

Ontaric GmbH

Resourcecloud im eigenen Rechenzentrum

Installation, erste Schritte nach der Installation und Updates für Kunden, die Resourcecloud selbst betreiben.

Gilt für Version 5.3.0 · Stand 10.10.2026 · Entwurf

1. Überblick
2. Voraussetzungen
3. Installation (Docker Compose hinter eigenem Proxy · einzelne VM mit automatischem HTTPS · Kubernetes)
4. Ersteinrichtung
5. Erste Schritte: Dateispeicher (S3) · Mail · Lizenz · KI · Microsoft 365 · Single Sign-On · Organisation
6. Betrieb und Sicherung
7. Update und Rückfall
8. Fehlersuche
9. Anhang: Variablen, Netzwerkverbindungen

1 · Überblick

Resourcecloud wird als ein Container-Image ausgeliefert, das die Anwendung (Apache und PHP 8.5) enthält. Daran hängen eine MariaDB-Datenbank, ein Scheduler-Container und ein S3-kompatibler Objektspeicher für hochgeladene Dateien. Konfiguriert wird ausschließlich über Umgebungsvariablen; im Image wird nichts angepasst.

Dienst	Aufgabe
<code>rescloud-web</code>	Apache und PHP. Legt beim Start das Datenbankschema an bzw. migriert es und liefert dann die Anwendung aus. Meldet sich unter <code>/healthz.php</code> .
<code>rescloud-mysql</code>	MariaDB 12.3 (mindestens 11.8, weil der KI-Index VECTOR-Spalten braucht).
<code>rescloud-cron</code>	Dasselbe Image, nur der Scheduler: Mailversand, Postfachabruf, Eskalationen, Statistiken, Lizenzmeldung. Ohne ihn werden keine Mails verschickt und nichts Geplantes läuft, die Anwendung sieht aber gesund aus.
<code>ollama</code> , <code>reranker</code> (optional)	Lokale KI-Dienste für Dokumentensuche und Assistent. Nur nötig, wenn keine Daten an einen externen KI-Anbieter gehen sollen (Abschnitt 5.4).

Ein Betreiber

Sie betreiben, sichern und aktualisieren die Instanz selbst. Ontaric liefert Image, Compose-Datei, Skripte und Release-Notes.

Keine Daten bei Ontaric

Ihre Daten liegen in Ihrer Datenbank und Ihrem S3-Speicher. Es gibt genau eine Verbindung nach außen, die zu Ontaric führt: die Lizenzmeldung (Abschnitt 5.3).

Eine Instanz je Mandant

Jede Instanz hat eigene Datenbank, eigenen Speicher und ein eigenes Netz. Mehrere Instanzen (z. B. Test und Produktion) laufen nebeneinander.

2 · Voraussetzungen

2.1 Server

	Minimum	Empfohlen
CPU / RAM	2 vCPU / 4 GB	4 vCPU / 8 GB; mit lokalen KI-Modellen zusätzlich 2–3 GB
Festplatte	20 GB	50 GB und mehr (Datenbank, Zwischenspeicher, Images)
Betriebssystem	Linux, 64 Bit (Debian oder Ubuntu LTS; die Skripte für Weg B setzen eines von beiden voraus)	
Software	Docker Engine 24 oder neuer mit dem Compose-Plugin (<code>docker compose version</code>), oder ein Kubernetes-Cluster ab 1.25	

2.2 Was Sie vorab bereitstellen

Was	Details
Hostname und Zertifikat	Ein DNS-Name (z. B. <code>rescloud.firma.de</code>) und ein TLS-Zertifikat. Das Zertifikat wird am vorgeschalteten Proxy bzw. Load Balancer eingebunden (Abschnitt 3.1) oder vom mitgelieferten Traefik per Let's Encrypt bezogen (Abschnitt 3.2, braucht öffentlich erreichbare Ports 80 und 443).
Zugang zum Image	Das Image liegt in der Registry <code>dockerreg.jannedy.de</code> und ist nicht öffentlich. Ontaric gibt Ihnen einen eigenen Zugang (Benutzername und Token, nur lesend), mit dem Sie sich per <code>docker login</code> anmelden (Abschnitt 3.0). Auf Wunsch gibt es ein Image-Archiv zum Einspielen ohne Internet.
S3-Speicher	Ein privater Bucket mit Zugangsschlüssel (Lesen, Schreiben, Löschen nur auf diesen Bucket). Hinweise in Abschnitt 5.1.
Mailserver	Ein SMTP-Konto, über das die Anwendung senden darf (Abschnitt 5.2).
Lizenzdaten	Kundenkennung und Token, die Ihnen Ontaric nach Vertragsschluss nennt (Abschnitt 5.3).
Je nach Bedarf	KI-Anbieter oder lokale Hardware (5.4) · Entra-ID-Administrator für Microsoft 365 (5.5) · Identitätsanbieter für Single Sign-On (5.6).

2.3 Netzwerk: wer muss wohin

Eingehend benötigen Sie nur HTTPS (443) zu Ihrem Proxy. Alle Verbindungen, die die Instanz selbst nach außen aufbaut, stehen in Anhang B. Wer eine strenge Firewall betreibt, gibt diese Ziele frei oder trägt einen Proxy ein (Abschnitt 5.7).

