FastAPI on AWS EC2 (t2.micro/t3.micro) with Docker & AWS SSM Parameter Store

Full Deployment Guide, Pitfalls, and Fixes

Project: AWS Deployment of `joke_api`
Aryan Singhal

November 11, 2025

Contents

1 Goal & Architecture

Deploy a public FastAPI app (`aloofzebra03/joke_api:latest`) on an Ubuntu `t2.micro` EC2 instance, using:

- Docker for the runtime,
- AWS Systems Manager Parameter Store for secrets (no secrets in the image),
- Supabase Postgres for LangGraph `PostgresSaver` (stateful checkpoints),
- Public HTTP on port 8000 (container listens on 8000, mapped from host:8000).

Key Choices

- Secrets namespaced under `/prod/fastapi/*` for clean IAM scoping and bulk fetch.
- EC2 IAM role allows read/`with-decryption` from Parameter Store (no access keys on box).
- Region: `ap-south-1`(Mumbai) (Preferrably be consistent everywhere).
- We assume that the docker image we are pulling is *aloofzebra03/joke-api:latest* and the name of the container made with docker on aws is *fastapi*.

2 Create Parameters in AWS SSM

Create secure parameters (once, from any machine with AWS CLI configured).This can be done on the aws console itself without any headaches of configuring ssh:

Listing 1: Create SSM parameters

```
aws ssm put-parameter --region ap-south-1 \
  --name "/prod/fastapi/POSTGRES_DATABASE_URL" \
  --type SecureString \
  --value "postgresql://postgres.<projectref>:<password>@aws-1-ap-south-1.pooler.supabase.
    com:5432/postgres?sslmode=require" \
  --overwrite

aws ssm put-parameter --region ap-south-1 \
  --name "/prod/fastapi/GOOGLE_API_KEY" \
  --type SecureString \
  --value "API-KEY" \
  --overwrite
```

Verify:

```
aws ssm get-parameters-by-path --region ap-south-1 \
  --path "/prod/fastapi/" --with-decryption
```

Note: AWS Console is NOT the same as EC2 terminal that we can access by connecting to it.

3 IAM Role & Policy for EC2

Create an IAM role with the minimal policy (name: `FastApiParamRead`) and attach it to an EC2 role when you launch the instance (name: `ec2-fastapi-ssm-role`). Policy JSON:

Listing 2: IAM policy for reading SSM and decrypting KMS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParametersByPath",
        "ssm:GetParameter",
        "ssm:GetParameters"
      ],
      "Resource": "arn:aws:ssm:ap-south-1:<YOUR_ACCOUNT_ID>:parameter/prod/fastapi/*"
    },
    {
      "Effect": "Allow",
      "Action": "kms:Decrypt",
      "Resource": "*"
    }
  ]
}
```

Why this policy? It scopes access to `/prod/fastapi/*` and allows decrypting SecureString values. Attach the role to the instance at launch.

When Create Policy Option not found While creating the role if you are not able to find the *Create Policy* button then you can assign a default policy like *AmazonSSMReadOnly-Acess.*After creating the role you can go to the permission policies editing page of that role and you will see an option to *Create Inline Policy* over there.

4 Security Group & EC2 Launch

Security Group: add inbound HTTP rule:

Type	**Protocol**	**Port**	**Source**
HTTP	CustomTCP	8000	0.0.0.0/0

Launch EC2

- AMI: Ubuntu 22.04/24.04 LTS, Instance: `t2.micro`
- Attach IAM Role: `ec2-fastapi-ssm-role`
- Auto-assign Public IPv4: **Enabled**

Connect using **EC2 Instance Connect** (browser shell as `ubuntu`).

5 Install Runtime Dependencies on EC2

Listing 3: Install Docker, AWS CLI v2, jq, psql

```
sudo apt-get update
sudo apt-get install -y docker.io
sudo systemctl enable --now docker
sudo usermod -aG docker $USER
newgrp docker

# AWS CLI v2 (official)
sudo apt install -y unzip curl jq
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip -q awscliv2.zip
sudo ./aws/install
aws --version
```

Notes

- `systemctl enable –now docker` starts Docker now and on reboot.
- `usermod -aG docker $USER` + `newgrp docker` lets you run Docker without sudo.

6 SSM → .env Script

Fetch all params from `/prod/fastapi/` and write `/etc/fastapi/.env` with `KEY=value`.

Listing 4: Create helper script

```
sudo tee /usr/local/bin/ssm-to-env.sh > /dev/null <<'EOF'
#!/usr/bin/env bash
set -euo pipefail
PARAM_PATH="/prod/fastapi/"
OUT_FILE="/etc/fastapi/.env"

aws ssm get-parameters-by-path \
  --region ap-south-1 \
  --path "$PARAM_PATH" \
  --with-decryption \
  --recursive \
  --query 'Parameters[].{Name:Name,Value:Value}' \
  --output json \
| jq -r '.[] | "\(.Name | split("/")[-1])=\(.Value)"' > "$OUT_FILE"

chmod 600 "$OUT_FILE"
EOF

sudo chmod 0755 /usr/local/bin/ssm-to-env.sh
```

Generate the file and verify:

```
sudo mkdir -p /etc/fastapi
sudo /usr/local/bin/ssm-to-env.sh
sudo cat /etc/fastapi/.env
# Expected:
# POSTGRES_DATABASE_URL=postgresql://...
# GOOGLE_API_KEY=AIzaSy...
sudo chmod 644 /etc/fastapi/.env
```

7 Run the Container

Listing 5: Pull and run

