Raspberry Pi 5 で Ollama v0.9.6 を使った LLM サーバー化ハンズオン資料

対象: Raspberry Pi 5（64bit OS, 推奨: 4GB 以上） **目的**: GitHub 配布の.tgz から Ollama v0.9.6 をインストール → サーバー起動 → 別ターミナルでモデル取得・実行までを体験

0. ゴール

.tgz を解凍し、ollama バイナリを移動して実行可能化できる
サーバーをバックグラウンド起動（ollama serve &）できる
別ターミナルから ollama pull でモデルを取得し、ollama run で対話開始できる

注意: 起動コマンドは ollama **serve** です（server ではありません）。

1. 事前確認

端末を開いて、次を確認します。

```
uname -m      # 例: aarch64 (= ARM64)

cat /etc/os-release | head -n 1  # 例: Debian GNU/Linux 12 (bookworm)
```

Raspberry Pi OS 64bit（ARM64）で進めます。

2. ダウンロード（GitHub から .tgz 取得）

ワーク用ディレクトリに移動して、v0.9.6 の ARM64 向けアーカイブを取得します。

```
mkdir -p ~/work/ollama && cd ~/work/ollama

wget https://github.com/ollama/ollama/releases/download/v0.9.6/ollama-linux-arm64.tgz# うまく落ちない場合は curl でも OK# curl -L -o ollama-linux-arm64.tgz ¥#
https://github.com/ollama/ollama/releases/download/v0.9.6/ollama-linux-arm64.tgz
```

（任意）チェックサム検証：

```
# 公式の sha256sum.txt を同ページから入手した場合# sha256sum -c sha256sum.txt | grep ollama-linux-arm64.tgz
```

ダウンロードがうまくいかず、別の PC がある場合 https://github.com/ollama/ollama/releases にアクセスし、1.04GB の ollama-linux-arm64.tgz をダウンロードし、USB メモリ等で移動してください。

3. .tgz の解凍とバイナリの移動・実行可能化

```
tar -xzf ollama-linux-arm64.tgz# 展開するとカレントに 'ollama' バイナリができます

ls -l ollama

# 実行権限を付与（念のため）

chmod +x ./ollama

# システム全体で使うなら /usr/local/bin へ移動（推奨）

sudo mv ./ollama /usr/local/bin/

# パスが通っていれば以下でバージョン表示

ollama --version
```

> 権限でエラーになる場合は sudo を付けて実行してください。

4. サーバーのバックグラウンド起動

```
# 11434 番ポートで API が上がります（デフォルト）

ollama serve# → バックグラウンドジョブ番号が返ってきます（例: [1] 12345）

# ログを見たいとき（例）

tail -f ~/.ollama/logs/server.log
```

> **止め方**: fg で前面化して Ctrl+C、または pkill -f "ollama serve"。再起動は再び ollama serve。

LAN + ローカルの同時アクセス（推奨）

0.0.0.0 で待ち受けると **LAN からもローカル（localhost）からも同時** にアクセスできます。

```
# （一時設定）全 IF で待ち受け → ローカルも OKexport OLLAMA_HOST=0.0.0.0:11434

ollama serve &
```

Pi の IP 確認：

```
hostname -I          # 例: 192.168.1.23# うまく出ない場合:

ip -4 addr show | grep -oP '(?<=inet¥s)¥d+(¥.¥d+){3}' | head -n1
```

動作確認：

```
# Pi 上（ローカル）

curl -s http://localhost:11434/api/tags | jq .# 別 PC（LAN）

curl -s http://<Pi の IP>:11434/api/tags | jq .
```

> **補足（ブラウザ連携する場合）**：Web UI などブラウザから叩くときは CORS 設定が必要になることがあります。OLLAMA_ORIGINS で許可するオリジンを指定してください（例：http://localhost:3000,http://<Pi の IP>:<port>）。簡易検証なら * も可ですが公開は非推奨。

