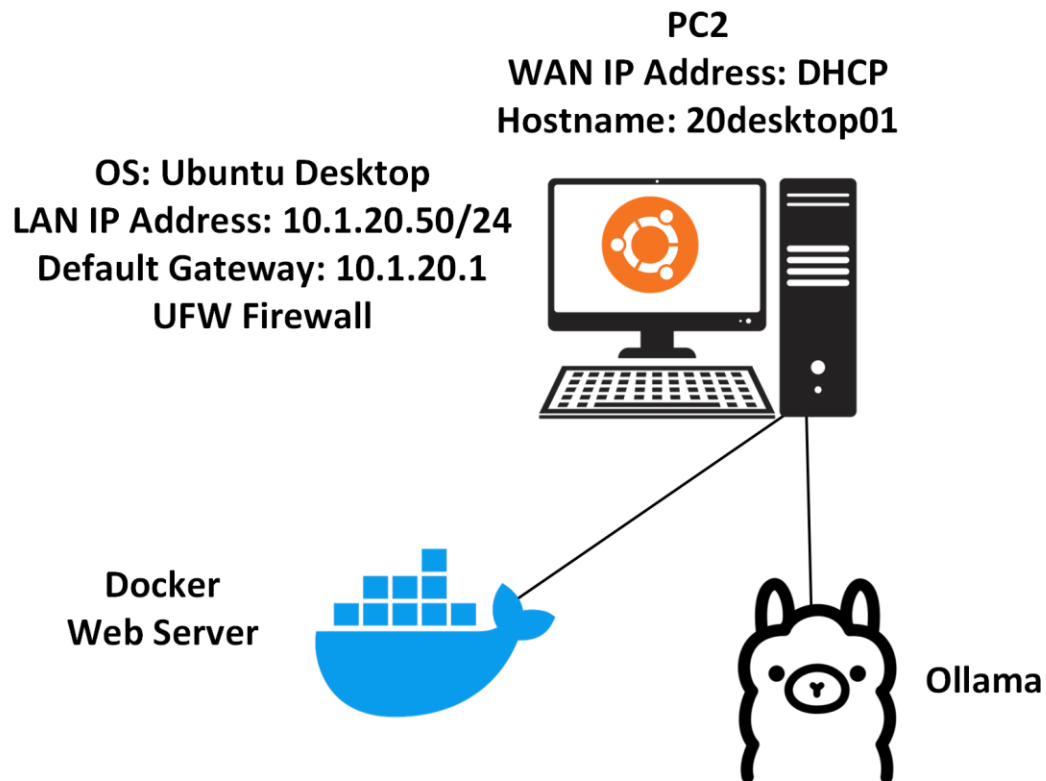
## NETWORK DIAGRAM

---



**Lab 6 – Deploying a Private AI Model with Ollama and OpenWebUI**  
**NACT-262 Fundamentals of Systems Administration**  
**November 23, 2025**

**Professor Jeremy**  
**By Jibreal Id-deen**



Hostname	Ai01
Virtual Machine or Physical?	Physical
Operating System	Ubuntu Server 24.04 LTS
RAM size	16GB
HD size	1TB
Ethernet #1 IP Address (if none – remove line)	10.1.20.60
Ethernet Gateway IP Address (if none – remove line)	10.1.20.1
Primary DNS IP Address (if none – remove line)	8.8.8.8
Network Services (if none – remove line)	Ollama, OpenWebUI, SSH



## PROCEDURE

---

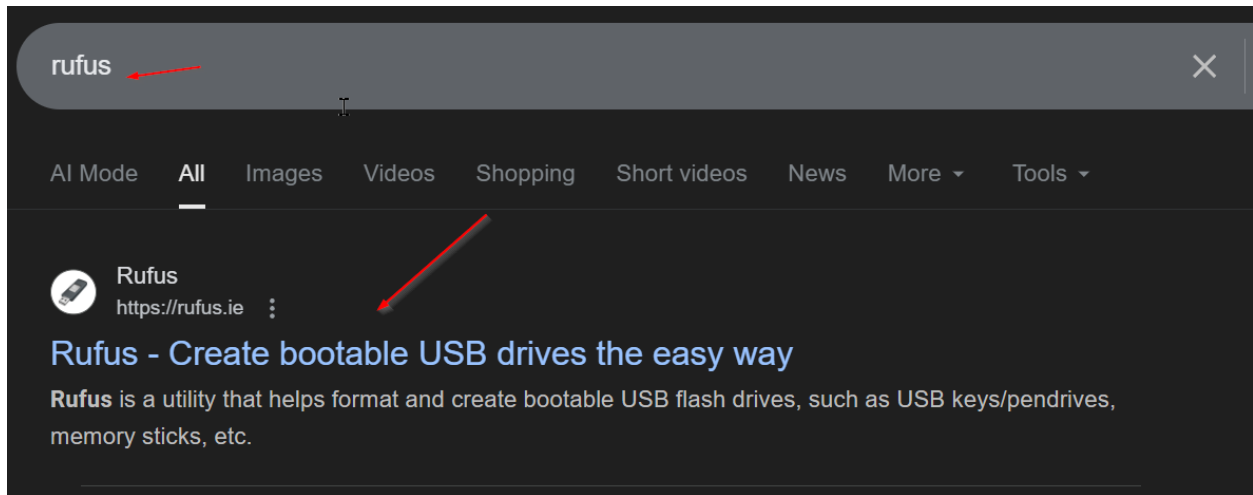


FIGURE 1 - TYPE "RUFUS" ON GOOGLE URL

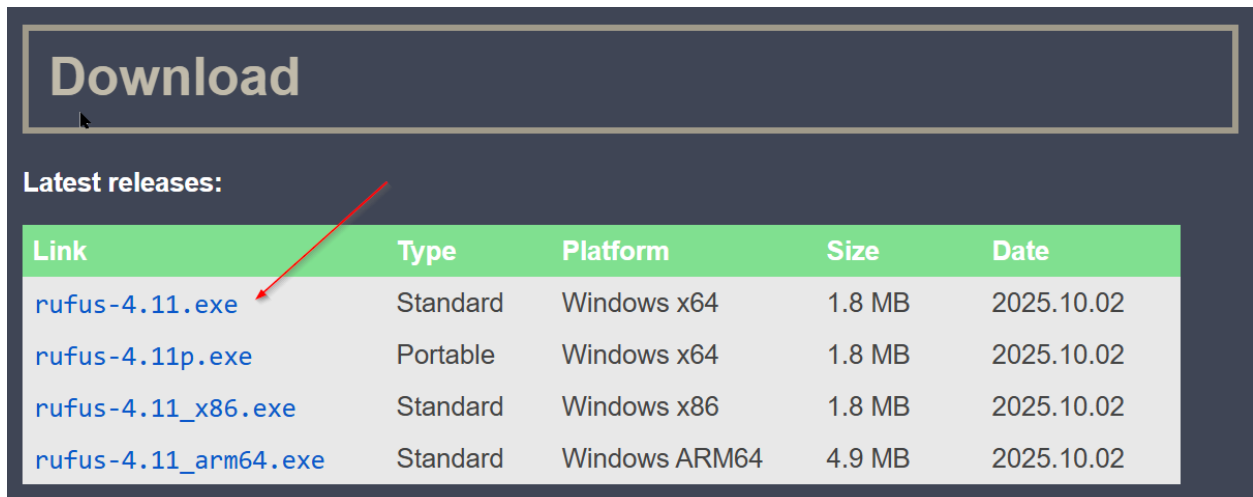
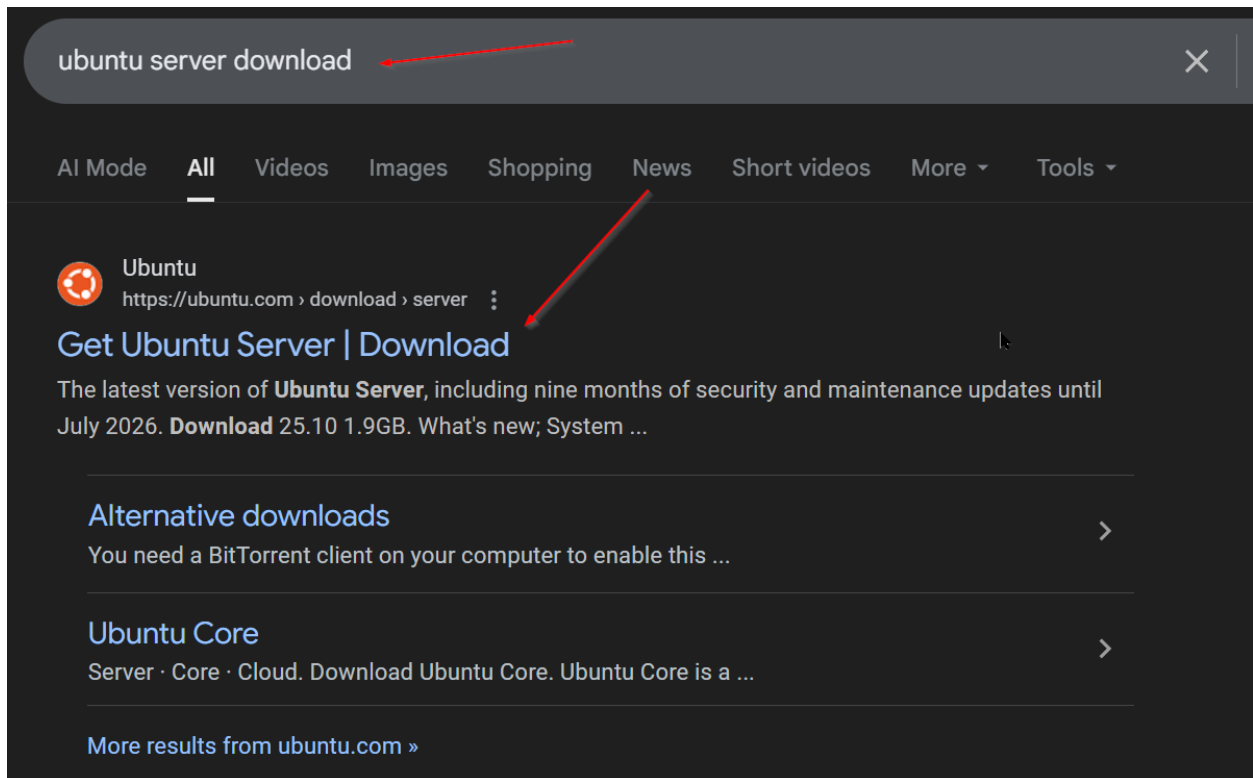
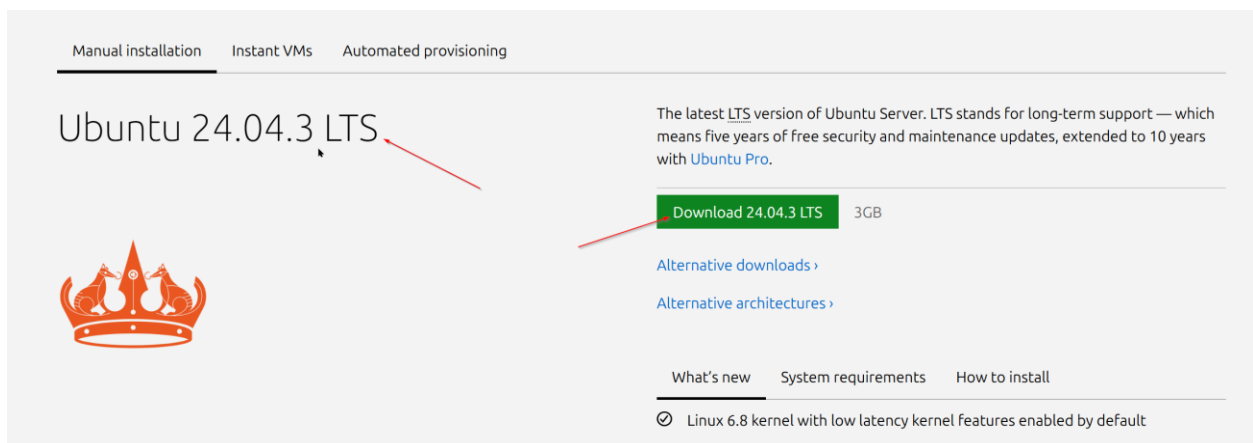


