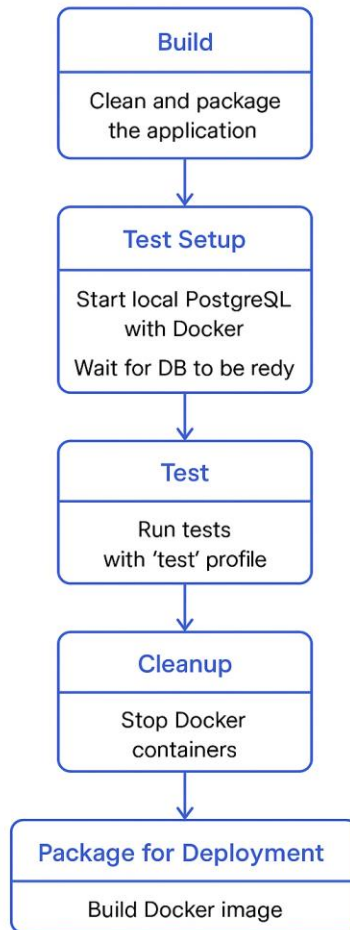


## SPRINT 3

### Build Pipeline



Die lokale Build-Pipeline mit einem Bash-Skript erstellt.

### Warum wir eine Build-Pipeline verwenden:

- Konsistenz: Jeder baut und testet den Code auf die gleiche Weise.
- Automatisierung
- Frühzeitiges Feedback: Bugs schneller finden.

## Erklärung der einzelnen Jobs in deiner Pipeline

- **Clean and Build (Kompilieren & Packen des Codes)**  
**Befehl:** `mvn clean package -DskipTests`  
Löscht alte Artefakte und verpackt die Anwendung in eine .jar-Datei. Tests werden zur Beschleunigung übersprungen.
- **Dokumentation generieren (JavaDoc)**  
**Befehl:** `mvn javadoc:javadoc`  
Generiert HTML-Dokumentation aus JavaDoc-Kommentaren.
- **Lokale Datenbank starten (PostgreSQL via Docker)**  
**Befehl:** `docker-compose up -d postgres_local`  
Startet einen lokalen PostgreSQL-Container zum Testen.  
**Befehl:** `sleep 10`  
Pausiert, um der Datenbank Zeit zum vollständigen Starten zu geben.
- **Tests ausführen (mit Test-Profil)**  
**Befehl:** `mvn test -Dspring.profiles.active=test`  
Führt automatisierte Tests mit der Testkonfiguration aus.
- **Docker-Container stoppen**  
**Befehl:** `docker-compose down`  
Fährt Docker herunter und räumt die Umgebung auf.
- **Docker-Image erstellen**  
**Befehl:** `docker build -t myapp:latest .`  
Verpackt die Anwendung in ein Docker-Image zur Bereitstellung.

## Produkt backlog

Backlog item	Einheit(Story Points)
<b>Als Haustierbesitzer: in möchte ich ein Profil erstellen können</b> , damit ich Infos über mein Tier und mich angeben kann.	<b>5</b>
<b>Als Betreuer: in möchte ich mich registrieren und ein Profil anlegen</b> , um meine Services wie Gassi gehen oder Urlaubsbetreuung anzubieten.	<b>5</b>
<b>Suchfunktion für Tierbesitzer: innen</b> , um passende Betreuer:innen zu finden	<b>3</b>
<b>Filteroptionen für Betreuungsangebote</b> , z. B. nur Notfallbetreuung oder nur für Hunde oder Katzen.	<b>3</b>
<b>Bewertungsfunktion</b> , damit Tierbesitzer:innen Feedback zu Betreuer:innen geben können	<b>3</b>
<b>Kalenderintegration für Terminbuchungen</b> , sodass Termine direkt vereinbart werden können	<b>15</b>
<b>Schnellzugriff auf Notfallkontakte oder Notfallbetreuer:innen</b>	<b>3</b>
<b>Chat- oder Nachrichtensystem</b> , um zwischen Besitzer:innen und Betreuer:innen kommunizieren zu können	<b>20+</b>
<b>Profilbilder für Nutzer:innen</b> , damit man weiß, mit wem man es zu tun hat	<b>8</b>
<b>lokale Build-Pipeline</b> erstellen, um Builds lokal zu testen.	<b>8</b>
<b>Als Nutzer</b> möchte ich mich sicher und vollständig ausloggen können, damit meine Sitzung geschützt ist.	<b>3</b>
<b>Mobile Optimierung</b> , also dass die App/Seite auch auf dem Handy gut funktioniert	<b>5</b>

Die werte(Fibonacci - Skalar) für storypoints beschreiben den relativen Aufwand im Vergleich zu anderen Aufgaben. also von 1(Sehr einfache Aufgabe, kaum Aufwand oder Risiko) bis 20+(Sehr groß oder unklar – sollte evtl. in kleinere Aufgaben aufgeteilt werden.)