3 · Installation

Drei Wege, je nach Umgebung. Der erste ist der übliche im Rechenzentrum, der zweite für eine einzelne VM mit Internetzugang, der dritte für vorhandene Kubernetes-Cluster. Danach geht es für alle gleich mit Abschnitt 4 weiter.

3.0 Image beziehen

Ontaric nennt Ihnen Benutzernamen (`customer-<ihr-name>`) und Token. Der Zugang erlaubt nur das Herunterladen des Anwendungs-Images. Melden Sie sich einmal je Server an; Docker merkt sich die Anmeldung:

```
docker login dockerreg.jannedy.de -u customer-<ihr-name>      # Token als Passwort eingeben
docker pull dockerreg.jannedy.de/cptsolo/rescloud:5.3.0
docker pull mariadb:12.3
```

Verwenden Sie eine **Versionsnummer** (`5.3.0`), nicht `latest` : So wissen Sie, was läuft, und ein Update geschieht, wann Sie es entscheiden. Der Zugang kann ein Ablaufdatum haben; fragen Sie rechtzeitig vorher nach einem neuen Token. Ist er abgelaufen oder gesperrt, laufen bestehende Instanzen weiter, Updates lassen sich aber nicht mehr laden.

Ohne Internetzugang: Ontaric liefert das Image als Archiv. Einspielen mit `docker load -i rescloud-5.3.0.tar` . Mit einer eigenen Registry (Harbor, Artifactory o. ä.) setzen Sie `RC_IMAGE` und `DB_IMAGE` in der Umgebungsdatei auf die Adressen in Ihrer Registry. Die optionalen KI-Images (`ollama/ollama` , `ghcr.io/ggml-org/llama.cpp`) sind in der Compose-Datei fest eingetragen und müssen dort bei Bedarf ebenfalls angepasst werden.

3.1 Weg A: Docker Compose hinter Ihrem Proxy (Standard im Rechenzentrum)

Die Anwendung läuft per HTTP auf einem lokalen Port. Ihr Proxy oder Load Balancer terminiert TLS und leitet an diesen Port weiter.

1. **Docker installieren** nach `docs.docker.com/engine/install` und prüfen: `docker --version` und `docker compose version` .
2. **Dateien bereitstellen.** Von Ontaric erhalten Sie `docker-compose.prod.yml` und `docker-compose.localtest.yml` (der Name „localtest“ ist historisch; die Datei veröffentlicht den Webport). Mehr brauchen Sie nicht, die Anwendung steckt im Image.

```
mkdir -p /opt/rescloud && cd /opt/rescloud
# beide Dateien hierher kopieren
```

3. **Umgebungsdatei `rescloud.env` anlegen:**

```
cat > resccloud.env <<EOF
RC_IMAGE=dockerreg.jannedy.de/cptsolo/resccloud:5.3.0
DB_NAME=resccloud
DB_USER=resccloud
DB_PASSWORD=$(openssl rand -hex 16)
DB_ROOT_PASSWORD=$(openssl rand -hex 16)
# Adresse, unter der die Anwender die Anwendung aufrufen (https, ohne Schrägstrich am Ende)
BASE_URL=https://resccloud.firma.de
ACCESS_METHOD=https
# Der Proxy spricht diesen Port auf dem Server an
HTTP_PORT=8080
# Von der Compose-Datei verlangt; bei eigenem Proxy ohne weitere Bedeutung
TENANT=resccloud
TENANT_HOST=resccloud.firma.de
EOF
chmod 600 resccloud.env
```

Sichern Sie diese Datei. Sie enthält die einzigen Kopien der Datenbankpasswörter. `BASE_URL` und `ACCESS_METHOD` müssen exakt zu dem passen, was der Anwender im Browser sieht, sonst zeigen Links in Mails und manche Weiterleitungen ins Leere.

4. **Netzwerk anlegen** (einmal je Server; die Compose-Datei erwartet es, auch wenn Sie keinen Traefik nutzen):

```
docker network create resccloud-edge
```

5. **Starten:**

```
docker compose -f docker-compose.prod.yml -f docker-compose.localtest.yml \
  --env-file resccloud.env -p resccloud up -d \
  --scale ollama=0 --scale ollama-init=0
```

Die beiden `--scale`-Angaben lassen die lokalen KI-Dienste weg (spart einen Download von rund 1 GB; siehe 5.4). Wiederholen Sie die Angaben bei jedem späteren `up`.

6. **Warten, bis die Instanz gesund ist.** Der erste Start legt das Schema an und dauert etwa eine Minute.

```
docker compose -p resccloud ps
curl -s http://localhost:8080/healthz.php      # {"status":"ok"}
```

Jeder andere Status nennt die fehlgeschlagene Prüfung (siehe Abschnitt 8).

7. **Proxy einrichten:** Weiterleitung von `https://resccloud.firma.de` an `http://<server>:8080`. Der Proxy muss den `Host`-Header und `X-Forwarded-Proto: https` weitergeben. Für den Gesundheitstest des Proxys eignet sich `/healthz.php`.