FIGURE 2 - DOWNLOAD IT



**FIGURE 3 - TYPE "UBUNTU SERVER" ON GOOGLE URL TO DOWNLOAD**



**FIGURE 4 - DOWNLOAD IT**

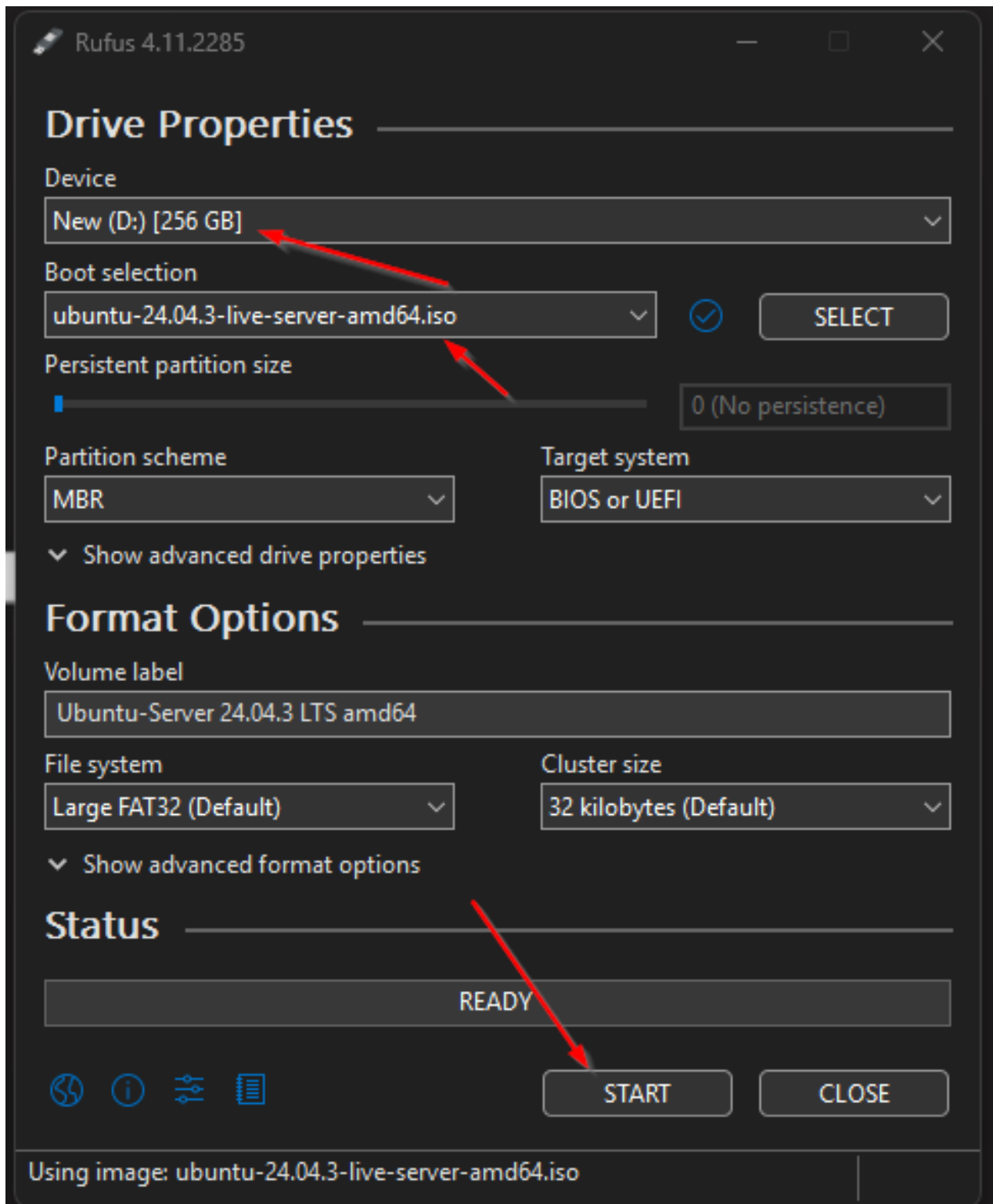


FIGURE 5 - USE UBUNTU SERVER TO START BOOTING ON THE USB

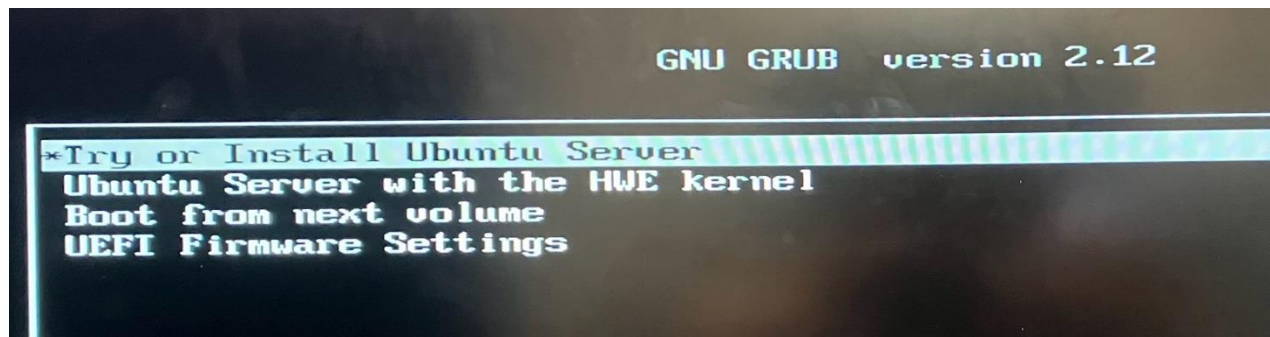


FIGURE 6 - INSTALL UBUNTU SERVER ON PC2

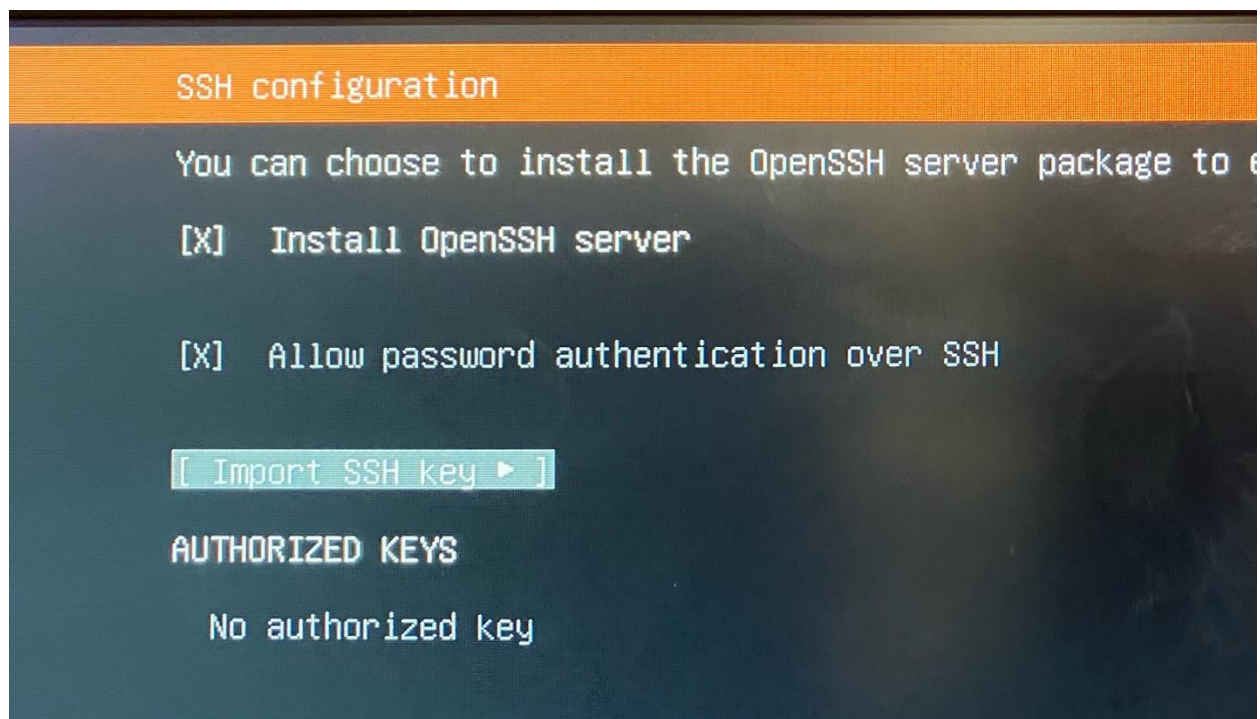


FIGURE 7 - INSTALL OPENSSH SERVER

```
cold@ai01:~$ sudo apt update & sudo apt upgrade -y
```

