# === 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) Stop system nginx to free port 80

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

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

# 1) Clone repo fresh

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

# 1.1) Handle lowercase dockerfile

if [ ! -f Dockerfile ] && [ -f dockerfile ]; then

ln -s dockerfile Dockerfile

fi

# 2) Write .env (always refreshed)

cat > .env <<EOF

NODE\_ENV=production

PORT=${APP\_PORT}

JWT\_SECRET=${JWT\_SECRET}

MONGODB\_URI=${MONGODB\_URI}

EOF

# 3) Only create nginx.conf if missing

if [ ! -f nginx.conf ]; then

echo "[info] nginx.conf not found, creating default one..."

cat > nginx.conf <<'NGINX'

events {}

http {

include /etc/nginx/mime.types;

default\_type application/octet-stream;

server {

listen 80;

server\_name \_;

root /usr/share/nginx/html;

index index.html;

# MIME fallback for minimal images

types {

text/css css;

application/javascript js mjs;

image/svg+xml svg;

font/woff2 woff2;

font/woff woff;

}

# proxy API to app:5055, keep /api prefix

location /api/ {

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 "";

proxy\_pass http://app:5055;

proxy\_redirect off;

}

location = /healthz { proxy\_pass http://app:5055/healthz; }

location = /api/healthz { proxy\_pass http://app:5055/api/healthz; }

}

}

NGINX

else

echo "[info] nginx.conf already exists, keeping your version."

fi

# 4) docker-compose.yml (recreate every time to ensure services ok)

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

healthcheck:

test: ["CMD-SHELL", "wget -qO- http://localhost:${APP\_PORT}/healthz || curl -fsS http://localhost:${APP\_PORT}/healthz"]

interval: 10s

timeout: 3s

retries: 10

nginx:

image: nginx:alpine

container\_name: cloudbooking-nginx-1

depends\_on:

app:

condition: service\_healthy

restart: unless-stopped

volumes:

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

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

ports:

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

YML

# 5) Rebuild and up

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

docker compose build

docker compose up -d

# 6) Quick checks

echo ""

docker compose ps

echo ""

echo "== Health checks =="

curl -sS -I "http://localhost:${HTTP\_PORT}/api/healthz" | head -n 1 || true

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

echo ""

echo "== CSS Content-Type check (should be text/css) =="

curl -sS -I "http://localhost:${HTTP\_PORT}/css/style.css" | grep -i '^Content-Type' || true

echo ""

SCRIPT

Roll update

export IMAGE\_REPO="ghcr.io/taoc3860-pixel/cloudbooking"

export APP\_VERSION="stack-v0.1.0"

docker stack deploy -c cloudbooking-stack.yml cloudbooking --with-registry-auth

watch -n 1 "docker service ps cloudbooking\_app --no-trunc"

roll back

export IMAGE\_REPO="ghcr.io/taoc3860-pixel/cloudbooking"

export APP\_VERSION="0.1.0"

docker stack deploy -c cloudbooking-stack.yml cloudbooking --with-registry-auth

watch -n 1 "docker service ps cloudbooking\_app --no-trunc"

#page link

[Dashboard · Room Booker](http://34.56.13.247:8080/dashboard.html)

#git

git status

git add -A

git commit -m "feat: unify to PORT=5055, /api prefix, nginx full reverse proxy; stable Atlas deploy"

git push origin main

git tag -fa v0.9.0 -m "CloudBooking 0.9.0 (stable: PORT=5055, /api, nginx proxy)"

git push origin v0.9.0 –force

docker login ghcr.io -u taoc3860-pixel -p ghp\_3XtCfrJk75xei0bu0lU1VSc5KfwBfC2YtEIp