3.2 Weg B: Einzelne VM mit automatischem HTTPS

Für eine frische Debian- oder Ubuntu-VM mit Internetzugang. Die Skripte installieren Docker, starten Traefik, das Zertifikate bei Let's Encrypt bezieht und erneuert, und richten die Instanz dahinter ein.

1. **DNS:** A-Record auf die Server-IP, Ports 80 und 443 von außen erreichbar. Ohne auflösbaren Namen und offenen Port 80 wird kein Zertifikat ausgestellt.
2. **Dateien kopieren** (Ordner `deploy/` und `docker-compose.prod.yml`, Struktur beibehalten):

```
/opt/rescloud-src/  
├─ docker-compose.prod.yml  
├─ docker-compose.localtest.yml  
└─ deploy/
```

3. **Server vorbereiten** (installiert Docker, begrenzt Container-Logs, legt das Netz an, startet Traefik; mehrfach ausführbar):

```
cd /opt/rescloud-src/deploy  
sudo bash bootstrap-server.sh --acme-email admin@firma.de --staging
```

`--staging` nutzt die Test-CA von Let's Encrypt: Browser warnen, aber ein DNS-Fehler kann das Limit der echten CA nicht aufbrauchen. Mit `--firewall` wird der Server zusätzlich auf SSH, HTTP und HTTPS eingeschränkt.

4. **Instanz anlegen** (schreibt `/opt/rescloud/customers/main.env` mit zufälligen Passwörtern, startet und wartet auf „gesund“):

```
./new-customer.sh main rescloud.firma.de --no-ai --start
```

Ohne `--admin-email`, `--admin-name` und `--company` erscheint der Einrichtungsassistent (Abschnitt 4). Mit allen dreien richtet sich die Instanz selbst ein und zeigt das Administratorpasswort einmalig an. `--no-ai` lässt die lokalen KI-Dienste weg.

5. **Echte Zertifikate**, sobald die Seite lädt:

```
sudo bash bootstrap-server.sh --acme-email admin@firma.de  
docker compose -p traefik up -d --force-recreate
```

3.3 Weg C: Kubernetes

Das Manifest `rescloud-k8s.yaml` legt Namespace `rescloud`, Datenbank (StatefulSet), Web- und Scheduler-Deployment, Volumes, Services und Ingress an. Alle Stellen, die Sie anpassen müssen, sind mit `# <-- CHANGE:` markiert.

Objekt	Einstellung
Secret rescloud-db	DB_PASSWORD , DB_ROOT_PASSWORD : lange Zufallswerte
ConfigMap rescloud-config	rc_baseurl (öffentliche URL), rc_mailsender
Deployments	image : dockerreg.jannedy.de/cptsolo/rescloud:5.3.0 (bei Web und Cron gleich, ggf. aus Ihrer Registry); dazu imagePullSecrets ergänzen (das Manifest enthält keinen): kubectl -n rescloud create secret docker-registry regcred --docker-server=dockerreg.jannedy.de --docker-username=customer-<ihr-name> --docker-password=<token>
Ingress	Hostname, TLS-Host, Ingress-Klasse, Zertifikatsaussteller

```
kubectl apply -f rescloud-k8s.yaml
kubectl -n rescloud get pods -w
kubectl -n rescloud logs deploy/rescloud-web -f
```

- Eine Web-Replik mit Strategie *Recreate*. Mehrere Repliken brauchen eine ReadWriteMany-Storage-Klasse für Uploads und Sitzungen. Migrationen sind per Datenbanksperr serialisiert.
- Genau ein `rescloud-cron` ; nie hochskalieren. Er läuft auf demselben Knoten wie der Web-Pod, weil beide das Upload-Volume teilen.
- Datenbankpasswörter gehören in Ihren Secret-Manager, nicht in ein eingetragtes Manifest.

4 • Ersteinrichtung

Auf einer leeren Datenbank öffnet die Anwendung den Einrichtungsassistenten unter `/basic_setup` . Er läuft genau einmal: sobald ein Team und ein Benutzer existieren, führt dieselbe Adresse zur normalen Anmeldung.

1. **Benutzer:** Vor- und Nachname, Passwort (zweimal), E-Mail-Adresse. Diese Person wird erster Administrator und Teamleitung. Den Anmeldenamen bildet die Anwendung selbst: erster Buchstabe des Vornamens plus Nachname in Kleinbuchstaben (Anna Schmidt → `aschmidt`). Ein Bild lassen Sie vorerst weg, damit vor der Speicheranbindung keine Datei geschrieben wird.
2. **Firma:** Name, Kürzel (höchstens sechs Buchstaben), Adresse, Land. Daraus entsteht die Startorganisation mit Bereich, Abteilung und Team sowie eine Willkommensaufgabe. Alles lässt sich später in der Organisationsverwaltung umbenennen.
3. **Fertigstellen.** Das eingebaute Admin-Konto wird aus Sicherheitsgründen deaktiviert; ab jetzt gilt der soeben angelegte Benutzer.

Anmelden: Feld „E-Mail“ nimmt den Anmeldenamen (`aschmidt`), nicht die Adresse. Sie landen bei „Meine Aufgaben“ mit der Willkommensaufgabe. Alle Einstellungen dieses Kapitels finden Sie unten im Menü unter *Admin*.

