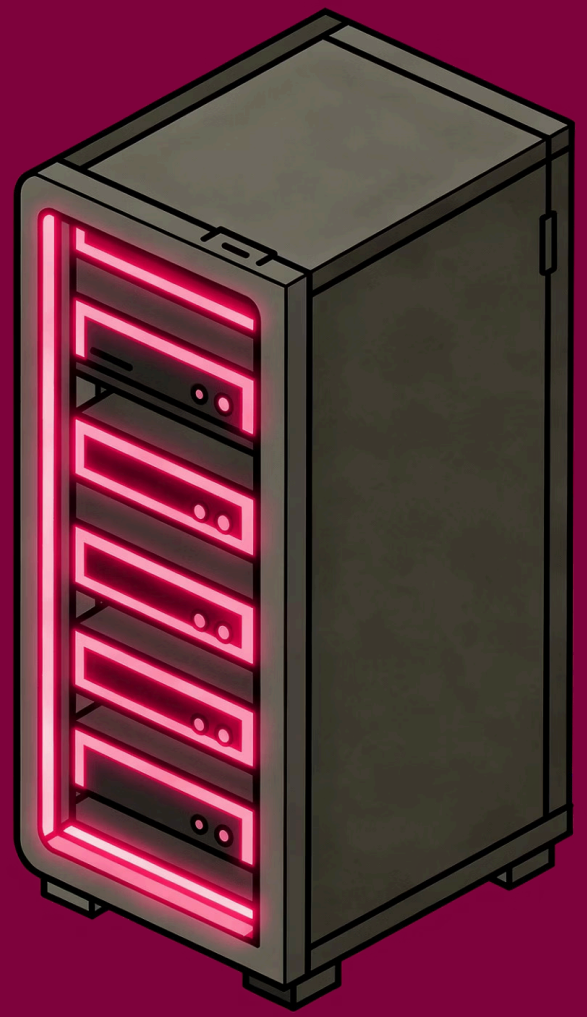


Hands-on Lab: Deploy Local LLM dengan RunPod & OpenClaw

Panduan Praktis Infrastruktur AI Terjangkau

HANDS-ON LAB GPU DEPLOYMENT

Edy Susanto
Independent Researcher in Artificial Intelligence, Cyber Intelligence & Blockchain Systems
Founder – C-SIX Cyber Intelligence Research Lab (CIRL)

🎯 Objektif Lab

Di akhir sesi lab ini, kamu akan memiliki **endpoint API AI yang berdiri sendiri** — dapat diakses langsung dari laptop masing-masing, tanpa bergantung pada layanan cloud berbayar per-token seperti OpenAI.

✅ Deploy

Provisioning GPU instance di RunPod Community Cloud

✅ Install

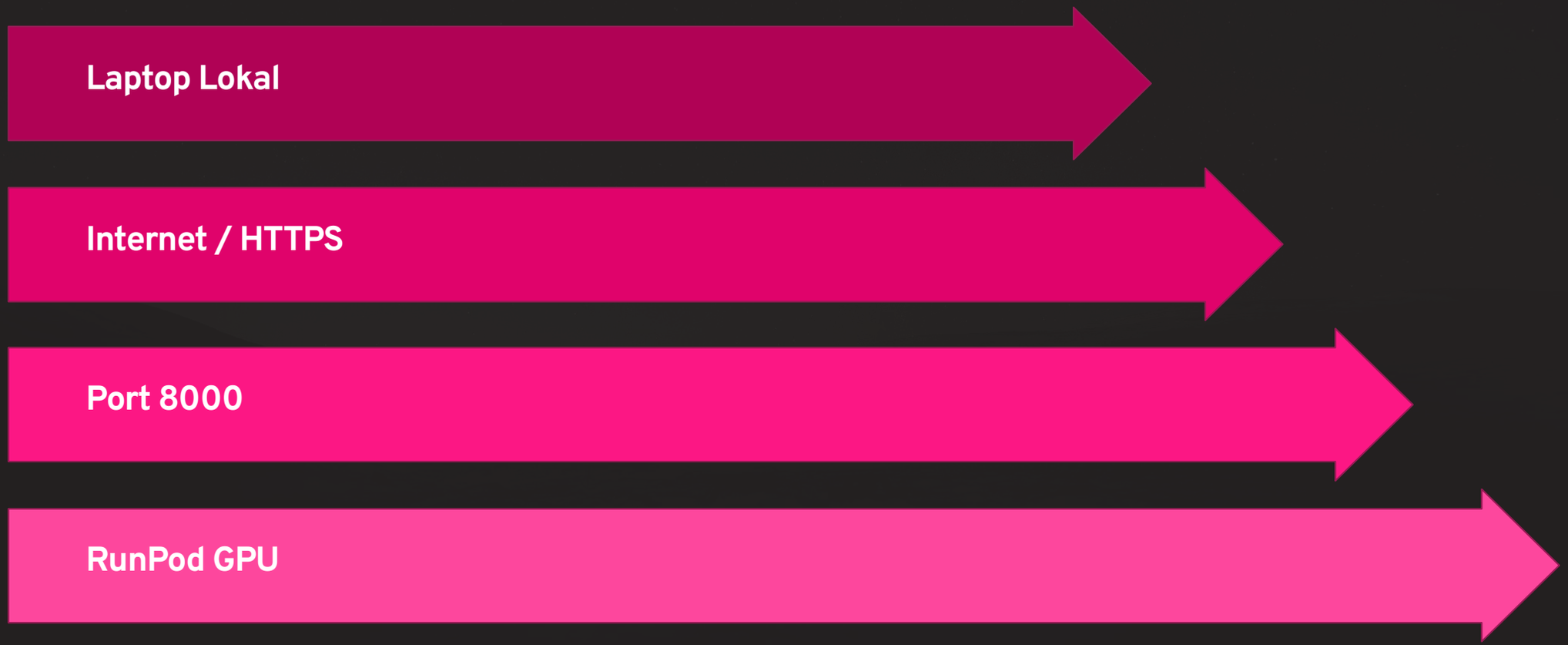
Setup OpenClaw server dan download model Llama-3 GGUF