FIGURE 8 - TYPE THE REQUIRED COMMAND TO UPDATE THE UBUNTU SERVER SYSTEM

```
cold@ai01:~$ curl -fsSL https://ollama.com/install.sh | sh
```

FIGURE 9 - TYPE THE REQUIRED COMMAND TO INSTALL OLLAMA ON UBUNTU SERVER

```
>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 bundle
##### 16.4%
```

FIGURE 10 – DOWNLOADING

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
linux-headers-6.8.0-87-generic is already the newest version (6.8.0-87.88).
linux-headers-6.8.0-87-generic set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
>>> Reboot to complete NVIDIA CUDA driver install.
cold@ai01:~$
```

FIGURE 11 - REBOOT THE UBUNTU SERVER SYSTEM

```
cold@ai01:~$ sudo systemctl start ollama
```

FIGURE 12 - AFTER REBOOTING, TYPE THE REQUIRED COMMAND TO START THE OLLAMA SERVICE

```
cold@ai01:~$ sudo systemctl enable ollama
```

FIGURE 13 - TYPE THE REQUIRED COMMAND TO ENABLE OLLAMA SERVICE

```
cold@ai01:~$ ollama pull mistral
```

FIGURE 14 - TYPE THE REQUIRED COMMAND TO PULL THE AI MODEL FROM OLLAMA

```
pulling manifest
pulling f5074b1221da: 14% ▢
```

FIGURE 15 - PULLING

```
pulling f5074b1221da: 100% ▢ 4.4 GB
pulling 43070e2d4e53: 100% ▢ 11 KB
pulling 1ff5b64b61b9: 100% ▢ 799 B
pulling ed11eda7790d: 100% ▢ 30 B
pulling 1064e17101bd: 100% ▢ 487 B
verifying sha256 digest
writing manifest
success
cold@ai01:~$
```

FIGURE 16 - MAKE SURE TO SEE THE "SUCCESS" WORD AFTER PULLING THE AI MODEL FROM OLLAMA

```
cold@ai01:~$ ollama run mistral
```

FIGURE 17 - TYPE THE REQUIRED COMMAND TO RUN THE AI MODEL, MISTRAL

```
cold@ai01:~$ ollama run mistral
>>> Send a message (/? for help)
```

FIGURE 18 - MISTRAL AI MODE NOW

```
cold@ai01:~$ ollama run mistral
>>> watch -n 0.5 nvidia-smi
```

FIGURE 19 - TYPE THE REQUIRED COMMAND TO 'WATCH' AI MISTRAL

Every 0.5s: nvidia-smi ai01: Mon Nov 10 23:59:02 2025

Mon Nov 10 23:59:02 2025

NVIDIA-SMI 560.35.05			Driver Version: 560.35.05			CUDA Version: 12.6		
GPU	Name	Perf	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr.	ECC
Fan	Temp		Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute	M. MIG M.
0	NVIDIA GeForce RTX 4060	P0	Off	00000000:01:00.0	Off	0%	Default	N/A
0%	37C		N/A / 115W		5170MiB / 8188MiB			N/A

Processes:							GPU Memory Usage
GPU	GI ID	CI ID	PID	Type	Process name		
0	N/A	N/A	1554	C	/usr/local/bin/ollama		5160MiB

**FIGURE 20 - MAKE SURE OLLAMA SERVICE IS RUNNING TO RESPOND WHILE YOU MAKE SURE TO WATCH THE MEMORY RAISE UP**

```
cold@ai01:~$ sudo apt install docker.io -y
```

**FIGURE 21 - TYPE THE REQUIRED COMMAND TO INSTALL DOCKER**

```
cold@ai01:~$ sudo systemctl start docker
```

**FIGURE 22 - TYPE THE REQUIRED COMMAND TO START THE DOCKER SERVICE**

```
cold@ai01:~$ sudo systemctl enable docker
```

**FIGURE 23 - TYPE THE REQUIRED COMMAND TO ENABLE THE DOCKER SERVICE**

```
cold@cold:~$ sudo docker run -d --network host -p 8080:8080 -v open-webui:/app/backend/data --name open-webui ghcr.io/open-webui/open-webui:main
```

FIGURE 24 - TYPE THE REQUIRED COMMAND TO RUN THE CONTAINER OF THE DOCKER

```
cold@cold:~$ sudo ufw allow 8080
Rule added
Rule added (v6)
cold@cold:~$ |
```

FIGURE 25 - TYPE THE REQUIRED COMMAND TO ALLOW UFW 8080 TO BE ON

```
cold@cold:~$ sudo ufw reload
Firewall reloaded
cold@cold:~$ |
```

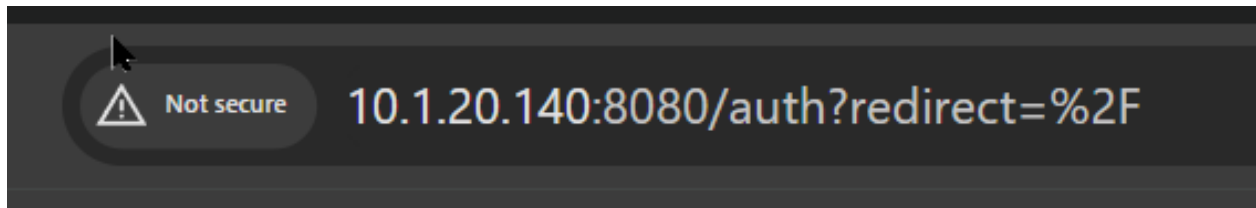
FIGURE 26 - TYPE THE REQUIRED COMMAND TO RELOAD THE FIREWALL