Automatisierte Einrichtung: Über die Variablen `rc_seed_admin_*` und `rc_seed_company_*` kann ein Deployment den ersten Administrator vorgeben; der Assistent erscheint dann nicht. Bei handgeschriebenen Umgebungsdateien muss jedes `$` im Passwort-Hash zu `$$` verdoppelt werden, sonst kommt der Hash gekürzt an. `new-customer.sh` übernimmt das.

5 · Erste Schritte nach der Installation

Reihenfolge: **zuerst der Dateispeicher**, bevor jemand etwas hochlädt; Mail und Lizenz direkt danach; KI, Microsoft 365 und Single Sign-On in der Reihenfolge, in der Ihr Projekt sie braucht. Alle Schritte brauchen die Rolle *RC Admin*.

5.1 Dateispeicher (S3)

Ohne Anbindung landen Anhänge, eingebettete Bilder und Avatare auf der lokalen Platte des Web-Containers, und ein Container-Neustart oder eine Umplanung kann sie verlieren. Bereiten Sie vorab bei Ihrem Speicheranbieter vor: einen **privaten Bucket** (nie öffentlich, Downloads laufen über kurzlebige signierte Links), einen **Zugangsschlüssel** mit Lese-, Schreib- und Löschrecht nur auf diesen Bucket, **Endpunkt** und **Region** und optional ein **Präfix**, damit mehrere Instanzen einen Bucket teilen können.

Anbieter	Endpunkt	Region	Pfad-Stil
Amazon S3	leer lassen	Region des Buckets, z. B. eu-central-1	aus
Eigener Speicher (MinIO, Ceph RGW u. a.)	z. B. https://s3.firma.de	Region des Servers (üblich: us-east-1)	an
Hetzner Object Storage	https://fsn1.your-objectstorage.com	fsn1, nbg1 oder hel1	an

Die Region muss zum Endpunkt passen, sie geht in die Signatur ein; eine falsche wird abgelehnt, bevor der Bucket überhaupt geprüft wird.

1. *Admin* → *Globale Einstellungen* → *Datenverwaltung*, Karte *Externer Speicher*: **Externen S3-Speicher verwenden** einschalten.
2. Endpunkt, Bucket, Region, Access Key und Secret Key eintragen; Pfad-Stil wie in der Tabelle.
3. **Verbindung testen**. Der Test schreibt, liest und löscht ein Probeobjekt und meldet das Ergebnis. Fehlt der Bucket, bietet der Test an, ihn anzulegen.
4. **Speicher-Einstellungen speichern**. Das Secret wird in der Datenbank abgelegt und nie wieder angezeigt; bei späterem Speichern leer lassen, um es zu behalten.

Prüfen Sie Ende-zu-Ende: Datei an die Willkommensaufgabe hängen und im Bucket nachsehen. Waren schon Dateien hochgeladen, bietet die Karte *Lokale Dateien nach S3 kopieren* an. Eine unvollständige Konfiguration fällt still auf lokalen Speicher zurück; verlassen Sie sich daher auf den bestandenen Test.

Keine Internetverbindung? Sie benötigen nur einen S3-Endpunkt, den die Instanz im Netz erreicht, nicht notwendig einen öffentlichen.

Der Bucket nimmt außerdem den Prüfschlüssel des Änderungsprotokolls und dessen tägliche Ankerkopien auf, unter `audit/` (unterhalb Ihres Präfixes). Richten Sie für dieses Präfix keine Ablauf- oder Lifecycle-Regel ein, und nehmen Sie es in Ihre Bucket-Sicherung oder Replikation auf.

Direkt danach: Audit-Schlüssel herunterladen

Resourcecloud führt ein Änderungsprotokoll darüber, wer was geändert hat. Es lässt sich nicht bearbeiten, und eine Prüfkette mit einem geheimen Schlüssel macht jede spätere Änderung erkennbar. Resourcecloud legt diesen Schlüssel selbst an und hält ihn im soeben angebundenen Bucket – **Sie brauchen aber eine eigene Kopie**. Ohne sie lässt sich das Protokoll nach einer Wiederherstellung, einem Wechsel des Buckets oder einem ersetzten Schlüssel nicht mehr prüfen. Als RC Admin sind Sie für diese Kopie verantwortlich.

1. *Admin* → *Änderungsprotokoll* öffnen. Ein gelber Hinweis lautet *Audit-Schlüssel herunterladen und sicher aufbewahren*.
2. **Schlüssel herunterladen** und die Datei dort ablegen, wo Sie wichtige Geheimnisse aufbewahren (Passwortmanager, Tresor), nicht nur auf dem Server und nicht im Datenbank-Backup.
3. „**Ich habe ihn gesichert**“ anklicken. Der Hinweis verschwindet, und die Seite zeigt, wer die Kopie bestätigt hat. Nur ein RC Admin kann bestätigen; die Rolle Auditor darf ebenfalls herunterladen.

Erledigen Sie das jetzt, solange noch nichts davon abhängt. Setzen Sie `RC_AUDIT_KEY` selbst in der Umgebungsdatei, erscheint kein Hinweis; bewahren Sie dann diesen Wert bei Ihren übrigen Geheimnissen auf. Mehr in der Anleitung *Änderungsprotokoll* (`docs/auditlog/index.html` , auch als PDF).

