# === Clone v0.7.0 and start with Docker Compose ===

set -e

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

GIT\_TAG="v0.6.0"

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

# 1) Free port 80 (optional) and get code @ tag

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

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

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

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

cd "$APP\_DIR"

git fetch --tags -q

git checkout -f "tags/${GIT\_TAG}" -q

# 2) Minimal .env (keep if already exists)

[ -f .env ] || printf 'JWT\_SECRET=mySuperSecretKey\nPORT=5055\nNODE\_ENV=production\nMONGODB\_URI=\n' | sudo tee .env >/dev/null

# 3) Start (auto-detect docker compose)

if docker compose version >/dev/null 2>&1; then

docker compose up -d --build

else

docker-compose up -d --build

fi

# 4) Show status

if docker compose version >/dev/null 2>&1; then

docker compose ps

else

docker-compose ps

fi

Token

ghp\_bbbltm0CueFLMr66odzwhSTtfpEwTD2t222t

echo ghp\_bbbltm0CueFLMr66odzwhSTtfpEwTD2t222t | docker login ghcr.io -u taoc3860-pixel --password-stdin

# ===== CloudBooking v0.7.0: fault-tolerant compose up (paste & run) =====

# ===== CloudBooking v0.8.0 — Minimal Redeploy (never exit) =====

set +e

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

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

TAG="v0.8.0"

ok(){ echo "✅ $\*"; }

warn(){ echo "⚠️ $\*"; }

err(){ echo "❌ $\*"; }

# 0) 准备目录 & 停系统 nginx（避免占80）

sudo mkdir -p /opt/app && sudo chown -R $(id -u):$(id -g) /opt/app

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

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

# 1) 取代码（git 优先，失败用流式 tarball）

rm -rf "$APP\_DIR" 2>/dev/null

if git clone --branch "$TAG" --single-branch --depth=1 "$REPO\_URL" "$APP\_DIR" 2>/tmp/gb.err; then

ok "git clone $TAG OK"

else

warn "git 失败，改用 tarball"

cd /opt/app || exit 0

curl -fsSL "https://github.com/taoc3860-pixel/cloudbooking/archive/refs/tags/${TAG}.tar.gz" | tar -xz

mv /opt/app/cloudbooking-${TAG#v}\* "$APP\_DIR" 2>/dev/null || err "tarball 展开有问题"

fi

cd "$APP\_DIR" 2>/dev/null || { err "进入 $APP\_DIR 失败"; exit 0; }

# 2) 生成 .env（若不存在）

[ -f .env ] || cat > .env <<'ENV'

JWT\_SECRET=change\_this\_to\_a\_long\_random\_secret

PORT=5055

NODE\_ENV=production

# 本地容器内 Mongo（入门验证用）；改 Atlas 时把这一行替换为你的 SRV/URI

MONGODB\_URI=mongodb://mongo:27017/cloudbooking

ENV

# 5) 构建 & 启动（不中断）

echo "== build =="

docker compose build || warn "compose build 有报错"

echo "== up -d =="

docker compose up -d || warn "compose up 有报错"

# 6) 状态与关键日志

echo "== ps =="

docker compose ps || true

echo "== tail logs (app/nginx) =="

docker compose logs --tail=80 app 2>/dev/null || true

docker compose logs --tail=80 nginx 2>/dev/null || true

IP=$(curl -s ifconfig.me 2>/dev/null || echo "<YOUR\_IP>")

echo

ok "访问：http://$IP:8080/"

echo "若 UI 能开但不入库，请把 .env 的 MONGODB\_URI 改为 Atlas SRV/URI 后：docker compose up -d --build"

echo "实时日志：docker compose logs -f app"