✅ Test

Hit endpoint API dari terminal lokal dan verifikasi response

🗺️ Arsitektur Lab Kita

Berikut adalah alur lengkap komunikasi dari laptop kamu hingga model AI berjalan di GPU cloud. Pahami setiap komponen sebelum mulai praktik.



Setiap request dari laptop diteruskan melalui internet ke port 8000 yang sudah di-expose pada instance RunPod. OpenClaw bertindak sebagai server inferensi yang memproses prompt dan mengembalikan response dari model Llama-3.

⚙️ Langkah 1: Provisioning GPU di RunPod

Instruksi Step-by-Step

01

Login ke RunPod

Buka **runpod.io** → pilih **Community Cloud** (lebih murah dari Secure Cloud)

02

Pilih GPU

Filter GPU → pilih **RTX 3090 (24GB VRAM)** — cukup untuk Llama-3 8B GGUF

03

Pilih Template

Gunakan template **RunPod PyTorch 2.x** sebagai base image

04

Customize Deployment

⚠️ Klik **"Customize Deployment"** → tambahkan expose port **8000** di bagian HTTP Ports

⚠️ ⚠️ **WAJIB!** Jangan lupa expose Port 8000. Tanpa ini, endpoint API kamu tidak akan bisa diakses dari luar pod. Ini adalah langkah yang paling sering terlewat!

Estimasi biaya RTX 3090 di Community Cloud: **~$0.30–$0.44/jam**. Pastikan kamu punya kredit cukup sebelum mulai lab.

💻 Langkah 2: Akses Web Terminal & Persiapan Environment

Setelah pod berstatus **Running**, klik tombol **"Connect"** → pilih **"Start Web Terminal"**. Kamu akan masuk ke shell Linux di dalam GPU instance. Jalankan perintah berikut untuk mempersiapkan environment:

```
# Update package list dan upgrade sistem
apt update && apt upgrade -y

# Install dependencies yang dibutuhkan
apt install -y python3-pip python3-venv git curl wget

# Verifikasi Python dan pip
python3 --version
pip3 --version

# Buat virtual environment (best practice)
python3 -m venv /workspace/llm-env
source /workspace/llm-env/bin/activate
```

ⓘ 💡 **Tips:** Selalu gunakan `/workspace/` sebagai direktori kerja di RunPod. Data di luar direktori ini bisa hilang saat pod restart.