5.2 Mailserver

Globale Einstellungen → *Externe Dienste* → *Mailserver-Konfiguration*: SMTP-Server und Port (587 mit STARTTLS oder 465 mit implizitem TLS), Benutzer und Passwort, Authentifizierung und TLS je nach Server, Absenderadresse (viele Server verlangen die des Kontos oder der Domäne), Markenname. Speichern und **Testmail senden**.

Der Scheduler leert die Mail-Warteschlange; die Testmail kann daher bis zu fünf Minuten brauchen. Bleibt das Feld leer, fällt die Instanz auf einen Standard-Server zurück, über den im eigenen Rechenzentrum nichts zugestellt wird.

5.3 Lizenz

Ihre Instanz meldet Ontaric die Anzahl der Plätze (aktive Benutzer), damit nach dem Monatsmaximum abgerechnet werden kann. Die Meldung geht von Ihrem Server nach außen, sodass kein Zugriff in Ihr Netz nötig ist.

Nach Vertragsschluss nennt Ihnen Ontaric drei Zeilen. In die Umgebungsdatei eintragen:

```
rc_license_customer=<Kundenkennung>
rc_license_token=<Token>
rc_license_report_url=https://licensing.resourcecloud.io/api/report
```

Umgebungsvariablen werden beim **Erzeugen** des Containers fixiert; ein Neustart liest die alten Werte erneut. Daher erneut „hochfahren“, nicht neu starten:

```
docker compose -f docker-compose.prod.yml -f docker-compose.localtest.yml \
  --env-file rescloud.env -p rescloud up -d --scale ollama=0 --scale ollama-init=0
```

Die Ausgabe muss *Recreated* nennen; bei *Running* hat Compose keine Änderung bemerkt. Prüfung: `docker exec rescloud-rescloud-cron-1 printenv | grep rc_license` zeigt drei Zeilen. Die Meldung läuft alle 15 Minuten.

Was übertragen wird. Mit Token: die Kundenkennung und die Platzzahl des laufenden und des Vormonats. Ohne Token meldet sich die Instanz alle 15 Minuten anonym mit Basis-URL, Hostname, Platzzahl und Schemaversion, damit nicht lizenzierte Installationen sichtbar werden. Inhalte, Personendaten und Kundendaten werden in keinem Fall übertragen. Möchten Sie die Meldung nicht zulassen, sprechen Sie das vorab mit Ontaric ab.

5.4 KI-Anbindung (optional)

Zwei Funktionen nutzen KI: die semantische Suche und Aufwandsschätzung (Einbettungen) sowie der Chat-Assistent. Ohne Einrichtung arbeitet eine schlagwortbasierte Suche. Der Assistent bekommt bewusst keine namentlichen Leistungsranglisten einzelner Personen, unabhängig vom gewählten Anbieter.

Betriebsart	Was die Instanz verlässt	Aufwand
Vollständig lokal (ollama und Reranker im selben Compose-Projekt, Chat über ein eigenes Modell, z. B. vLLM, Ollama oder einen internen OpenAI-kompatiblen Dienst)	Nichts. Mit <code>AI_LOCAL_ONLY=1</code> darf kein KI-Aufruf die Instanz verlassen; eine Funktion, die noch nach außen zeigt, wird nicht eingeschaltet.	Eigene Hardware; Einbettungen laufen auf CPU, für den Chat braucht es ein Modell auf eigener GPU
Lokale Einbettungen, externer Chat	Fragen und die dazu ausgewählten Textstellen gehen an den Chat-Anbieter	Mittel
Extern (OpenAI, Gemini, Anthropic für den Chat)	Texte für Einbettungen und Chat	Gering

Lokale Dienste starten

In der Umgebungsdatei (`new-customer.sh`) schreibt sie von selbst, wenn Sie `--no-ai` weglassen):

```
OLLAMA_EMBED_MODEL=embeddinggemma
AI_EMBED_BASE_URL=http://ollama:11434/v1
COMPOSE_PROFILES=reranker
AI_RERANK_BASE_URL=http://reranker:8080/v1
# optional: nichts darf die Instanz verlassen
AI_LOCAL_ONLY=1
```

Dann `up -d` ohne die `--scale` -Angaben. Beim ersten Start trägt die Instanz diese Adressen in *Externe Dienste* → *Konfiguration des KI-Dienstes* ein, und zwar nur in leere Felder und nur einmal; was ein

Administrator später ändert, bleibt. Modelle: `embeddinggemma` (etwa 1,1 GB RAM beim Einbetten) oder `bge-m3` (etwa 1,7 GB, etwas besser); `nomic-embed-text` taugt für deutsche Inhalte nicht. Ollama lädt das Modell beim ersten Start (0,3 bis 1,7 GB), der Reranker rund 440 MB von Hugging Face. **Ohne Internetzugang** müssen die Modelldateien vorher in die Volumes kopiert werden; fragen Sie dazu bei Ontaric nach.