```
docker pull aloofzebra03/joke_api:latest

docker run -d --name fastapi \
  --restart unless-stopped \
  --env-file /etc/fastapi/.env \
  -p 8000:8000 \
  aloofzebra03/joke_api:latest

docker ps
docker logs -f fastapi
```

Port Mapping

- App listens on `0.0.0.0:8000` *inside* the container.
- Exposed as `EC2:8000` via `-p 8000:8000`.
- Access at `http://<EC2_PUBLIC_IP:port>/docs`.(Remember to add the *:<port>* **manually** otherwise it won't work)

8 Health Checks

```
# Inside EC2
curl http://localhost:8000/openapi.json
curl http://localhost:8000/docs | head -n 5
```

If logs show Uvicorn running but browser says “site can’t be reached”: ensure SG allows port 8000, instance has Public IPv4, and you’re using `http://` (not https).

9 Application Code Expectation (DB URL)

Your app reads:

Listing 6: Code snippet (DB URL env)

```
pool = ConnectionPool(
    conninfo=os.getenv('POSTGRES_DATABASE_URL'),  # expects this exact name
    max_size=20,
    kwargs={"autocommit": True, "prepare_threshold": 0},
)
```

Therefore, `.env` must export `POSTGRES_DATABASE_URL=...`. Using a different key name (e.g., `DATABASE_URL`) will result in `NoneType` errors.

10 Pitfalls Encountered & Fixes

1) `awscli` missing

Install AWS CLI v2 via official zip (Listing 3).

2) `/usr/local/bin/ssm-to-env.sh: command not found`

Likely CRLF line endings or not executable. Fix:

```
sudo sed -i 's/\r$//' /usr/local/bin/ssm-to-env.sh
sudo chmod 0755 /usr/local/bin/ssm-to-env.sh
sudo bash /usr/local/bin/ssm-to-env.sh
```

3) `ParameterNotFound`

Wrong name, region, or path:

- Verify names under `/prod/fastapi/`.
- Ensure region `ap-south-1` throughout.

4) `.env` contains path prefix

If you see:

```
/prod/fastapi/POSTGRES_DATABASE_URL=...
```

your JQ split is wrong. Use the script in Listing 4 (note `split("/")[-1]`).

5) `docker: open ... .env: permission denied`

Make readable:

```
sudo chmod 644 /etc/fastapi/.env
```

6) "Site can't be reached" externally

- SG must allow inbound TCP 8000 from `0.0.0.0/0`.
- Your url MUST be starting from http and contain the ec2 port number.
- Instance must have a Public IPv4 and be in a public subnet (default VPC works).
- Local check: `curl localhost:8000` should return JSON/HTML.

7) Psycopg pool errors / `NoneType.encode`

- Ensure `POSTGRES_DATABASE_URL` exists in `.env`.
- Include `?sslmode=require`.Otherwise the connection drops and you will get errors while interacting with the api.

11 Updating Secrets & Restarting

Listing 7: Rotate value, reload .env, restart container

```
aws ssm put-parameter --region ap-south-1 \
  --name "/prod/fastapi/GOOGLE_API_KEY" \
  --type SecureString --value "<newvalue>" --overwrite

sudo /usr/local/bin/ssm-to-env.sh
sudo chmod 644 /etc/fastapi/.env

docker stop fastapi && docker rm fastapi
docker run -d --name fastapi \
  --restart unless-stopped \
  --env-file /etc/fastapi/.env \
  -p 8000:8000 \
  aloofzebra03/joke_api:latest
```

12 Final Command Cheat Sheet

```
# -- Create parameters (once on AWS Console) --
aws ssm put-parameter --region ap-south-1 \
  --name "/prod/fastapi/POSTGRES_DATABASE_URL" \
  --type SecureString \
  --value "postgresql://postgres.<projectref>:<password>@aws-1-ap-south-1.pooler.supabase.
    com:5432/postgres?sslmode=require" \
  --overwrite
aws ssm put-parameter --region ap-south-1 \
  --name "/prod/fastapi/GOOGLE_API_KEY" \
  --type SecureString --value "AIzaSy..." --overwrite

# -- On EC2 (Ubuntu) --
sudo apt-get update
sudo apt-get install -y docker.io
sudo systemctl enable --now docker
sudo usermod -aG docker $USER
newgrp docker

sudo apt install -y unzip curl jq
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
unzip -q awscliv2.zip && sudo ./aws/install && aws --version
```

```
sudo tee /usr/local/bin/ssm-to-env.sh > /dev/null <<'EOF'
#!/usr/bin/env bash
set -euo pipefail
PARAM_PATH="/prod/fastapi/"
OUT_FILE="/etc/fastapi/.env"
aws ssm get-parameters-by-path --region ap-south-1 --path "$PARAM_PATH" \
  --with-decryption --recursive \
  --query 'Parameters[].{Name:Name,Value:Value}' --output json \
| jq -r '.[] | "\(.Name | split("/")[-1])=\(.Value)"' > "$OUT_FILE"
chmod 600 "$OUT_FILE"
EOF
sudo chmod 0755 /usr/local/bin/ssm-to-env.sh

sudo mkdir -p /etc/fastapi
sudo /usr/local/bin/ssm-to-env.sh
sudo chmod 644 /etc/fastapi/.env
sudo cat /etc/fastapi/.env

docker pull aloofzebra03/joke_api:latest
docker run -d --name fastapi --restart unless-stopped \
  --env-file /etc/fastapi/.env -p 8000:8000 \
  aloofzebra03/joke_api:latest

docker ps
docker logs -f fastapi
curl http://localhost:8000/openapi.json
```

13 FAQ

Which port is exposed? The app runs on container port 8000; host port 8000 is mapped to it via `-p 80:8000`. Security group must allow inbound 8000.

Can we use plain names instead of `/prod/fastapi/...`? Technically yes, but then you lose hierarchical fetch & neat IAM scoping. The script here assumes the namespaced path and strips the prefix.