```
cold@cold:~$ sudo ufw status
Status: active
```

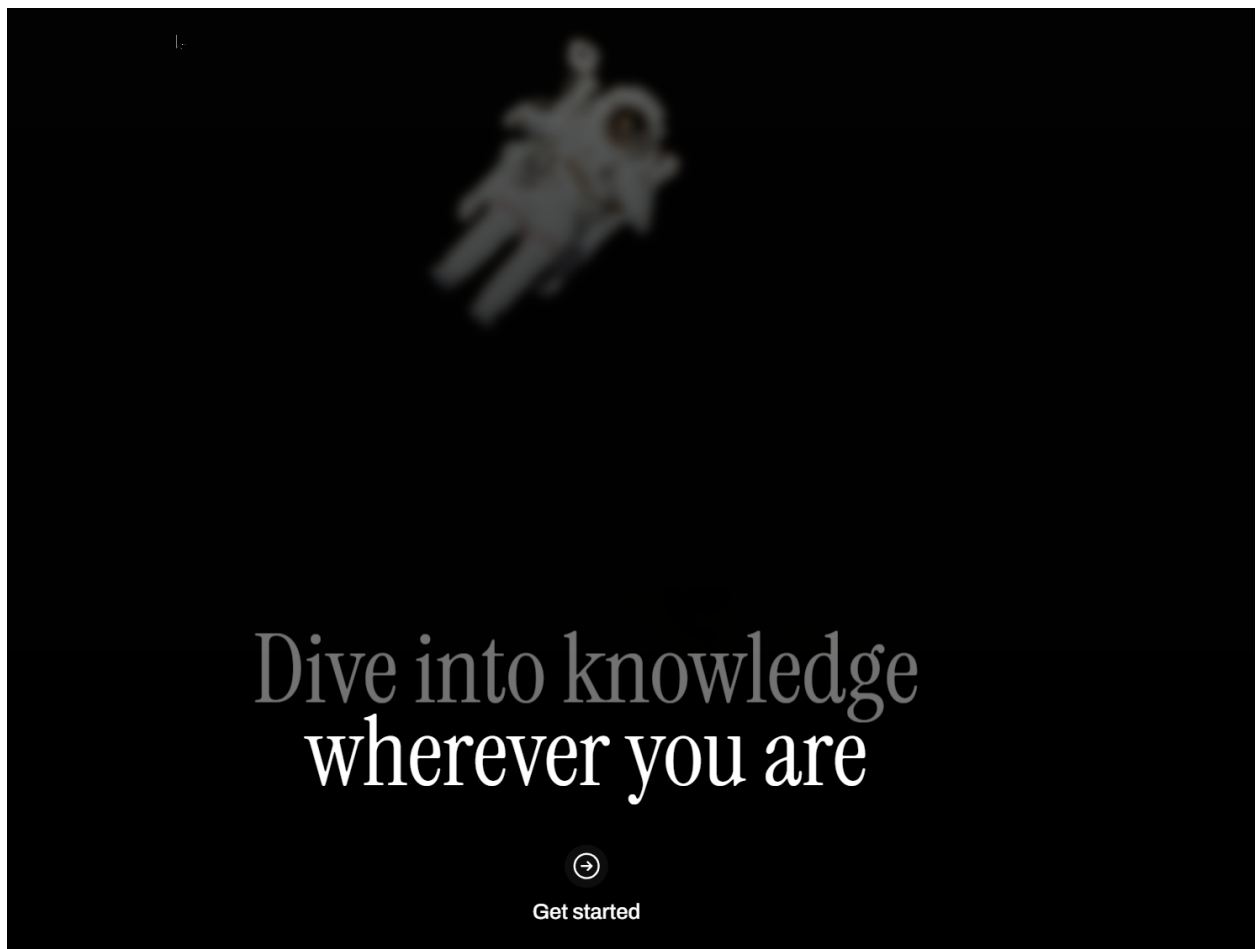
To	Action	From
--	-----	----
22/tcp	ALLOW	Anywhere
8080	ALLOW	Anywhere
22/tcp (v6)	ALLOW	Anywhere (v6)
8080 (v6)	ALLOW	Anywhere (v6)

FIGURE 27 - TYPE THE REQUIRED COMMAND TO CHECK THE FIREWALL STATUS





**FIGURE 28 - TYPE THE DHCP IP ADDRESS ON THE GOOGLE URL**



**FIGURE 29 - THE SHOWCASE OF THE OPENWEBUI**

# Get started with Open WebUI

① Open WebUI does not make any external connections, and your data stays securely on your locally hosted server.

**Name**

Enter Your Full Name

**Email**

Enter Your Email

**Password**

Enter Your Password

👁

Create Admin Account

**FIGURE 30 - CREATE ADMIN ACCOUNT**

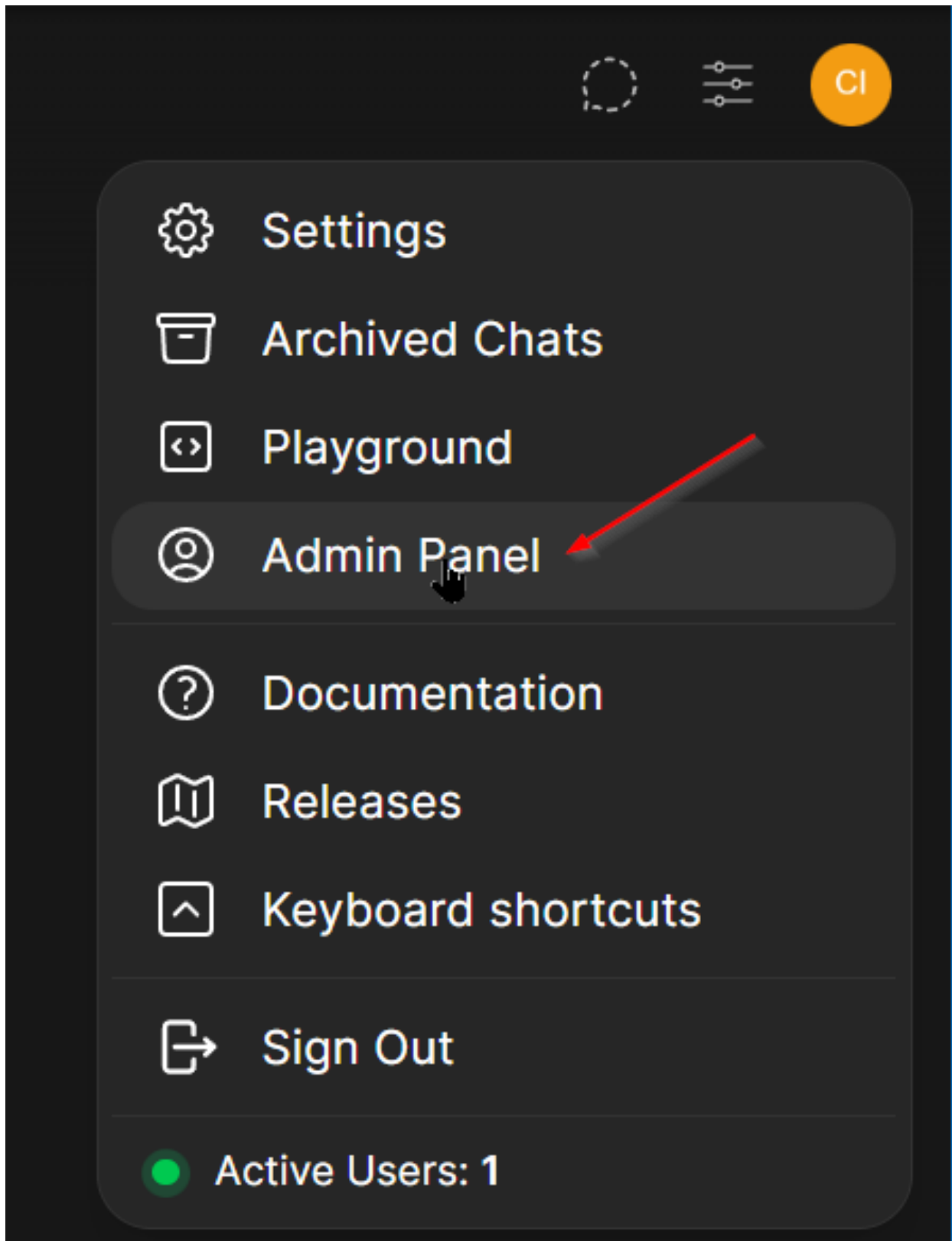
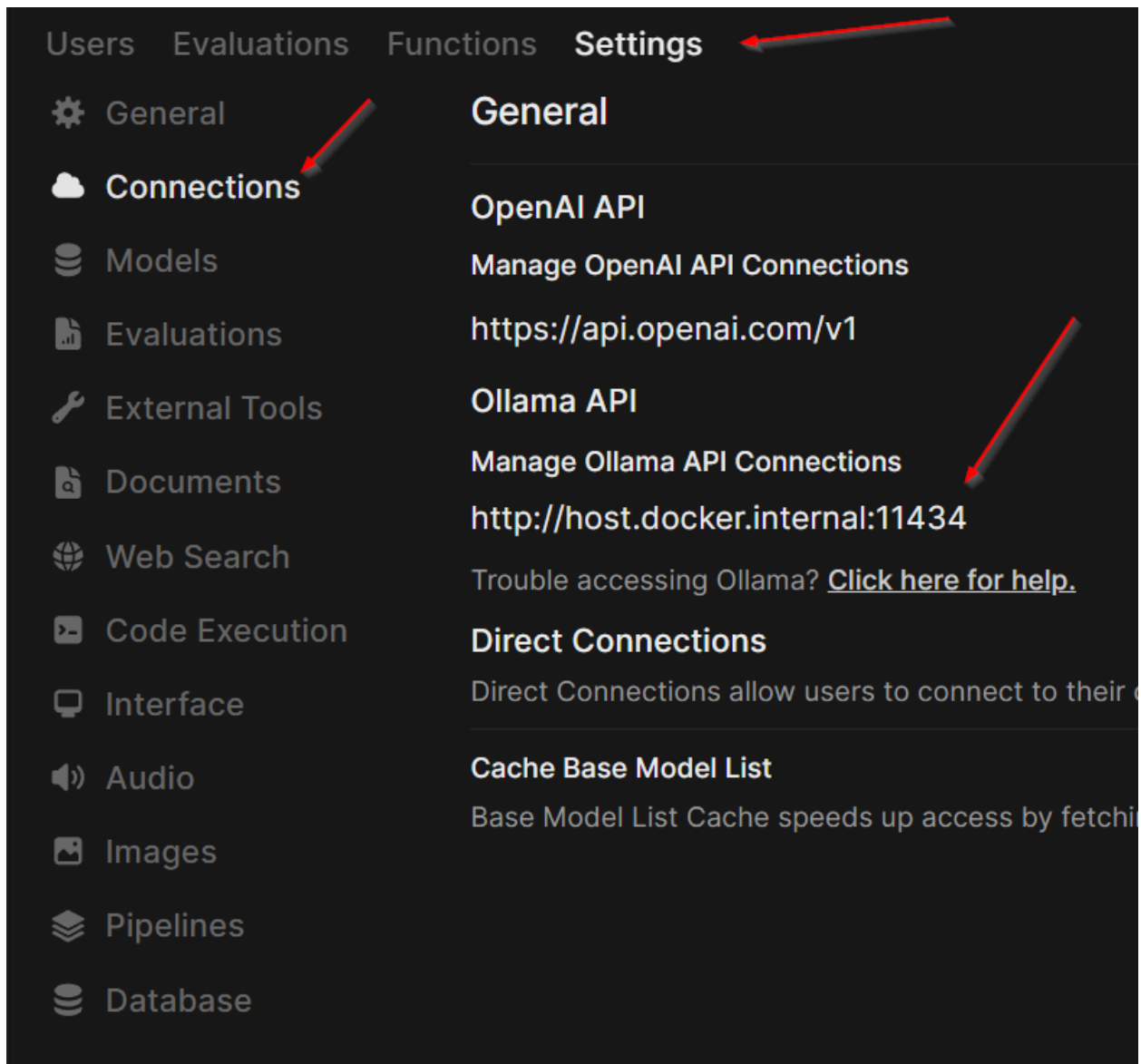


FIGURE 31 - CLICK "ADMIN" PANEL



**FIGURE 32 - CLICK SETTINGS THEN CONNECTIONS, AND CHANGE THE API CONNECTIONS**

## Edit Connection

✕