Chat-Endpunkt eintragen

Globale Einstellungen → *Externe Dienste* → *Konfiguration des KI-Dienstes*: API-Schlüssel, Basis-URL, Chat-Modell; die Schalter für Schätzung/Suche und Chat-Assistent einschalten und speichern. Jeder OpenAI-kompatible Endpunkt funktioniert. Anthropic bietet keine Einbettungen an; kombinieren Sie sie mit einem lokalen Dienst oder einem anderen Anbieter. Das Feld „Proxy“ derselben Karte nutzen Sie, wenn die KI-Aufrufe über einen Proxy gehen müssen.

5.5 Microsoft 365: Kalender und Teams

Stand: Die Anbindung an Microsoft Graph wurde gegen die dokumentierte Schnittstelle entwickelt und mit einem simulierten Graph-Dienst getestet, aber noch nicht gegen einen echten Microsoft-365-Tenant. Probieren Sie sie in einem Testtenant aus, bevor Sie sie für alle einschalten.

Die Verbindung wird **einmal vom Administrator für die ganze Instanz** hergestellt. Niemand verbindet sein eigenes Microsoft-Konto, und Resourcecloud fragt nie nach einem Microsoft-Passwort. Dazu gibt es einen Teil ohne jede Einrichtung: Jeder Anwender kann in seinem Profil unter *Kalender-Abo* einen privaten Link erzeugen und in Outlook, Google Kalender oder Apple Kalender abonnieren.

Benötigte Rechte: Entra-Rolle zum Registrieren von Anwendungen und für die Administratorzustimmung (Anwendungsadministrator plus Globaler Administrator oder Administrator für privilegierte Rollen), Exchange-Online-Administrator für die Zugriffsrichtlinie, RC Admin in Resourcecloud. Die Instanz muss `login.microsoftonline.com` und `graph.microsoft.com` per HTTPS erreichen.

1. **App in Entra ID registrieren.** Entra Admin Center → App-Registrierungen → Neue Registrierung, ein Mandant, keine Umleitungs-URI. Unter API-Berechtigungen → Microsoft Graph → **Anwendungsberechtigungen** hinzufügen und die Administratorzustimmung erteilen:

Berechtigung	wofür
<code>Calendars.ReadWrite</code>	Termine anlegen, verschieben, absagen; Antworten lesen; Frei/Gebucht
<code>OnlineMeetings.ReadWrite.All</code>	Teams-Besprechungen
<code>OnlineMeetingArtifact.Read.All</code>	Teams-Anwesenheitsbericht

Unter Zertifikate & Geheimnisse einen **Clientschlüssel** erzeugen und den Wert sofort kopieren; Verzeichnis-ID (Tenant) und Anwendungs-ID (Client) von der Übersichtsseite notieren.

2. **App auf die richtigen Postfächer beschränken.** `Calendars.ReadWrite` als Anwendungsberechtigung erreicht ohne Einschränkung jedes Postfach des Tenants. Tun Sie das,

bevor Sie etwas einschalten. In der Exchange-Online-PowerShell:

```
New-ApplicationAccessPolicy -AppId <Client-ID> `
  -PolicyScopeGroupId rc-calendar-users@firma.de `
  -AccessRight RestrictAccess -Description "Resourcecloud Kalender"
Test-ApplicationAccessPolicy -AppId <Client-ID> -Identity jemand@firma.de
```

Die Gruppe ist eine E-Mail-aktivierte Sicherheitsgruppe mit den Resourcecloud-Anwendern. Die Richtlinie kann bis zu einer Stunde brauchen. Für Teams-Besprechungen gilt zusätzlich die Teams-Anwendungszugriffsrichtlinie (`New-CsApplicationAccessPolicy` , `Grant-CsApplicationAccessPolicy`) für die Organisatoren.

3. **In Resourcecloud eintragen.** *Globale Einstellungen* → *Externe Dienste* → *Videokonferenz*: Tenant-ID, Client-ID, Clientschlüssel (Felder von Microsoft Teams). Optional eine Organisator-UPN als Ersatz für Organisatoren ohne Postfach.

4. **Schalter, aufeinander aufbauend:**

Schalter	Wirkung	Bedenken
1 · Besprechungen in Outlook-Kalender schreiben	Besprechungen werden Outlook-Termine mit Teams-Link; Exchange verschickt die Einladungen; Antworten kommen zurück.	Beim Einschalten erhalten die Teilnehmer der Besprechungen der nächsten 90 Tage eine Einladung. Wählen Sie einen ruhigen Zeitpunkt oder kündigen Sie es an.
2 · Abwesenheiten als „Abwesend“ (setzt 1 voraus)	Genehmigter Urlaub und Krankheitstage erscheinen als ganztägiges „Abwesend“. Nur Daten, kein Grund, keine Art. Jeder kann im Profil widersprechen.	Sichtbar für alle, die den Kalender oder Frei/Gebucht der Person sehen.
3 · Belegte Zeit senkt die Kapazität (setzt 1 voraus, optional)	Für Personen, die zugestimmt haben, werden die belegten Minuten je Arbeitstag gelesen und von der Kapazität abgezogen.	Mit dem Betriebsrat abstimmen. Nur mit Einwilligung je Person, nur Zahlen, nie Inhalte.

