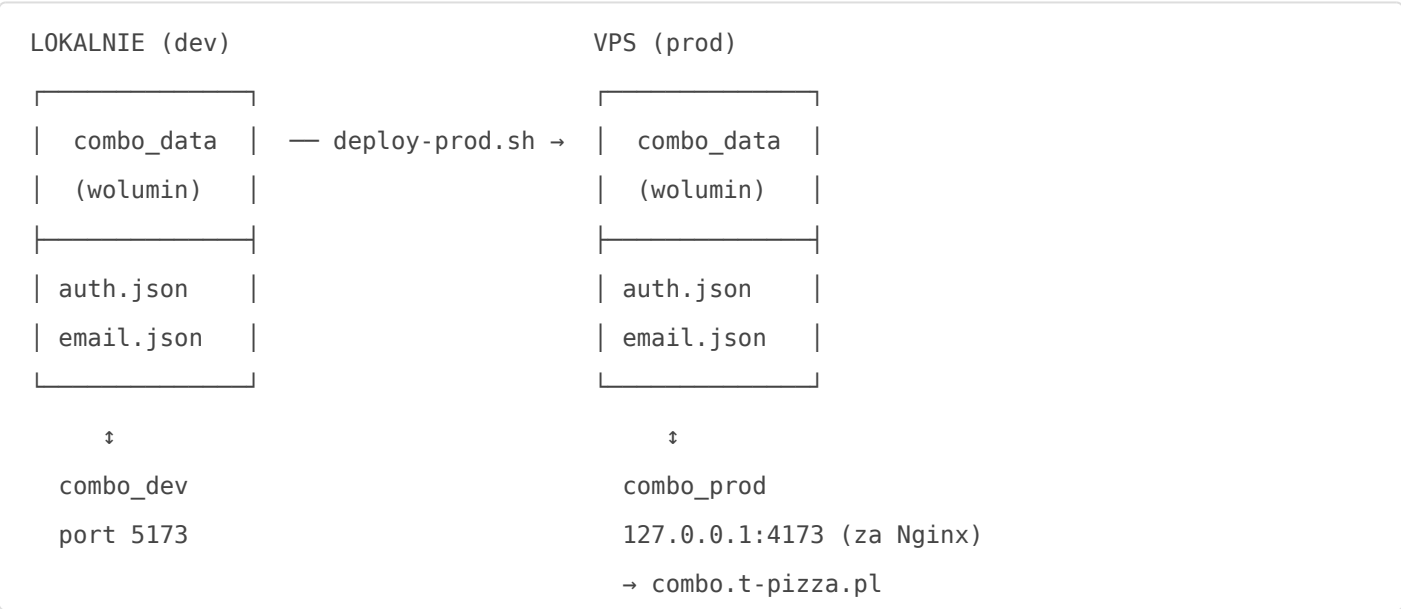


Deploy na produkcj?

Deploy na produkcj?

Skrypt `scripts/deploy-prod.sh` automatyzuje przenoszenie aplikacji + danych z lokalnego środowiska dev na serwer produkcyjny (home.pl ComboVPS → `combo.t-pizza.pl`).

Architektura danych



Plik	Zawartość
<code>auth.json</code>	Użytkownicy, hasła (scrypt), JWT secret, logi audytowe
<code>email.json</code>	SMTP config, harmonogramy raportów, szablony email, lista odbiorców

Jednorazowy setup

1. Skonfiguruj po??czenie SSH

Klucz SSH do VPS (`root@212.132.103.157`). Sugerowany wpis w `~/.ssh/config`:

```
Host vps
  HostName 212.132.103.157
  User root
  IdentityFile ~/.ssh/jprdl
```

2. Utwórz plik `.deploy.env`

```
cp .deploy.env.example .deploy.env
```

Domyślna konfiguracja (`host=vps`, `klucz=~/.ssh/jprdl`, `katalog=/opt/docker/combo-raport`, `profil=prod`) działa bez zmian, jeśli używasz wpisu `vps` z `~/.ssh/config`.

```
DEPLOY_HOST=vps
DEPLOY_SSH_PORT=22
DEPLOY_USER=root
DEPLOY_SSH_KEY=~/.ssh/jprdl
DEPLOY_APP_DIR=/opt/docker/combo-raport
DEPLOY_PROFILE=prod
DEPLOY_CONTAINER=combo_prod
```

“`.deploy.env` jest w `.gitignore` — nigdy nie trafi do repozytorium.

