# === CloudBooking deploy — Atlas + unified ports (5055 <- app, 8080 <- nginx) ===

Cd ~

MONGODB\_URI='mongodb+srv://taoc3860:J1g8PzxdZn62n9m2@tao.tzyxbdv.mongodb.net/cloudbooking?retryWrites=true&w=majority' \

JWT\_SECRET='cloudbooking\_secret\_key\_2025' \

TAG='v0.9.0' \

HTTP\_PORT=8080 \

bash -s <<'SCRIPT'

set -euo pipefail

: "${MONGODB\_URI:?MONGODB\_URI required}"

JWT\_SECRET="${JWT\_SECRET:-cloudbooking\_secret\_key\_2025}"

TAG="${TAG:-v0.9.0}"

HTTP\_PORT="${HTTP\_PORT:-8080}"

REPO\_URL="https://github.com/taoc3860-pixel/cloudbooking.git"

APP\_DIR="/opt/app/cloudbooking"

APP\_PORT="5055"

ME="$(whoami)"

echo "== Using TAG=$TAG HTTP\_PORT=$HTTP\_PORT APP\_DIR=$APP\_DIR =="

# 0) Ensure port 80 is not occupied by system nginx

sudo systemctl stop nginx >/dev/null 2>&1 || true

sudo systemctl disable nginx >/dev/null 2>&1 || true

# 1) Fresh clone and hand over ownership

sudo rm -rf "$APP\_DIR"

sudo mkdir -p "$(dirname "$APP\_DIR")"

sudo git clone --depth=1 "$REPO\_URL" "$APP\_DIR" >/dev/null

sudo chown -R "$ME":"$ME" "$APP\_DIR"

cd "$APP\_DIR"

git fetch --tags >/dev/null 2>&1 || true

if git rev-parse -q --verify "refs/tags/$TAG" >/dev/null; then

git checkout -q "$TAG"

else

echo "[info] tag $TAG not found, using default branch"

fi

# 2) .env — force PORT=5055 (unified)

cat > .env <<EOF

NODE\_ENV=production

PORT=${APP\_PORT}

JWT\_SECRET=${JWT\_SECRET}

MONGODB\_URI=${MONGODB\_URI}

EOF

# 3) Make sure server listens on 0.0.0.0:PORT (patch if needed)

if [ -f server.js ] && grep -q "app.listen" server.js; then

sed -i -E 's@app\.listen\([^)]\*\);@app.listen(process.env.PORT || '"$APP\_PORT"', "0.0.0.0", () => console.log(`API on :\${process.env.PORT||'"$APP\_PORT"'}`));@' server.js || true

fi

# 4) nginx — proxy to app:5055

cat > nginx.conf <<'NGINX'

events {}

http {

server {

listen 80;

server\_name \_;

root /usr/share/nginx/html;

index index.html;

location /api/ {

proxy\_pass http://app:5055/;

proxy\_http\_version 1.1;

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 Upgrade $http\_upgrade;

proxy\_set\_header Connection "upgrade";

}

}

}

NGINX

# 5) docker-compose — service names: app + nginx; expose 8080:80

cat > docker-compose.yml <<YML

services:

app:

build:

context: .

dockerfile: Dockerfile

image: cloudbooking:${TAG#v}

container\_name: cloudbooking-app-1

restart: unless-stopped

env\_file: .env

# 不对外暴露 5055（由 nginx 内部访问即可）

nginx:

image: nginx:alpine

container\_name: cloudbooking-nginx-1

depends\_on:

app:

condition: service\_started

restart: unless-stopped

volumes:

- ./web:/usr/share/nginx/html:ro

- ./nginx.conf:/etc/nginx/nginx.conf:ro

ports:

- "${HTTP\_PORT:-8080}:80"

YML

# 6) Build & up

docker compose down >/dev/null 2>&1 || true

docker compose build --no-cache

docker compose up -d

# 7) Quick checks

echo ""

echo "== Deployed. Try: http://<VM-IP>:${HTTP\_PORT}/ =="

docker compose ps

echo ""

echo "Log tail (app):"

docker compose logs --tail=80 app || true

echo ""

echo "Health:"

curl -sS -i http://localhost:${HTTP\_PORT}/ | head -n 1 || true

curl -sS -i http://localhost:${HTTP\_PORT}/api/health | head -n 1 || true

SCRIPT