Zuordnung der Personen: über die Entra-Objekt-ID, wenn das Konto per SCIM angelegt wurde, sonst über die E-Mail-Adresse im Profil. Gäste ohne Konto werden nicht als Teilnehmer eingetragen, sie behalten die Einladung mit ihrem persönlichen Link. Vollständige Beschreibung mit Fehlersuche: *Kalender-Anbindung (Calendar Integration Guide)*.

5.6 Single Sign-On und Benutzeranlage (optional)

Anmeldung über Ihren Identitätsanbieter (OpenID Connect für Entra ID, Keycloak, Okta u. a.; SAML 2.0 für ADFS, Shibboleth u. a.) und automatisches Anlegen, Ändern und Deaktivieren von Konten samt Teamzuordnung per SCIM 2.0. Die Passwortanmeldung bleibt daneben bestehen, eine falsche Einstellung sperrt also niemanden aus.

1. *Globale Einstellungen* → *Externe Dienste* → *Single Sign-On (OpenID Connect)* öffnen und die dort angezeigte **Umleitungs-URI** kopieren (`https://rescloud.firma.de/oidc_callback.php`).
2. Beim Identitätsanbieter Resourcecloud als Web-Anwendung (vertraulicher Client) mit genau dieser URI registrieren; Client-ID, Client-Secret und Issuer-URL notieren.
3. Issuer, Client-ID und Secret eintragen, Anspruchszuordnung prüfen, Single Sign-On einschalten und speichern. Die Anwendung fragt den Anbieter vorab und speichert nicht, wenn er nicht antwortet.
4. Im **privaten Fenster** testen, während Sie im normalen Fenster als RC Admin angemeldet bleiben.

Voraussetzungen: https-Adresse für Resourcecloud, ausgehendes HTTPS vom Server (nicht vom Browser) zum Anbieter, mindestens ein RC Admin mit Passwort als Rückweg. Bei SAML signiert der Anbieter mit SHA-256 (SHA-1 wird abgelehnt). Ausführlich, mit Entra, Keycloak, ADFS und SCIM: *Single Sign-On und Provisioning (SSO Guide)*.

5.7 Organisation, Benutzer und Proxy

- **Organisation:** *Admin* → *Organisationsverwaltung*. Die Struktur ist eine Leiter von der Firma über Bereiche und Abteilungen zu den Teams. Beginnen Sie mit den Teams, die Sie sicher brauchen.
- **Benutzer:** Tab *Benutzer* → *Benutzer hinzufügen* oder *Massenimport*. Resourcecloud erzeugt ein Einmalpasswort, das nur einmal angezeigt wird; die Person wählt beim ersten Anmelden ein eigenes. Die Wochenarbeitszeit speist die Kapazitätsplanung, bei Teilzeit bitte realistisch eintragen. Wer von Jira kommt, nutzt den Datenimport (*Admin* → *Datenimport*, Anleitung „Von Jira wechseln“).
- **Auswertungen einzelner Personen** (Berichte und Auslastung) sind **standardmäßig aus**. Wer sie einschalten will, erhält eine Warnung; mit dem Einschalten wird jeder Beschäftigte benachrichtigt, und der Hinweis steht in jedem Profil. Klären Sie das vorher mit Ihrem Betriebsrat (siehe das Betriebsrats-Paket).
- **Ausgehender Proxy:** *Globale Einstellungen* → *Konfiguration* → *Proxy-Konfiguration* trägt einen HTTP-Proxy für die Verbindungen der Anwendung nach außen ein. Der Docker-Daemon braucht für Image-Downloads einen eigenen Proxy-Eintrag.

Checkliste vor der Einladung der ersten Kollegen

- ☐ `/healthz.php` antwortet `{"status":"ok"}`, `rescloud-cron` läuft
- ☐ S3-Verbindungstest bestanden, Datei an Aufgabe angehängt und im Bucket gefunden
- ☐ Audit-Schlüssel heruntergeladen, bei den Geheimnissen abgelegt und mit „Ich habe ihn gesichert“ bestätigt (*Admin* → *Änderungsprotokoll*)
- ☐ Testmail angekommen
- ☐ Lizenzzeilen eingetragen, Container neu erzeugt, Meldung gelaufen
- ☐ KI bewusst eingerichtet oder bewusst aus
- ☐ Microsoft 365 / Single Sign-On im Test geprüft (falls genutzt)

- ☐ Sicherung der Datenbank, des Buckets und der Umgebungsdatei eingerichtet und einmal zurückgespielt (Abschnitt 6)
- ☐ Mindestens zwei RC Admins mit Passwortanmeldung