3. Przygotuj VPS

Na serwerze (jednorazowo):

```
# Sieć Docker `web` (external) – utworzona w trakcie VPS init
docker network ls | grep web

# Klon repo
mkdir -p /opt/docker
git clone git@github.com:matzxp84/combo-raport-app.git /opt/docker/combo-raport

# Vhost Nginx → docker/nginx-vhost.conf
cp /opt/docker/combo-raport/docker/nginx-vhost.conf \
  /etc/nginx/sites-available/combo.t-pizza.pl.conf
```

```
ln -s /etc/nginx/sites-available/combo.t-pizza.pl.conf /etc/nginx/sites-enabled/  
nginx -t && systemctl reload nginx  
  
# SSL (po wskazaniu DNS A → 212.132.103.157)  
certbot --nginx -d combo.t-pizza.pl
```

U?ycie

Pe?ny deploy (tylko kod, domy?Inie)

```
./scripts/deploy-prod.sh
```

Co robi:

1. Kopiuje `.env` (secrets GoPos) na serwer przez SCP
2. Na VPS: `git pull --ff-only` + `docker compose --profile prod up -d --build`
3. Sprawdza healthcheck kontenera `combo_prod`

Kod + synchronizacja danych

```
./scripts/deploy-prod.sh --with-data
```

Dodatkowo:

1. Eksportuje `auth.json` i `email.json` z lokalnego wolumenu `combo_data`
2. Kopiuje je na serwer przez SCP
3. Importuje do wolumenu `combo_data` na VPS

Tylko dane

```
./scripts/deploy-prod.sh --data-only
```

Przydatne gdy dodałeś użytkowników w panelu admina na dev i chcesz ich przenieść na prod.

One-shot: commit + push + deploy

```
./scripts/publish.sh "commit message"  
# lub
```

Co jest przenoszone

Element	Jak przenoszone	Gdzie ląduje
Kod źródłowy	<code>git pull --ff-only</code>	<code>/opt/docker/combo-raport/</code>
<code>.env</code> (GoPos secrets)	SCP	<code>/opt/docker/combo-raport/.env</code>
<code>auth.json</code> (użytkownicy)	Docker volume export/import	wolumin <code>combo_data</code>
<code>email.json</code> (SMTP, szablony)	Docker volume export/import	wolumin <code>combo_data</code>

Co NIE jest przenoszone

- `node_modules` — instalowane na serwerze podczas `docker build`
- `dist/` — budowane na serwerze podczas `docker build`
- Logi Docker — per kontener, nie migrowane

Secrets — bezpieczeństwo

Secret	Plik	Ochrona
<code>GOPOS_CLIENT_ID /</code> <code>GOPOS_CLIENT_SECRET</code>	<code>.env</code>	<code>.gitignore</code> + <code>.dockerignore</code>
JWT secret	<code>auth.json</code> (w wolumenie)	nie w repo, w Docker volume
Hasła użytkowników	<code>auth.json</code> (script hash)	hashowane, nie w repo
Hasło SMTP	<code>email.json</code> (w wolumenie)	nie w repo, w Docker volume

Zasady:

- `.env` i `.deploy.env` NIGDY nie trafiają do gita
- Dane runtime (`auth.json`, `email.json`) żyją w Docker volumes, nie w repo
- Backup danych: `scripts/backup-gdrive.sh` (patrz [Backup](#))
- Przy pierwszym uruchomieniu bez danych serwer tworzy seed users — **zmień hasła natychmiast**

Seed users (domyślne)

Email	Hasło	Rola
matfl@tuta.com	pułtusk	admin
daniel.piekarski@t-pizza.pl	daniel	user

“ Tworzone TYLKO gdy `auth.json` jest pusty (brak użytkowników). Jeśli deployujesz z danymi — seed nie występuje.

Troubleshooting

Nie mog? po??czy? si? z serwerem

```
# Test ręczny
ssh vps "echo ok"
# lub bezpośrednio
ssh -i ~/.ssh/jprdl root@212.132.103.157 "echo ok"
```

Sprawdź: klucz SSH, wpis w `~/.ssh/config`, czy IP nie jest zbanowane w fail2ban (`fail2ban-client status sshd`).

Kontener nie startuje

```
ssh vps "docker logs combo_prod --tail 50"
```

Brak miejsca na dysku

Przed deployem wyczyść dysk na serwerze:

```
ssh vps "docker system prune -a && apt clean"
```

Dane nie przenios?y si?

Sprawdź zawartość wolumenu na serwerze:

```
ssh vps "docker run --rm -v combo_data:/data alpine cat /data/auth.json"
```

Nginx / SSL

```
ssh vps "nginx -t && systemctl reload nginx"  
ssh vps "certbot certificates"
```

Revision #4

Created 2026-05-28 04:01:01 CEST by matzxp84

Updated 2026-05-28 04:34:05 CEST by matzxp84