Connection Type

Local

URL

http://host.docker.internal:11434

↺ ☒

Auth

Bearer 

▼

 API Key

👁

Prefix ID

Prefix ID

Model IDs

Leave empty to include all models from  
"http://host.docker.internal:11434/api/tags" endpoint

Add a model ID 

+

Tags

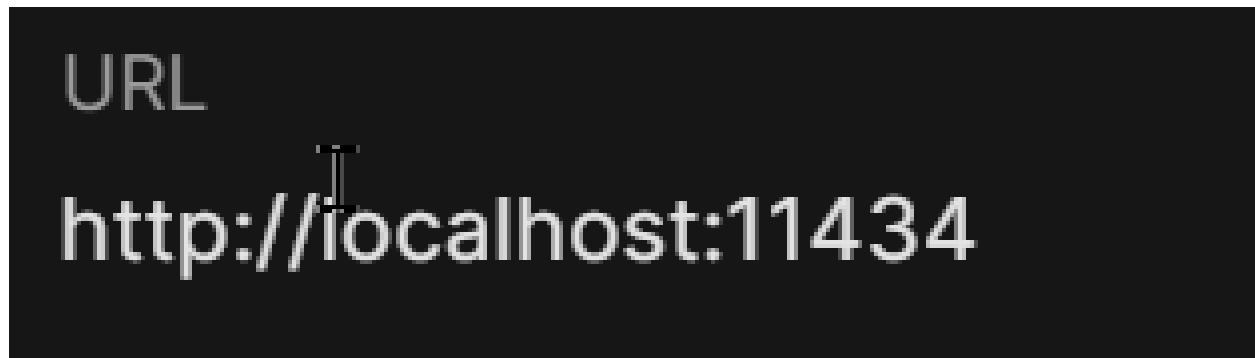
+

 Add Tags

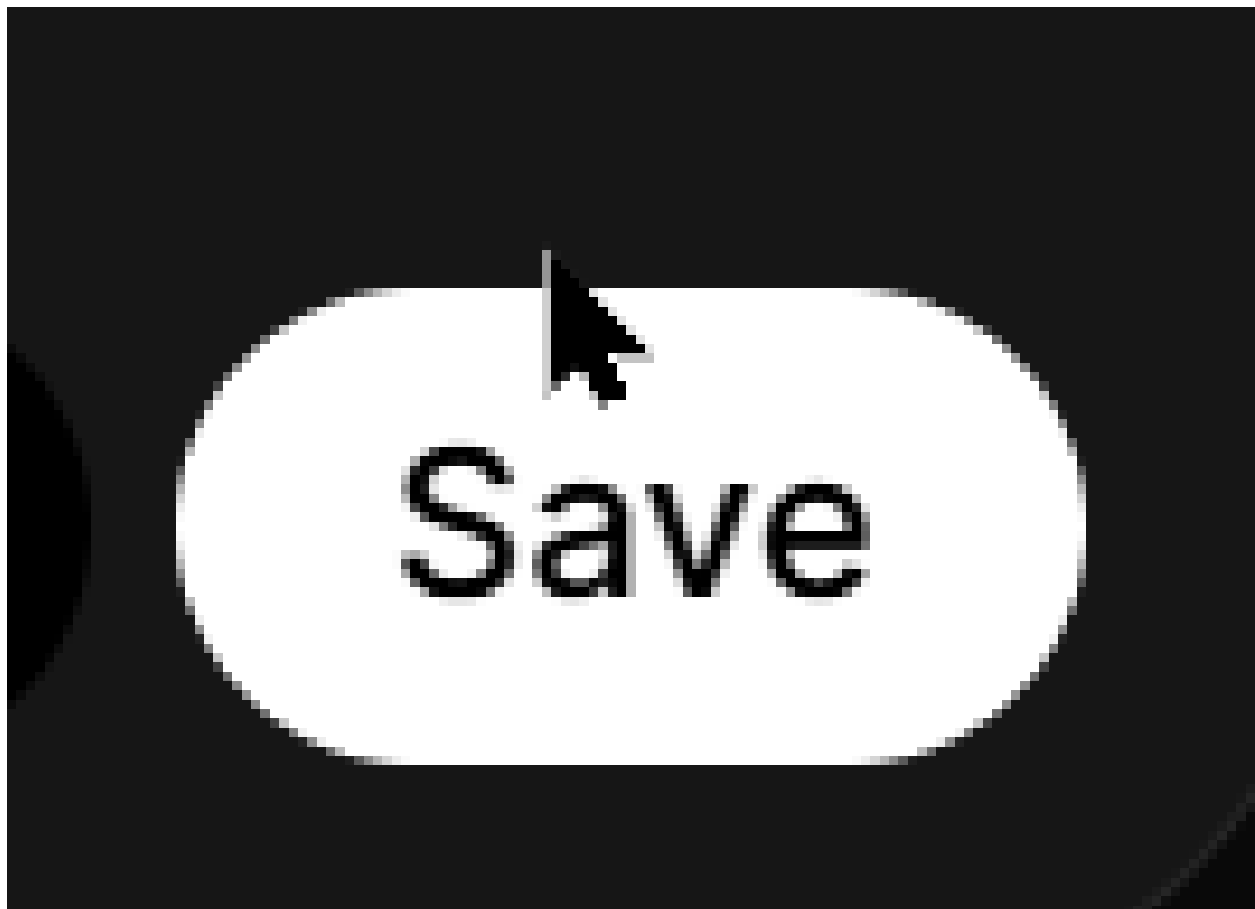
Delete

Save

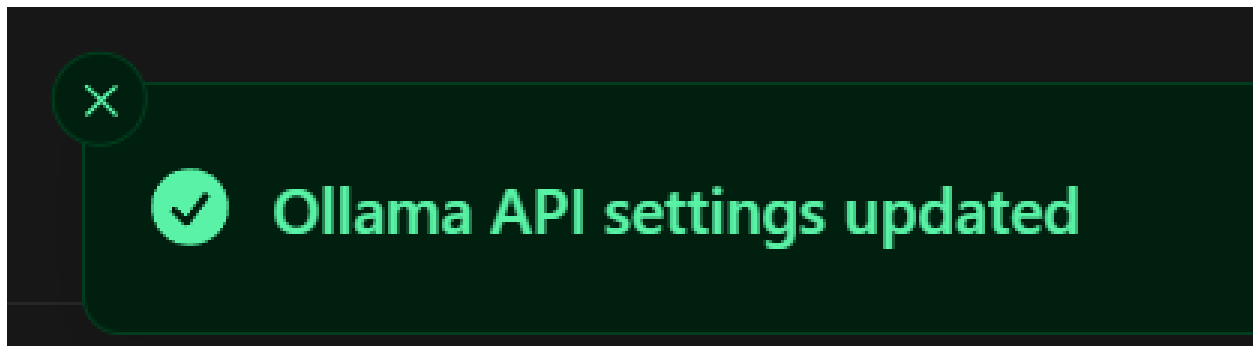
FIGURE 33 - CHANGE THE HOST URL



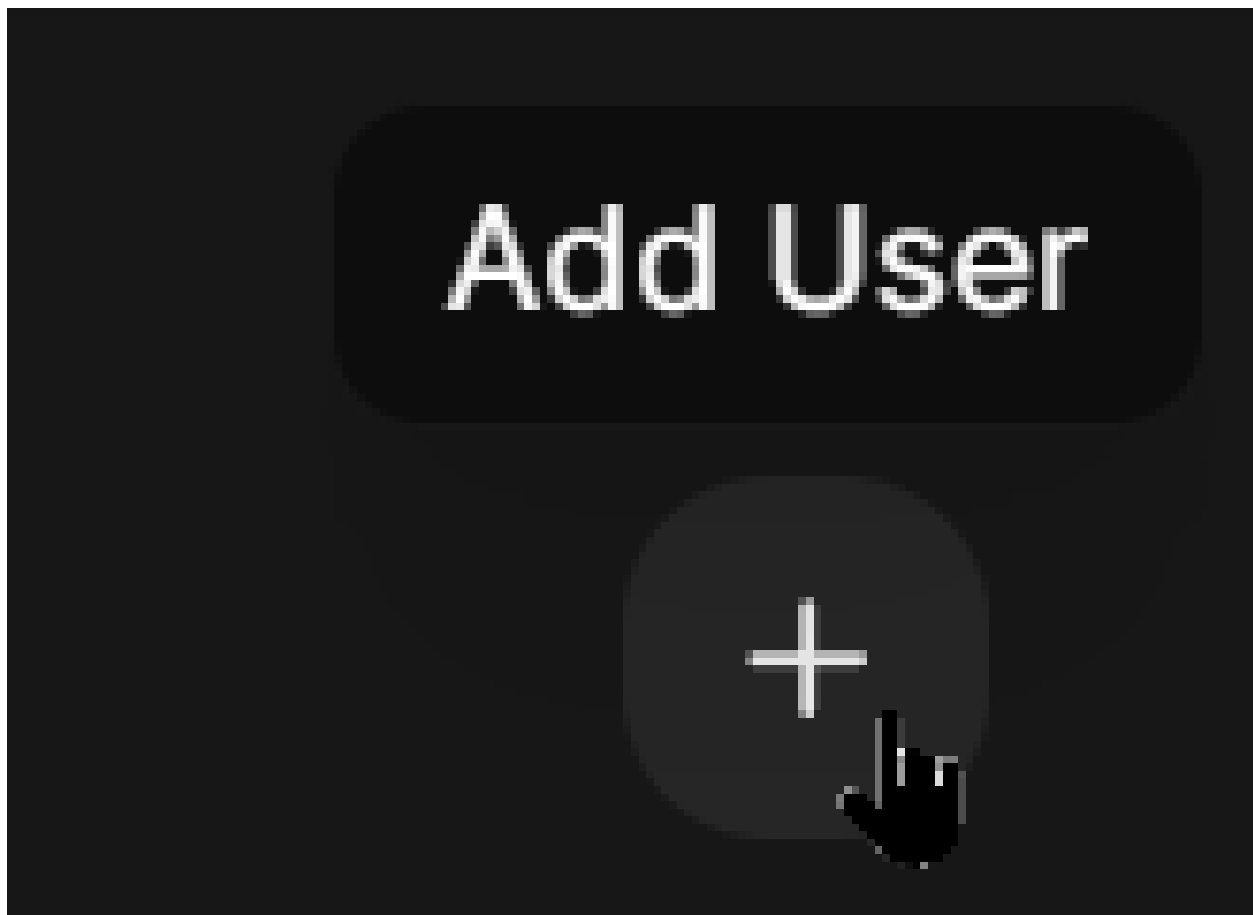
**FIGURE 34 - MAKE SURE TO USE THE LOCALHOST AS THE URL**



**FIGURE 35 - SAVE IT**



**FIGURE 36 - SUCCESSFULLY API SETTINGS UPDATED**



**FIGURE 37 - ADD USER**

# Edit User

T

Test-Player

Created at November 17, 2025

User Groups

Users

Role

User

Name

Test-Player

Email

testemail@gmail.com

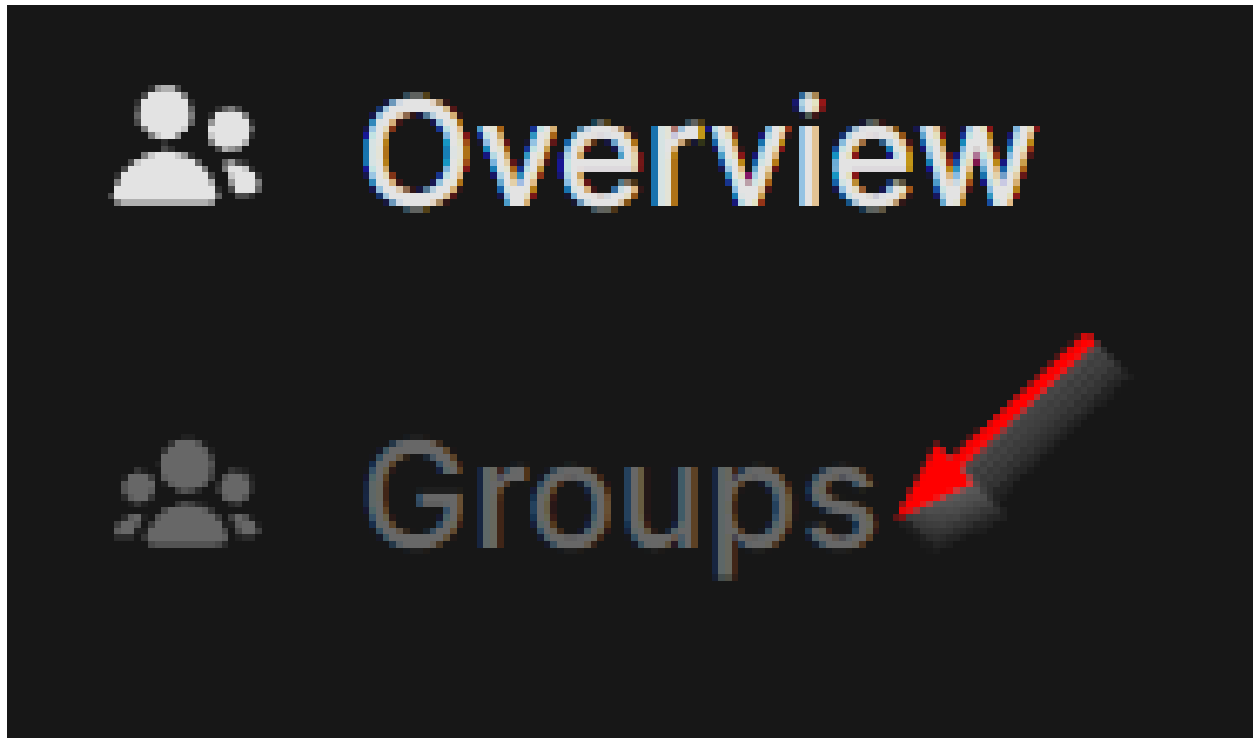
New Password

Enter New Password

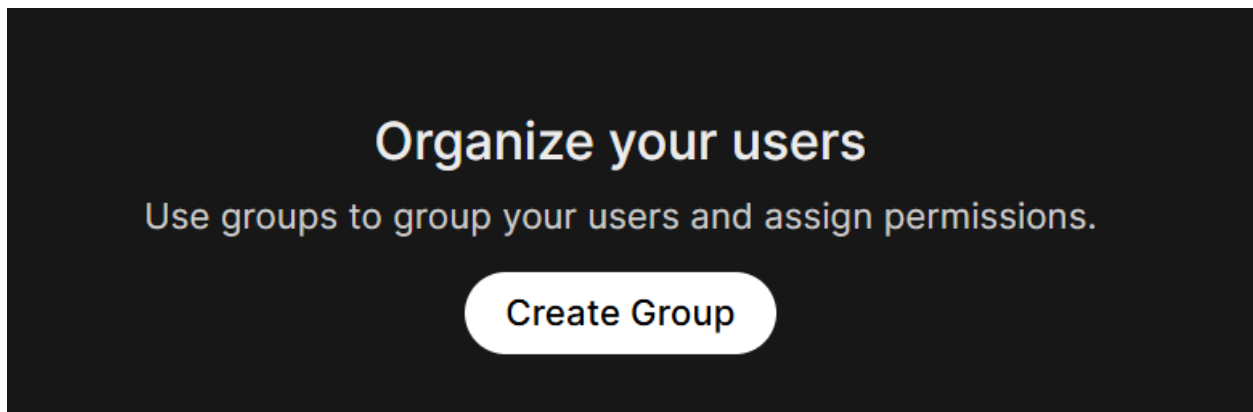
