TWCC
運行EEMS

TWCC運行EEMS:

改換成使用TWCC虛擬運算，把ollama放在上面運行，當作server.

1. 要開虛擬運算需要先開虛擬網路(錢包管理員)
2. 注意公鑰(.pem)，需要下載備用。
3. 安全性群組先設port 22(for SSH)
4. VM裝完，跑ssh -i <path to .pem> ubuntu@<public IP>
5. 連進去先載ollama(ollama官網選linux版指令直接載)
6. 跑sudo systemctl edit --full ollama.service將0.0.0.0(or 0.0.0.0:11434) 開放全域監聽(不知道有無作用，似乎TWCC還是會擋即使有開11434port)
7. 設置防火牆(不知道有無作用)
8. curl 0.0.0.0:11434 -> Ollama is running代表成功

```
Description=Ollama Service
After=network-online.target

[Service]
ExecStart=/usr/local/bin/ollama serve
User=ollama
Group=ollama
Restart=always
RestartSec=3
Environment="PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin"
Environment="OLLAMA_HOST=0.0.0.0"

[Install]
WantedBy=default.target
```

```
sudo firewall-cmd --zone=public --add-port=11434/tcp --permanent

sudo firewall-cmd --reload

sudo firewall-cmd --zone=public --query-port=11434/tcp

firewall-cmd --list-ports

r run the application as superuser.
sudo firewall-cmd --list-ports
```

TWCC
運行EEMS

TWCC運行EEMS:

改換成使用TWCC虛擬運算，把ollama放在上面運行，當作server.

1. 檔案傳輸:scp -i <path to .pem> <local filename> ubuntu@< remote_server_ip >:<target path,ex:~>
2. 設置 SSH Tunneling，ssh -i <path to .pem> -L 8000:localhost:11434 <username>@<remote_server_ip> -N
3. Port 8000可以隨意設，但要在安全性群組開對應port
4. 永久設置 SSH Tunneling ，autossh -M 0 -f -N -i <path to .pem> -L 8000:localhost:11434 <username>@<remote_server_ip>(optional)
5. 連上後，在http://localhost:8000看到ollama is running就ok

TWCC運行EEMS

TWCC運行EEMS:

改換成使用TWCC虛擬運算，把ollama放在上面運行，當作server.

Request的部分(python):

```
from ollama import Client

client = Client(  host='http://localhost:8000',  headers={'x-some-header': 'some-value'})

def chat_with_ollama(model, message_content):
        response=client.chat(model=model, messages=[{'role': 'user', 'content': message_content}])
        return response
```

TWCC 運行EEMS

TWCC運行EEMS:

2. #不用另外再裝docker運行ollama，再裝完ollama後，另外開終端ollama serve即可。

3. ollama create <model name> -f <modelfile>，注意<modelfile>不用打副檔名。

參考:
https://github.com/ollama/ollama/blob/main/docs/faq.md

https://github.com/ollama/ollama/blob/main/docs/linux.md

https://github.com/ollama/ollama/issues/2132

https://blog.csdn.net/qq_32594047/article/details/137343545?utm_medium=distribute.pc_relevant.none-task-blog-2~default~baidujs_baidulandingword~default-1-137343545-blog-145631841.235^v43^pc_blog_bottom_relevance_base6&spm=1001.2101.3001.4242.2&utm_relevant_index=3

https://github.com/datawhalechina/handy-ollama/blob/main/docs/C4/2.%20%E5%9C%A8%20Python%20%E4%B8%AD%E4%BD%BF%E7%94%A8%20Ollama%20API.md

TWCC
運行EEMS

Ollama server多台local運行:

改換成使用TWCC虛擬運算，把ollama放在上面運行，當作server.

在local生成公私鑰對應:
Local(其他想加入伺服器的電腦):$ssh-keygen -t rsa -f“$env:USERPROFILE\.ssh\<隨意金鑰名>“

會回應: Enter passphrase (empty for no passphrase):---------這邊直接按Enter跳過

Get-Content “$env:USERPROFILE\.ssh\<剛才設定的金鑰名>” ---------可以查看私鑰(不建議公開)
Get-Content “$env:USERPROFILE\.ssh\ <剛才設定的金鑰名>.pub“----------可以查看公鑰(需要給有原私鑰.pem的主機)

Remote(server):$ nano ~/.ssh/authorized_keys
貼上(登錄)其他電腦的公鑰，Ctrl+S(儲存) ，Ctrl+X(退出)

Local(其他想加入伺服器的電腦):$ ssh -i ~/.ssh/ <剛才設定的金鑰名> -L 8000:localhost:11434 user@server_ip -N

TWCC運行EEMS

TWCC運行EEMS:

似乎rtx4060 的時脈比v100快，導致在跑8b LLM可能速度沒差多少，v100甚至略輸。

但v100可以跑8b以上的LLM(14b,32b)， rtx4060在跑14b基本龜速(甚至耗材)。

目前跑看看14b以上的效果如何(高參數量的LLM會處理較多的垃圾資訊。)