Installation & Configuration Guide for Ubuntu+Openclaw

Preface

This customized ISO integrates Ubuntu 24.04.3 Desktop, AMD ROCm acceleration stack, OpenClaw scheduling framework, Llama series models and Qwen- 9B4 large language models for local offline AI inference on AMD GPUs. The whole workflow is divided into preparation → OS installation → ROCm environment validation → OpenClaw initialization → local LLM deployment.

Stage 1 Pre- Installation Preparation

1.1 Hardware Compatibility Requirements

1. GPU: AMD ROCm - supported graphics cards (RX 6000/RX7000 consumer series, MI professional accelerator series); NVIDIA GPUs cannot use ROCm acceleration.
2. Memory: Minimum 16 GB RAM; 32 GB RAM is strongly recommended for running the Qwen-9B4 model smoothly.
3. Storage: Reserve at least 100 GB free disk space for system, dependencies and model cache.

1.2 BIOS Configuration

Enter motherboard BIOS during boot, disable Secure Boot permanently: unsigned ROCm kernel modules will fail to load if Secure Boot remains enabled.

1.3 Bootable USB Creation

1. Prepare a blank USB drive with capacity ≥8 GB.
2. Re- download the full ISO file (your current download task shows Canceled, verify MD5 checksum after download to rule out file corruption).
3. Burn the ISO to USB:
   - Windows: Use Rufus
   - Cross- platform: Use BalenaEtcher

   Select the target ISO and USB drive, then complete burning without extra modification of partition schemes.

Stage 2 Ubuntu OS Installation via Bootable USB

1. Plug in the bootable USB and restart your device; press the boot menu hotkey (F2/F12/Del varies by motherboard brand) to boot from the USB drive.
2. Select your preferred display language on the welcome screen, then click Install Ubuntu.
3. Network setup: Connect wired/Wi- Fi network; tick two options: Download updates while installing Ubuntu and Install third- party software for graphics and Wi- Fi hardware, additional media formats.
4. Disk Partitioning
   - For beginners: Choose Erase disk and install Ubuntu (target your system disk instead of the USB drive, auto partitioning).
   - For advanced users (manual partition):
     - `/boot`: 1 GB, ext4 format
     - `swap`: 16 GB

- `/`: All remaining free space, ext4 format

5. Set timezone to `Shanghai`, create your username and login password, wait 20 – 40 minutes for installation to finish.
6. Unplug the USB drive and reboot to enter the Ubuntu desktop environment.

Stage 3 ROCm Acceleration Environment Verification

3.1 Grant GPU Access Permissions

Open Terminal and run the commands below, reboot to make permissions effective:

```bash
sudo usermod -aG render,video $USER
sudo reboot
```

3.2 Validate Pre- installed ROCm

Run these commands to check hardware recognition:

```bash
Print ROCm version and GPU hardware details
rocminfo
Monitor GPU status, utilization and temperature
rocm-smi
```

If the commands are missing, install official ROCm packages for Ubuntu 24.04 manually:

```bash
wget
https://repo.radeon.com/amdgpu-install/7.0.2/ubuntu/noble/amdgpu-install_7.0.2.70002-1_all.deb
sudo apt install ./amdgpu-install_7.0.2.70002-1_all.deb
sudo apt update && sudo amdgpu-install --usecase=rocm
```

Stage 4 OpenClaw Framework Deployment

4.1 Initialize Python Virtual Environment

```bash
sudo apt install python3-pip python3-venv -y
mkdir ~/ai-work && cd ~/ai-work
python3 -m venv ai-env
source ai-env/bin/activate
pip install --upgrade pip vllm transformers sentencepiece
```

4.2 Pull and Launch OpenClaw

```bash
git clone https://github.com/OpenClaw/OpenClaw.git
cd OpenClaw
```

pip install -r requirements.txt
  Start OpenClaw dashboard service
python main.py
```
Open your browser and visit `http://localhost:18789` to enter the OpenClaw web management panel.
  4.3 Official OpenClaw CLI Initialization (Matching Official Guide)
```bash
  Check OpenClaw version
openclaw --version
  Run environment diagnosis
openclaw doctor
  Install background daemon service
openclaw onboard --install-daemon
  Pair Telegram bot (optional): create bot token via BotFather first
openclaw pairing approve telegram YOUR_BOT_TOKEN
```

  Stage 5 Run Pre- built Llama & Qwen- 9B4 Models with ROCm Acceleration
  Option A: High- performance Inference via vLLM
```bash
  Launch Qwen- 9B- Instruct with AMD ROCm backend
vllm-run Qwen/Qwen-9B-Instruct --device rocm
```
  Option B: Lightweight Inference via Ollama
```bash
curl -fsSL https://ollama.com/install.sh | sudo sh
sudo systemctl enable --now ollama
  Pull and run Qwen 9B locally
ollama run qwen:9b
```

  Stage 6 Common Troubleshooting
1. `rocminfo: command not found`
    Confirm Secure Boot is disabled in BIOS, and your user account has been added to `render` and `video` groups with a reboot applied.
2. Slow model inference
    Disable desktop visual effects and reserve ⩾8 GB VRAM for large model inference.
3. Black screen during OS installation
    Use the pre- bundled AMD driver shipped in the ISO; do not replace it with generic open-source drivers manually.
4. Corrupted ISO file
```

Resume interrupted downloads with multi- thread download tools such as IDM or Xtreme Download Manager, and verify MD5 hash before burning the USB drive.

Final Usage Note

After finishing the above steps, you can perform fully offline local conversation with Llama/Qwen models, with hardware acceleration powered by AMD ROCm.