恒久設定（systemd）

再起動後も有効にするには以下。

```
# systemd オーバーライド編集

sudo systemctl edit ollama.service
```

エディタが開いたら追記：

```
[Service]Environment="OLLAMA_HOST=0.0.0.0:11434"# （ブラウザ連携が必要なときのみ）#
Environment="OLLAMA_ORIGINS=http://localhost:*,http://127.0.0.1:*,http://<Pi のホスト
名>.local:*,http://<Pi の IP>:*"
```

反映：

```
sudo systemctl daemon-reload

sudo systemctl restart ollama

sudo systemctl status ollama --no-pager
```

最低限の防御（任意）

LAN 内限定にしたい場合はファイアウォールで制限します（UFW 例）。

```
sudo apt-get update && sudo apt-get install -y ufw

sudo ufw default deny incoming

sudo ufw allow from 192.168.0.0/16 to any port 11434 proto tcp

sudo ufw enable

sudo ufw status
```

> ルーターのポート開放は **しない** でください。WAN に直接晒すのは危険です。必要なら SSH トンネル等を使いましょう：
>
> ```
> ssh -N -L 11434:localhost:11434 pi@<Pi の IP>
> ```

5. 別ターミナルでモデルを取得（ollama pull）

新しいターミナルを開く → モデルを 1 つ取得します。Raspberry Pi 5（4GB）でも試しやすい超軽量モデルとして yuiseki/sarashina2.2:1b を例にします。

```
ollama pull yuiseki/sarashina2.2:1b
```

> 他にも軽量モデルがあります。メモリに余裕がなければ **1B 前後**の小さいモデルを選びましょう（例：gemma3:1b など）。

6. モデルの実行（ollama run）

```
ollama run yuiseki/sarashina2.2/1b# 初回はプロンプトが表示されたら試しに入力# 例）こんにちは。自己紹介してください。
```

終了は Ctrl+C（または /bye と入力）で。

7. API での疎通確認（おまけ）

サーバーが動いていれば、HTTP でも確認できます。

```
# ローカルに読み込まれているモデル一覧

curl -s http://localhost:11434/api/tags | jq .

# 一般出力（簡易テスト）

curl -s http://localhost:11434/api/generate ¥

  -H 'Content-Type: application/json' ¥

  -d '{"model":"yuiseki/sarashina2.2:1b","prompt":"Raspberry Pi で自己紹介して","stream":false}'

# よく使われてるやつ（Chat 形式）

curl -s http://localhost:11434/api/chat ¥

  -H 'Content-Type: application/json' ¥

  -d'{"model": "tinyllama","messages": [{"role": "user","content": "Raspberry Pi で自己紹介して"}],"stream": false}'
```

8. よくあるつまずき

- **ollama: command not found** → /usr/local/bin が PATH に入っているか確認。hash -r で再読込。
- **serve と server** → 正しくは ollama serve。server はコマンド名ではありません。
- **メモリ不足** → まずは 1B 前後の超軽量モデルを選ぶ。必要に応じて swap を増やす。
- **ポート競合** → 11434 を他サービスが使用していないか確認。環境変数 OLLAMA_HOST=127.0.0.1:11500 などで回避可。

9. 片付け（停止・削除）

```
# サーバー停止（ターミナルでそのまま起動した場合）

起動中のターミナルを選択して "Ctrl + C"

# サーバー停止（バックグラウンドの場合）

pkill -f "ollama serve"

# モデル削除（例）

ollama rm yuiseki/sarashina2.2:1b
```

10. 参考スニペット（配布用まとめ）

以下を丸ごと配布可能なチートシートとしてどうぞ：

```
# === Install ===cd ~/work/ollama

wget https://github.com/ollama/ollama/releases/download/v0.9.6/ollama-linux-arm64.tgz

 tar -xzf ollama-linux-arm64.tgz

 sudo mv ./ollama /usr/local/bin/

 ollama --version

# === Serve (BG) ===

ollama serve &

# === Pull & Run (2nd terminal) ===

ollama pull tinyllama

ollama run tinyllama
```