

Boa! 🚀

Antes de subir na VPS, deixo um checklist rapidinho de “produção” pra evitar dor de cabeça:

1) Variáveis de ambiente (.env)

- `FLASK_ENV=production`
- `SECRET_KEY=<um valor bem aleatório>`
- `SQLALCHEMY_DATABASE_URI=mysql+pymysql://USER:PASS@HOST:3306/NOME_DB?charset=utf8mb4`
- Se usar login: `MAIL_*` (se houver), etc.

2) Gunicorn (tuning básico)

No `Dockerfile` já deixamos `-w 3`. Em VPS pequena:

- CPU 1–2 vCPUs → `-w 2` a `-w 3`, opcional `--threads 2`.

Ex.:

```
CMD ["python", "-m", "gunicorn", "-w", "3", "--threads", "2", "-b", "0.0.0.0:8000", "run:app"]
```

3) Reverse proxy (Nginx ou Caddy)

Exemplo Nginx (porta 80 → 8000 do container):

```
server {
    server_name sua.dominio.com;
    client_max_body_size 20m;

    location / {
        proxy_pass          http://127.0.0.1:8000;
        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;
    }
}
```

Depois, HTTPS com certbot:

```
sudo certbot --nginx -d sua.dominio.com
```

4) Banco (MySQL externo)

- Garanta que a VPS tem acesso à porta do MySQL (firewall/security group liberado).
- Usuário do MySQL com permissão apenas no schema do app.

- Backup agendado (mysqldump) é sempre bom.

5) Logs & saúde

- `docker compose logs -f web`
- `docker stats` para consumo
- Probes simples: `curl -I http://127.0.0.1:8000/health` (se quiser criar uma rota healthcheck).

6) Deploy/upgrade

```
git pull
docker compose build
docker compose up -d
docker image prune -f
```

7) Segurança rápida

- `ufw`/firewalld com 22, 80, 443 liberados.
- Atualizações de sistema (`apt upgrade`) periódicas.

Se pintar qualquer coisa na VPS (log, erro de proxy, timeouts), me manda que eu já te guio no ajuste fino.

Boa subida! ☺