Save

FIGURE 38 - CREATE AN STANDARD ACCOUNT







**FIGURE 39 - CLICK "GROUPS"**




**FIGURE 40 - CREATE GROUP**

# Add User Group

 General

 Permissions

 Users (0)

Name

User Group

Description

Limit/Ban

FIGURE 41 - DESCRIBE THE GROUP

✕

### Workspace Permissions

Models Access

☐

Knowledge Access

☐

Prompts Access

☐

Tools Access

☐

### Sharing Permissions

Models Public Sharing

☐

Knowledge Public Sharing

☐

Prompts Public Sharing

☐

Tools Public Sharing

☐

Notes Public Sharing

☐

### Chat Permissions

**FIGURE 42 - MAKE SURE TO WORK ON THE PERMISSIONS FOR THE STANDARD ACCOUNT TO HAVE SOME PERMISSIONS AND NOT**

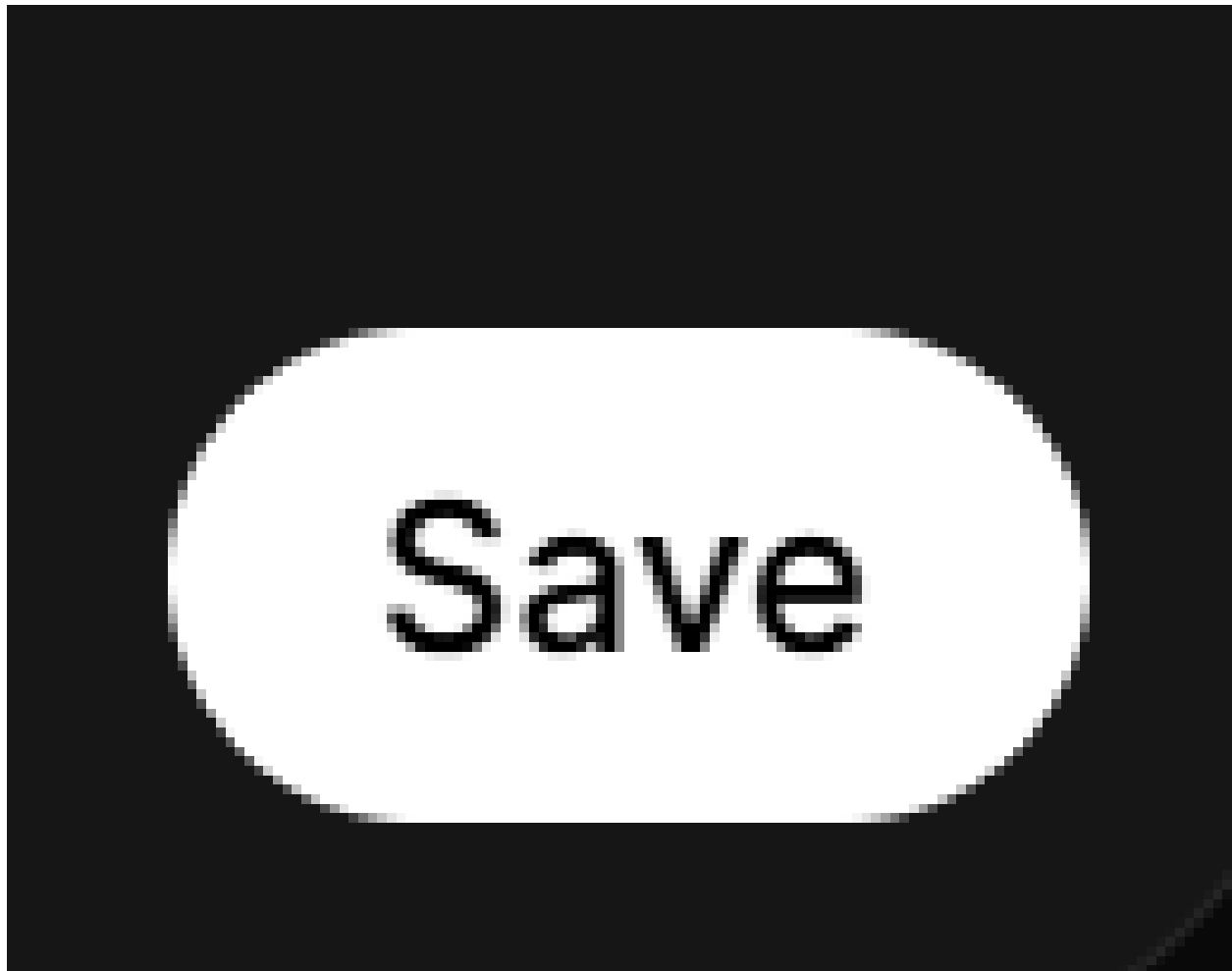


FIGURE 43 - SAVE IT

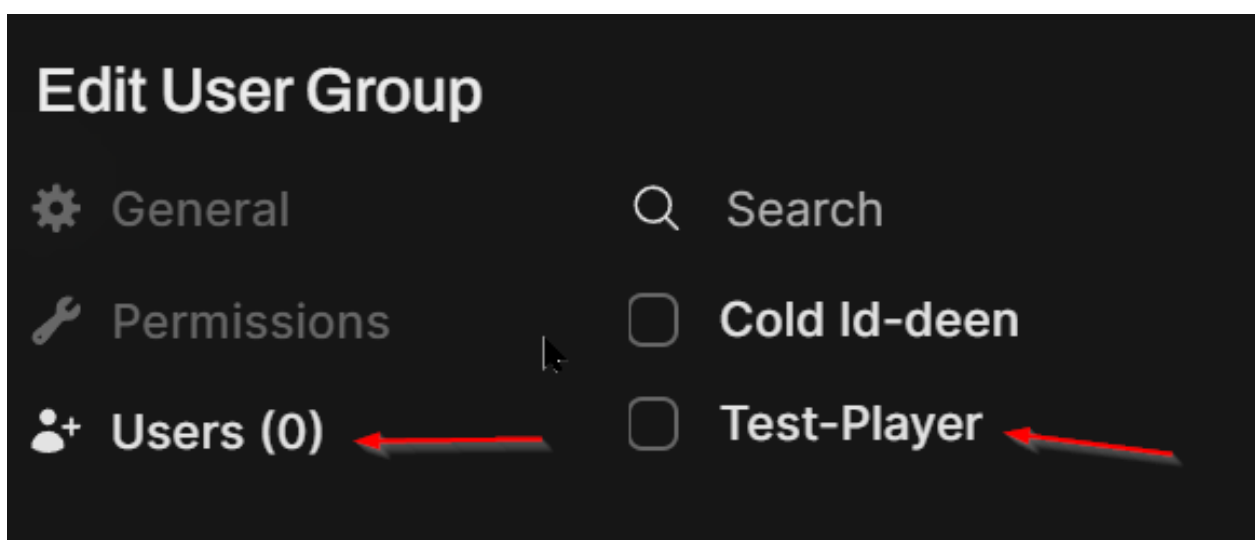


FIGURE 44 - CLICK "TEST-PLAYER" TO DRAG TO THE GROUP

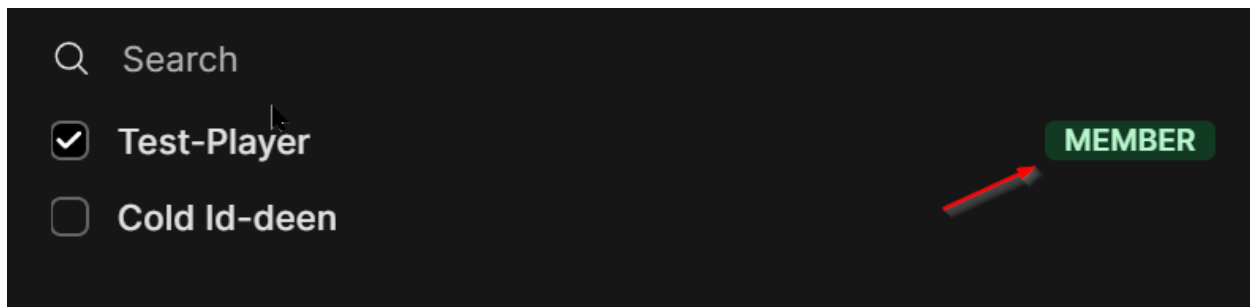


FIGURE 45 - SUCCESSFULLY ADDED THE STANDARD ACCOUNT TO THE GROUP

