🚀 Deploying a Persistent FastAPI Service on UCloud

This guide walks you through deploying a **FastAPI** web application on **UCloud** using two persistent interactive jobs:

1. An **interactive compute job** running your FastAPI backend.
2. An **NGINX job** acting as a reverse proxy to expose your API securely over **HTTPS**.

🧩 System Architecture

Component	Suggested App	Recommended Machine Type
FastAPI Backend	Terminal / PyTorch / TensorFlow / JupyterLab / Coder	`uc1-l4-1` (1× NVIDIA L4 GPU)
NGINX Proxy	NGINX	`uc1-gc1-4` (4 vCPU, 16 GB RAM)

🛠️ Setup Overview

1. **Start** a persistent interactive compute job and run FastAPI on port **8000**.
2. **Launch** an NGINX job:
   - Use *Connect to other jobs* to link it to the FastAPI job. Choose an arbitrary *hostname*, e.g., `my-fastapi-app`.
   - Enable *Configure custom links to your applications* to generate and attach a public URL to your web app (e.g., `https://app-fastapi-demo.cloud.aau.dk`).

1️⃣ Prepare the FastAPI Project

Open a terminal interface in the compute job:

```
mkdir -p fastapi-demo/app
cd fastapi-demo
python -m venv .venv
source .venv/bin/activate
pip install fastapi uvicorn[standard]
```

Create `app/main.py`:

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel

app = FastAPI(title="Hello FastAPI")
```

```
class Item(BaseModel):
    name: str
    description: str = None
    price: float

@app.get("/ping")
async def ping():
    return {"status": "ok"}

@app.get("/items/{item_id}")
async def read_item(item_id: int):
    if item_id < 0:
        raise HTTPException(status_code=400, detail="Negative ID not
allowed")
    return {"item_id": item_id}

@app.post("/items/")
async def create_item(item: Item):
    return {"message": "Item created", "item": item}
```

2️⃣ Launch the FastAPI Service

Run your service using a persistent shell (e.g., `tmux` or `screen`):

```
#!/bin/bash

source /work/fastapi-demo/.venv/bin/activate
uvicorn app.main:app --host 0.0.0.0 --port 8000
```

3️⃣ Set Up the NGINX Reverse Proxy

In the NGINX job, open a terminal interface and edit `/etc/nginx/nginx.conf`:

```
worker_processes auto;
error_log /dev/stdout info;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    access_log /dev/stdout combined;

    log_format logger-json escape=json

'{"source":"nginx","time":$msec,"resp_body_size":$body_bytes_sent,'
```

```
'"host":"$http_host","address":"$remote_addr","request_length":$request_le
ngth,'

'"method":"$request_method","uri":"$request_uri","status":$status,'
        '"user_agent":"$http_user_agent","resp_time":$request_time,'
        '"upstream_addr":"$upstream_addr"}';

    server {
        listen 8080 so_keepalive=on;

        location / {
            proxy_pass http://my-fastapi-app:8000;
            proxy_set_header Host $host;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            proxy_set_header X-Forwarded-Proto $scheme;
        }
    }
}
```

Validate and reload the configuration:

```
nginx -t        # Check for syntax errors
nginx -s reload # Reload NGINX with the new config
```

NGINX will now forward HTTPS requests from your public URL to the internal FastAPI service.

4️⃣ Smoke Test Your API

1. Check if it's live:

```
curl -X GET 'https://app-fastapi-demo.cloud.aau.dk/ping' \
     -H 'accept: application/json'
# {"status": "ok"}
```

2. Create an item:

```
curl -X POST "https://app-fastapi-demo.cloud.aau.dk/items/" \
     -H "Content-Type: application/json" \
     -d '{"name": "Book", "description": "A mystery novel", "price":
12.99}'
# {"message":"Item created","item":{"name":"Book","description":"A mystery
novel","price":12.99}}
```

3. Swagger UI available at:

   👉 `https://app-fastapi-demo.cloud.aau.dk/docs`

✅ Recap

- **FastAPI Job**: Runs your backend on port `8000` (GPU optional).
- **NGINX Job**: Connects to the backend and publishes a secure public URL over HTTPS.

Customize your hostname and URL, and your FastAPI application will be publicly accessible.