

Ticketpilot einrichten – Schritt für Schritt

Von der Installation aus dem Marktplatz mit der kostenlosen 30-Tage-Lizenz bis zum vollständig verbundenen System: Ticketsystem, KI-Backend, Engine, GitHub und Gateway.

ENGINE

localhost:56100

AUTHORITY

localhost:56120

LOWCODE

localhost:56130

FLOW-EDITOR

localhost:1880

1 App auswählen und Trial starten

WO	ProcessCube Studio (CUBY) → Startseite „Meine Apps“ → Bereich „Kostenlos starten“
WAS	Karte „Ticketpilot Lokal“ (lokal im Container – volle DSGVO-/NIS2-Kontrolle) wählen
WOMIT	Button „Kostenlos starten“ → die 30-Tage-Testlizenz wird automatisch erstellt

2 Instanznamen vergeben

Im Dialog **„Instanzname vergeben“** einen Namen eintragen (z. B. `Ticketpilot`) und auf **„Installieren“** klicken.

3 Komponenten installieren lassen

Im **Marketplace** werden automatisch installiert: **ProcessCube Docker (Ticketpilot)**, **Ticketpilot Lokal** und das **Ticketpilot Add-On**.

Warten, bis alle Komponenten den Status **„Installiert“** anzeigen.

Nach der Installation öffnet sich der Konfigurations-Assistent automatisch.

4 ProcessCube-Instanz auswählen (Schritt 1)

Die Instanz wählen, auf der Ticketpilot laufen soll. Angezeigt werden die Endpunkte Engine :56100, LowCode :56130 und Authority :56120.

Bestätigung im Assistenten: „✓ Instanz ausgewählt“.

5 Ticketing-System wählen

Auswahl zwischen **Odoo** (Helpdesk-Modul) und **Jira** (Atlassian). In der Aufnahme → **Odoo**.

6 Odoo-Account & API einrichten (Schritt 2 · Pflicht)

ACCOUNT Kostenlosen Account unter **odoo.com** registrieren

API-KEY In Odoo: **Profilbild (oben rechts) → „Meine Präferenzen“ → Tab „API Keys“ → „Neuen API-Key erstellen“**

EINTRAGEN **Odoo Base URL · Odoo API Key · Odoo User ID** in die Felder des Assistenten

⚠ Der API-Key wird in Odoo **nur einmalig** angezeigt – sofort kopieren und sicher aufbewahren. Erscheint „**Odoo Server Error**“, die drei Werte (URL, Key, User ID) prüfen.

Anschließend auf „**Weiter**“.

Ort für alle folgenden Befehle: App-Ansicht „Ticketpilot“ → Abschnitt „Terminal“
(Terminal im Docker-Container).

7 Setup starten & KI-Backend wählen

```
Terminal – Ticketpilot

$ pc-runtime setup

Welches AI-Backend soll genutzt werden?
  1) Claude
  2) Codex
  3) Hybrid (beide)
Auswahl [1/2/3]: 1
Gewähltes Backend: claude
```

Claude-Subscription anmelden (nutzt deine Subscription, **kein API-Key**): Es öffnet sich der Claude-Login im Browser → **Organisation auswählen** → angezeigten **Code kopieren** und im Terminal **einfügen**.

```
Terminal – Abschluss

Default model set to anthropic/claude-opus-4-8
✓ Claude (Subscription) eingerichtet.
– Abschlussprüfung (inkl. Smoke-Test gegen Agent 'main') –
```

Hinweis aus dem Setup: eine offene **Pairing-Anfrage** (`openclaw devices approve <id>`) wird für Teil 4 relevant.

8 An der Engine anmelden

Terminal

```
$ pc login http://host.docker.internal:56100
```

Open this URL in your browser to log in:

<http://authority:56120/device>

Enter code: **WRTM-PWXD**

You are now logged in.

Browser auf <http://authority:56120/device> öffnen und den angezeigten Code eingeben (oder den vorausgefüllten Link nutzen).

9 GitHub verbinden

Terminal

```
$ gh auth login
```

```
? Where do you use GitHub? GitHub.com
```

```
? Authenticate GitHub CLI? Login with a web browser
```

```
! First copy your one-time code: 44B1-8BBA
```

```
→ https://github.com/login/device
```

```
✓ Congratulations, you're all set!
```

Einmal-Code kopieren und auf github.com/login/device eingeben.

TEIL 4

Gateway-Pairing im LowCode-Flow

10 Gateway-Node prüfen & Pairing bestätigen

WO Quick-Link „**LowCode Ticketpilot**“ (`localhost:1880`) → Flow (z. B. „Ticket umsetzen Odoo“) → Node **openclaw-gateway**

PRÜFEN Gateway URL `ws://processcube-agents:18789` · Token `OPENCLAW_GATEWAY_TOKEN`

Bei der Meldung „**Pairing erforderlich**“ den angezeigten Befehl auf dem Gateway-Host ausführen:

```
Gateway-Host  
  
$ openclaw devices approve <id>
```

Danach im Node auf „**Verbindung testen**“ klicken.

TEIL 5

Verifizieren, dass alles läuft

11 Prozesse deployed

Quick-Link „**LowCode Ticketpilot**“ → **Engine** → **Process Models** zeigt: *Support Ticket verarbeiten, Support Ticket verteilen, Ticket releasen, Ticket umsetzen.*

12 Helpdesk & End-to-End-Test

Quick-Link „**Odoo Kundendienst**“ → Kanban (*Work in progress · In Review · Awaiting approval · Done · Cancelled*) mit einlaufenden Tickets.

Ein Odoo-Ticket erzeugt über den Coding-Agent automatisch einen **GitHub Pull Request** (Label `github-coding-agent`); die **Phasen wechseln automatisch** im Ticket.



Kurz-Checkliste

SCHRITT	WO	WOMIT
Trial starten	CUBY → Meine Apps	„Ticketpilot Lokal“ → Kostenlos starten
Instanz benennen	Installations-Dialog	Name + „Installieren“
Ticketsystem	Assistent Schritt 1–2	Odoo Base URL + API Key + User ID
KI-Backend	App-Terminal	<code>pc-runtime setup</code> → Claude-Login
Engine-Login	App-Terminal	<code>pc login ...:56100</code> + Device-Code
GitHub	App-Terminal	<code>gh auth login</code> + Device-Code
Gateway-Pairing	LowCode (:1880)	<code>openclaw devices approve <id></code>
Check	LowCode + Odoo Kundendienst	Process Models & Kanban