```
cold@cold:~$ 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: enp9s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 40:16:7e:7a:ab:03 brd ff:ff:ff:ff:ff:ff
   inet 10.1.20.140/24 metric 100 brd 10.1.20.255 scope global dynamic enp9s0
       valid_lft 5285sec preferred_lft 5285sec
   inet6 fe80::4216:7eff:fe7a:ab03/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
   link/ether 6e:5d:5e:aa:79:b6 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
cold@cold:~$
```

FIGURE 46 - MAKE SURE TO CHECK THE UBUNTU'S IP ADDRESS

```
cold@cold:~$ sudo nano /etc/netplan/50-cloud-init.yaml
```

FIGURE 47 - TYPE THE REQUIRED COMMAND TO CHANGE THE UBUNTU SERVER'S IP ADDRESS

```
GNU nano 7.2
network:
  version: 2
  ethernet:
    enp9s0:
      dhcp4: false
      addresses:
        - 10.1.20.60/24
      routes:
        - to: default
          via: 10.1.20.1
      nameservers:
        addresses:
          - 8.8.8.8
```

FIGURE 48 - SET THE REQUIRED CONFIGURATION MANUALLY TO CHANGE THE IP ADDRESS

```
cold@cold:~$ sudo netplan apply
```

FIGURE 49 - TYPE THE REQUIRED COMMAND TO APPLY THE NETPLAN TO CONFIRM CHANGES

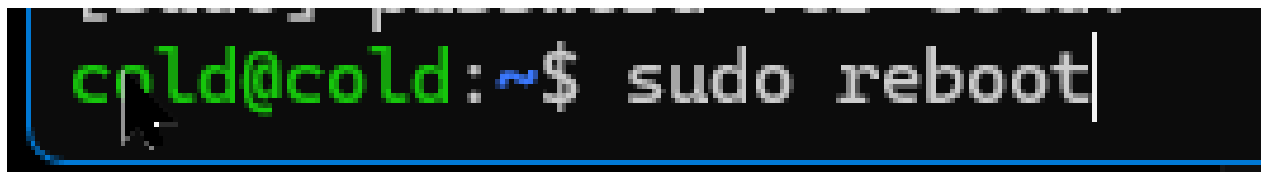


FIGURE 50 - MAKE SURE TO REBOOT THE UBUNTU SERVER

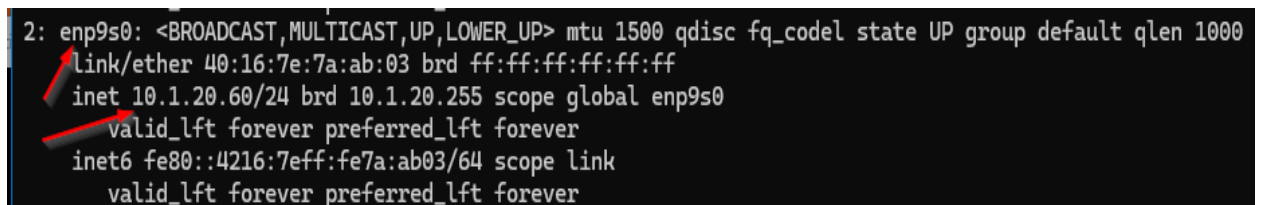


FIGURE 51 - CHECK THE UBUNTU SERVER'S IP ADDRESS AGAIN

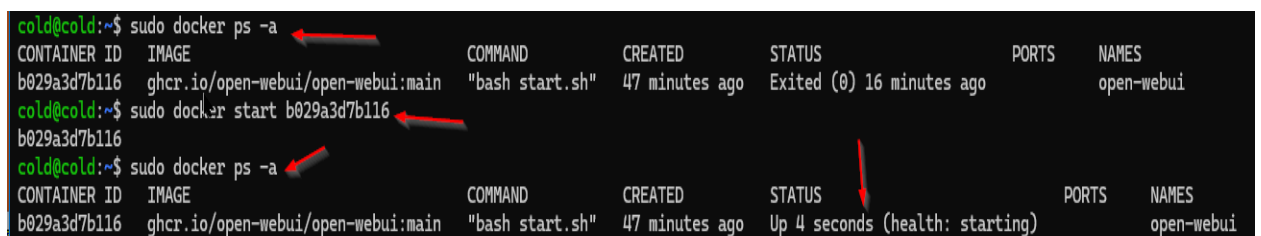


FIGURE 52 - MAKE SURE TO START THE CONTAINER OF THE DOCKER ONCE AGAIN

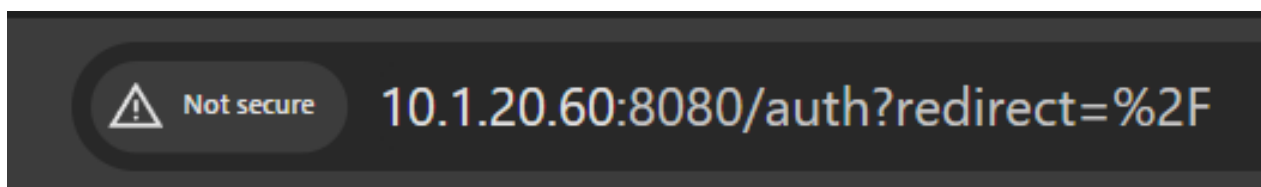
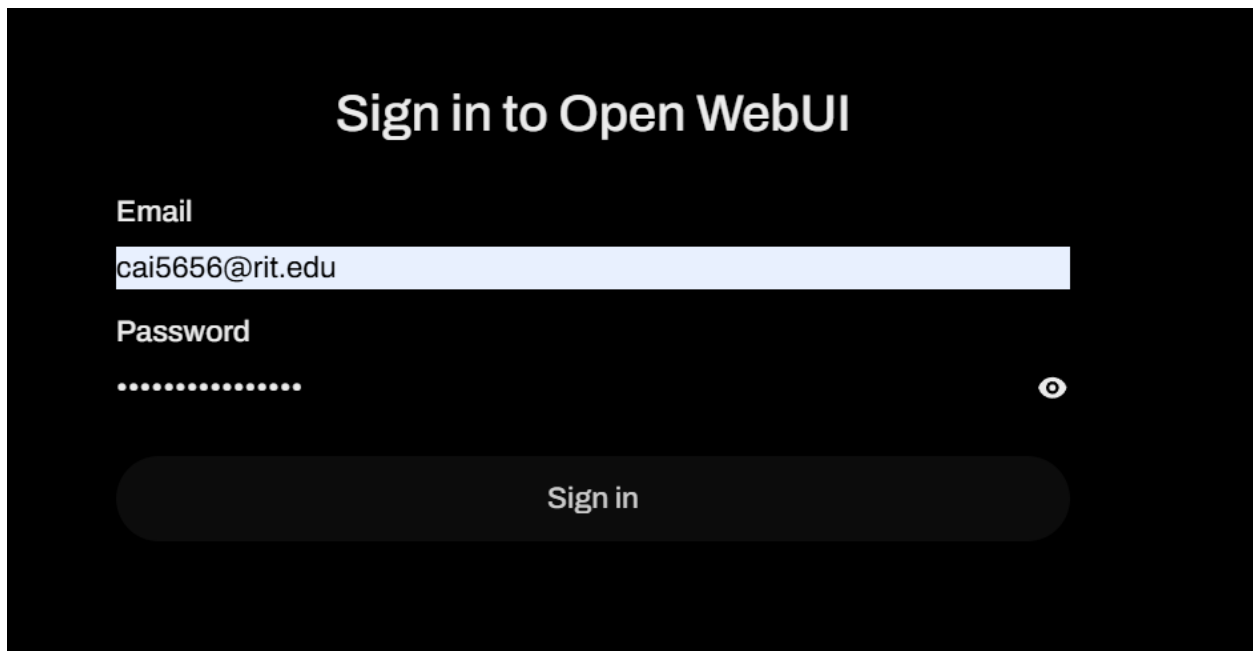


FIGURE 53 - TYPE THE STATIC IP ADDRESS ON THE GOOGLE URL



Sign in to Open WebUI

Email

cai5656@rit.edu

Password

.....

Sign in

FIGURE 54 - SIGN THE OPENWENUI ACCOUNT

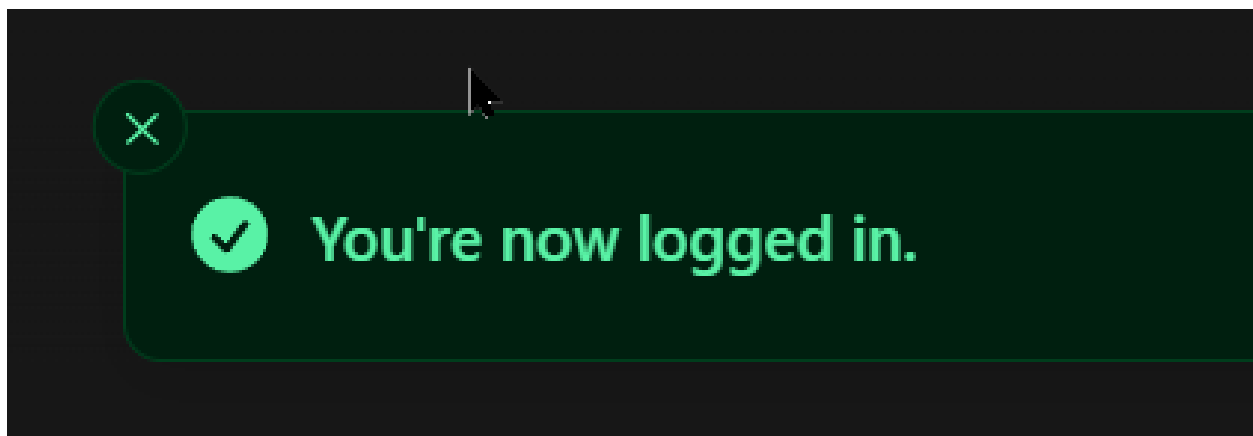


FIGURE 55 - SUCCESSFULLY LOGGED IN

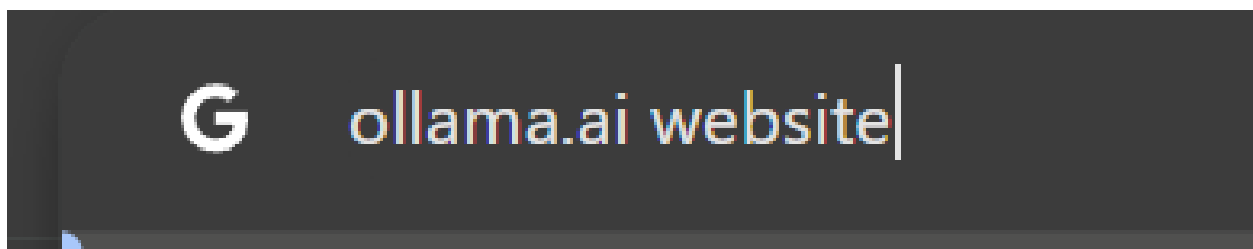


FIGURE 56 - TYPE "OLLAMA.AI" WEBSITE ON THE GOOGLE URL





Models

GitHub

Discord

Docs

Cloud

FIGURE 57 - CLICK "MODELS" ON OLLAMA WEBSITE



## llama3.1

Llama 3.1 is a new state-of-the-art model from Meta available in 8B, 70B and 405B parameter sizes.

tools

8b

70b

405b



105.9M Pulls



93 Tags



Updated 11 months ago

## llama3.1



105.9M Downloads



Updated 11 months ago

Llama 3.1 is a new state-of-the-art model from Meta available in 8B, 70B and 405B parameter sizes.

tools

8b

70b

405b

ollama run llama3.1



FIGURE 58 - TYPE THE COMMAND TO RUN THE ANOTHER AI MODEL