📦 Langkah 3: Instalasi OpenClaw & Download Model

Dengan virtual environment aktif, install OpenClaw dan download model Llama-3 dalam format GGUF dari HuggingFace. Format GGUF dioptimalkan untuk inferensi CPU/GPU yang efisien.

```
# Install OpenClaw (llama.cpp-compatible inference server)
pip install openclaw

# Verifikasi instalasi
openclaw --version

# Download model Llama-3 8B GGUF dari HuggingFace
# Gunakan huggingface-hub CLI untuk kemudahan
pip install huggingface_hub

python3 -c "
from huggingface_hub import hf_hub_download
hf_hub_download(
 repo_id='bartowski/Meta-Llama-3-8B-Instruct-GGUF',
 filename='Meta-Llama-3-8B-Instruct-Q4_K_M.gguf',
 local_dir='/workspace/models/'
)
print('
```

🚀 Langkah 4: Menjalankan OpenClaw Server

Jalankan OpenClaw agar mendengarkan di semua interface (`0.0.0.0`) pada port 8000. Flag `--host 0.0.0.0` sangat penting agar server dapat diakses dari luar pod melalui IP publik RunPod.

```
# Jalankan OpenClaw server
openclaw \
  --model /workspace/models/Meta-Llama-3-8B-Instruct-Q4_K_M.gguf \
  --host 0.0.0.0 \
  --port 8000 \
  --n-gpu-layers 35 \
  --ctx-size 4096 \
  --chat-format llama-3

# Output yang diharapkan:
# llama server listening at http://0.0.0.0:8000
# INFO:     Application startup complete.
```

✅ **Server aktif** jika muncul pesan `Application startup complete` tanpa error.

💡 Flag `--n-gpu-layers 35` memaksimalkan offloading ke VRAM RTX 3090 untuk inferensi yang lebih cepat.

🔌 Langkah 5: Testing Endpoint dari Laptop

Dapatkan **Public URL** pod kamu: buka dashboard RunPod → klik pod → bagian **"Connect"** → copy URL untuk port 8000. URL berbentuk `https://[POD-ID]-8000.proxy.runpod.net`. Kemudian test dari terminal laptop:

```
# Test sederhana dengan curl dari laptop lokal
curl -X POST https://[POD-ID]-8000.proxy.runpod.net/v1/chat/completions \
  -H "Content-Type: application/json" \
  -d '{
    "model": "llama-3",
    "messages": [
      {"role": "user", "content": "Jelaskan apa itu Large Language Model dalam 2 kalimat."}
    ],
    "max_tokens": 150,
    "temperature": 0.7
  }'

# Response yang diharapkan (format OpenAI-compatible):
# {"choices": [{"message": {"role": "assistant", "content": "..."}}]}
```

✅ 🎉 Jika mendapat JSON response dengan field `choices`, **endpoint kamu berhasil berjalan!** API ini kompatibel dengan OpenAI SDK.

☠️ Aturan Emas Lab: Housekeeping Wajib

🚨 **PERINGATAN KERAS:** "STOP IS NOT ENOUGH – YOU MUST DESTROY!" Pod yang di-Stop tetap menagih biaya storage. Kamu harus TERMINATE/DESTROY pod setelah lab selesai.

❌ Yang Salah: Hanya Stop Pod

Menekan tombol "Stop" hanya menghentikan komputasi, namun **storage tetap berjalan dan ditagih**. Dalam semalam, biaya bisa tetap menumpuk tanpa kamu sadari.

✅ Yang Benar: Terminate/Destroy Pod

Klik titik tiga (⋮) pada pod → pilih **"Terminate Pod"**. Konfirmasi dialog. Pod beserta semua data akan dihapus permanen dan tagihan berhenti total.

💾 Backup Sebelum Destroy

Sebelum terminate, salin semua file penting ke lokal. Gunakan `scp` atau download manual dari web terminal. Data yang hilang **tidak bisa dipulihkan**.

🏁 Selesai! Rangkuman & Tantangan Minggu Depan

✅ Yang Sudah Kita Capai

- Provision GPU instance di RunPod Community Cloud
- Setup environment Python dan install OpenClaw
- Download model Llama-3 8B GGUF dari HuggingFace
- Jalankan inference server di port 8000
- Hit endpoint API dari laptop sendiri

🚀 Tantangan Minggu Depan

1 Integrasi Python SDK

Buat script Python menggunakan `openai` library yang mengarah ke endpoint lokal kamu

2 Ganti Model

Coba model lain: **Mistral 7B** atau **Phi-3 Mini** — bandingkan kecepatan dan kualitasnya

3 Build Mini Chatbot

Buat simple chatbot CLI atau web (Flask/FastAPI) yang terhubung ke endpoint AI kamu

🎓 **Selamat!** Kamu kini memiliki infrastruktur AI pribadi yang bisa digunakan untuk eksperimen, fine-tuning, dan pengembangan aplikasi AI tanpa batasan rate limit atau biaya per-token.