**Deploying a Streamlit App with Docker on Azure VM — A Step-by-Step Guide**

Reference Blogs:

<https://medium.com/@harshitsundriyal/rapid-deployment-of-streamlit-app-in-cloud-virtual-machine-7193f629be58>

<https://medium.com/@srijaanaparthy/step-by-step-guide-creating-a-virtual-machine-in-azure-from-scratch-fbacccc57635>

GitHub repository for deployment: <https://github.com/Swapnil-Ransing/AnsweringQuestionsUsingLLM>

**Overview**

In this guide, we will:

1. Dockerize a local Python/Streamlit application.
2. Test it locally.
3. Create and configure an Azure Virtual Machine.
4. Deploy and run the Docker container on the VM.
5. Access the app from the browser.

This process is useful if you want a portable, reproducible, and cloud-hosted environment for your application.

**1. Prerequisites**

Before starting, ensure you have the following installed locally:

* [Python 3.10+](https://www.python.org/downloads/)
* Docker Desktop
* Azure CLI
* Git
* An Azure account

**2. Dockerizing the Streamlit App**

**2.1 Project Structure**

Example project layout: We mention .env in dockerignore file so that it does not get committed to image. Also we mention .env in gitignore so that it does not get committed to git repo.

AnsweringQuestionsUsingLLM/

│

├── docker\_streamlit\_app.py

├── requirements.txt

├── .env # Environment variables (Not committed to GitHub)

└── Dockerfile

Also note that since we are not passing .env file with docker image, environment variable code sample will look like as :

import os

OPENAI\_API\_KEY = os.getenv("OPENAI\_API\_KEY")

LANGSMITH\_API\_KEY = os.getenv("LANGSMITH\_API\_KEY")

LANGSMITH\_PROJECT = os.getenv("LANGSMITH\_PROJECT")

os.environ["LANGSMITH\_TRACING"] = "true"

**we have not used load\_env**

**2.2 Creating the Dockerfile**

Here’s an example Dockerfile:

dockerfile

# Use official Python base image

FROM python:3.10-slim

# Set working directory

WORKDIR /app

# Copy requirements first (for caching)

COPY requirements.txt .

# Install dependencies

RUN pip install --no-cache-dir -r requirements.txt

# Copy application files

COPY . .

# Expose the Streamlit port

EXPOSE 8501

# Run the Streamlit app

CMD ["streamlit", "run", "docker\_streamlit\_app.py", "--server.port=8501", "--server.address=0.0.0.0"]

**3. Testing the Docker Image Locally**

For building docker image locally, you need to install docker preferably docker desktop as it comes with UI. Check how to install docker and prerequisite for it before installing.

**3.1 Build the Image**

docker build -t epfo-app .

**3.2 Run the Container**

docker run -d --name epfo-container -p 8501:8501 --env-file .env epfo-app

We pass .env file which is present locally in the repo folder to the container.

**3.3 Test in Browser**

Visit:

http://localhost:8501

If the app works locally, you’re ready to move to Azure.

**4. Creating an Azure VM**

**4.1 Create VM from Azure Portal**

1. Go to **Azure Portal → Virtual Machines → Create**.
2. Select:
   * **Image:** Ubuntu 22.04 LTS
   * **Size:** At least Standard\_B2s for small apps (adjust if you need more CPU/RAM).
3. Enable **SSH authentication** and download the .pem key file.
4. Allow inbound port 22 (SSH).
5. Create the VM.

**For 1.2 GB of docker image memory, around 11 GB vm disk space is utilized.**

**4.2 Add Docker Port**

Later, we’ll also open **8501** for the Streamlit app.

**5. Connecting to the VM**

**5.1 SSH into the VM**

ssh -i "/c/Swapnil/GenerativeAI/epfo-web-app\_key.pem" azureuser@<VM\_PUBLIC\_IP>

**6. Installing Docker on the VM**

Run the following inside the VM:

bash

CopyEdit

# Update packages

sudo apt update && sudo apt upgrade -y

# Install Docker

sudo apt install docker.io -y

# Enable Docker service

sudo systemctl enable docker

sudo systemctl start docker

# Add current user to Docker group

sudo usermod -aG docker $USER

Then **log out and log in again** to apply group changes.

**7. Deploying the App on the VM**

**7.1 Clone Your Repo**

git clone https://github.com/Swapnil-Ransing/AnsweringQuestionsUsingLLM.git

cd AnsweringQuestionsUsingLLM

**7.2 Transfer the .env File**

From your local machine: Use git bash as cmd and power shell did not worked:

scp -i "/c/Swapnil/GenerativeAI/epfo-web-app\_key.pem" "/c/Swapnil/GenerativeAI/AnsweringQuestionsUsingLLM/.env" azureuser@40.81.225.102:/home/azureuser/epfo-app/AnsweringQuestionsUsingLLM/

**7.3 Build the Docker Image**

You can now start using cmd or powershell

docker build -t epfo-app .

**7.4 Run the Container**

docker run -d \

--name epfo-container \

-p 8501:8501 \

--env-file /home/azureuser/AnsweringQuestionsUsingLLM/.env \

epfo-app

**8. Opening Port 8501 in Azure**

az vm open-port --port 8501 --resource-group <your-resource-group> --name <your-vm-name>

Or via Portal:

* Go to **VM → Networking → Add inbound port rule** for port 8501.

**9. Testing the Deployment**

From your browser:

http://<VM\_PUBLIC\_IP>:8501

You should see your Streamlit app live.

**10. Stopping and Cleaning Up**

docker stop epfo-container

docker rm epfo-container

To avoid costs:

az vm delete --name <your-vm-name> --resource-group <your-resource-group>

**11. Key Notes**

* Keep .env out of GitHub (use .gitignore).
* Always restrict inbound ports to what’s necessary.
* Shut down/delete the VM when not in use to save costs.