```
cold@cold:~$ ollama run llama3.1
```

FIGURE 59 - TYPE THE COMMAND TO RUN THE LLAMA

## phi4

Phi-4 is a 14B parameter, state-of-the-art open model from Microsoft.

14b



6.1M Pulls



5 Tags



Updated 10 months ago

```
cold@cold:~$ ollama run phi4
```

FIGURE 60 - TYPE THE COMMAND TO RUN PHI4

01

phi4:latest

## System Administrator Role: Long-Term vs. Short-Term Perspectives

### Introduction

The role of a System Administrator (SysAdmin) is critical for managing, maintaining, and ensuring the smooth operation of an organization's IT infrastructure. The responsibilities and focus areas may vary depending on whether we are considering short-term or long-term perspectives.

### Long-Term Perspective

#### Strategic Planning

- **Infrastructure Growth:** Plan for scalable solutions to accommodate future growth in user base, data volume, and application demands.
- **Technology Roadmap:** Develop a technology roadmap that aligns with the organization's strategic goals, ensuring that systems are future-proofed against obsolescence.

Today at 5:28 PM

xplain what is the system adminstarior in long term and long parapaha as if you are writing document

01

llama3.1:latest

**Document Title:** System Administrator Role: Long-Term vs. Short-Term Perspectives

**Document ID:** SYSADM-LTST-PERSPECTIVES-001

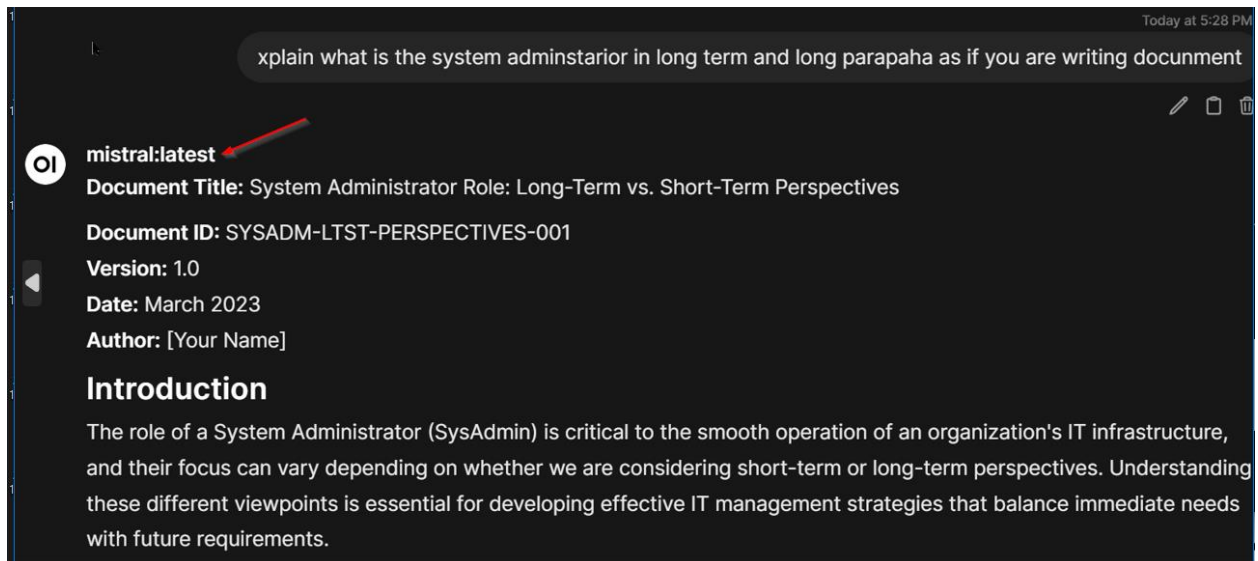
**Version:** 1.0

**Date:** March 2023

**Author:** [Your Name]

### Overview

This document outlines the roles and responsibilities of a System Administrator (SysAdmin) from both long-term and short-term perspectives. Understanding these differing viewpoints is crucial for implementing effective IT management strategies that balance immediate needs with future requirements.



**FIGURE 61 - TEST THE THREE AI MODELS TO SEE HOW FAST THEY RESPOND**

## QUESTIONS AND ANSWERS

---

- What are the benefits of running a private AI model?
  - You keep data on your own network, so privacy is strong. Latency is low and cost is predictable. You can customize models and work offline.
- How would you secure this service in a production environment?
  - Use HTTPS, strong login, and a firewall. Put it on its own VLAN or behind a VPN. Keep the OS, Docker, and models updated, and log access.
- What are the limitations of local LLMs compared to cloud-based ones?
  - Big models may be too heavy for your GPU. Quality and features can be lower than top cloud models. Scaling to many users is harder and you must maintain everything.

## **OBSERVATIONS**

---

It was easy to understand, so it made things easier, and I learned that blocking ads can be easier thing to do when you use PI!