6 • Betrieb und Sicherung

6.1 Was gesichert wird

Was	Wo	Hinweis
Datenbank	Volume <code>rescloud-db</code>	Als Dump sichern (unten). Ein Kopieren des Dateisystems bei laufender Datenbank ist keine Sicherung.
Hochgeladene Dateien	S3-Bucket	Versionierung oder Replikation des Speichers nutzen. Das Volume <code>rescloud-userdata</code> hält nur Zwischenspeicher und alles, was vor der Speicheranbindung geschrieben wurde.
Audit-Schlüssel	Datei, die Sie heruntergeladen haben (5.1); zusätzlich im Bucket unter <code>audit/</code>	Bei den Geheimnissen ablegen, getrennt vom Server. Datenbank und Bucket gemeinsam sichern.
Umgebungsdatei	<code>rescloud.env</code> bzw. <code>/opt/rescloud/customers/*.env</code>	Einzige Kopie der Datenbankpasswörter, getrennt vom Server ablegen.
Zertifikate (nur Weg B)	Volume <code>traefik-letsencrypt</code>	
Sitzungen	Volume <code>rescloud-sessions</code>	Nicht sichern. Verlust kostet jeden eine Anmeldung.

```
# Datenbank sichern (täglich, per Cron; mariadb-dump, nicht mysqldump: MariaDB 12 kennt den alten Namen nicht mehr)
docker exec rescloud-rescloud-mysql-1 sh -c \
  'mariadb-dump -uroot -p"$MARIADB_ROOT_PASSWORD" --single-transaction --routines "$MARIADB_DATAB
ASE"' \
  | gzip > rescloud-$(date +%F).sql.gz
```

6.2 Zurückspielen

```
docker compose -p rescloud stop rescloud-web rescloud-cron
gunzip -c rescloud-2026-10-10.sql.gz | docker exec -i rescloud-rescloud-mysql-1 sh -c \
'mariadb -uroot -p"$MARIADB_ROOT_PASSWORD" "$MARIADB_DATABASE"'
docker compose -p rescloud start rescloud-web rescloud-cron
```

Wiederherstellung gehört in einen Test **vor** dem Ernstfall: auf einem zweiten Server, mit der Umgebungsdatei, dem Dump und dem Bucket. Spielen Sie Datenbank und Bucket zusammen zurück: Dann liegt auch der Prüfschlüssel des Änderungsprotokolls wieder bereit. Kommt die Datenbank in einen neuen, leeren Bucket, fügt ein RC Admin den heruntergeladenen Schlüssel unter *Änderungsprotokoll* → *Aufbewahrung und Datenschutz* → *Älteren Schlüssel hinzufügen* ein. Der Web-Container meldet sich erst gesund, wenn das Schema zur Version des Images passt.

6.3 Überwachen

- **Gesundheit:** `/healthz.php` liefert 200 nur, wenn Konfiguration, Datenbank und Schema in Ordnung sind; sonst 503 mit der Bezeichnung der fehlgeschlagenen Prüfung.
- **Scheduler:** Steht er 15 Minuten, sehen alle Administratoren ein rotes Banner. Zusätzlich kann `RC_CRON_PING_URL` auf einen Totmannschalter zeigen (Healthchecks, Uptime Kuma o. ä.); nach jedem Durchlauf wird gemeldet, bei Fehlern mit angehängtem `/fail`. Der Container selbst kann nicht melden, dass er steht, deshalb ist das die wichtigere Überwachung.
- **Logs:** `docker compose -p rescloud logs -f rescloud-web` bzw. `rescloud-cron`; Docker-Logs auf eine sinnvolle Größe begrenzen (Rotation).
- **Speicherplatz:** Datenbank-Volume und Docker-Verzeichnis beobachten. `docker image prune` entfernt alte Images nach Updates.

7 • Update

Ein Update tauscht das Image aus. Beim Start migriert der Web-Container das Datenbankschema und meldet sich erst dann gesund. Ontaric liefert zu jeder Version Release-Notes; lesen Sie sie vor dem Update, vor allem Hinweise zu neuen Umgebungsvariablen, geänderter Compose-Datei oder neuen Mindestversionen von MariaDB.

7.1 Vor dem Update

1. **Release-Notes lesen.** Gibt es eine neue Compose-Datei oder neue Skripte, ersetzen Sie die Kopien auf dem Server vorher (die Anwendung ist im Image, die Compose-Datei nicht). Neue Volumes, Variablen oder Proxy-Labels wirken erst damit.
2. **Datenbank sichern** (Abschnitt 6.1) und prüfen, dass die Datei nicht leer ist. Der Dump ist Ihr Weg zurück, falls eine Migration nicht rückgängig zu machen ist.
3. **Aktuellen Stand notieren:**

```
docker inspect --format '{{index .RepoDigests 0}}' dockerreg.jannedy.de/cptsolo/rescloud:5.3.0
docker compose -p rescloud exec rescloud-web head -6 /var/www/html/htdocs/CHANGELOG
```

Der Digest (`sha256:...`) bleibt unveränderlich und führt Sie zum exakt gleichen Image zurück, auch wenn jemand ein Tag später überschreibt.

4. **Zeitpunkt wählen.** Je nach Weg gibt es eine kurze Unterbrechung (Weg A) oder keine (Weg B, Abschnitt 7.2).
5. **Erst die Testinstanz**, dann die Produktion, wenn Sie beides betreiben.

7.2 Update durchführen

Weg A: Docker Compose hinter eigenem Proxy

Neue Versionsnummer in `RC_IMAGE` der Umgebungsdatei eintragen (z. B. `dockerreg.jannedy.de/cptsolo/rescloud:5.4.0`), dann:

```
cd /opt/rescloud
docker compose -f docker-compose.prod.yml -f docker-compose.localtest.yml --env-file rescloud.env
-p rescloud pull rescloud-web
docker compose -f docker-compose.prod.yml -f docker-compose.localtest.yml \
--env-file rescloud.env -p rescloud up -d --scale ollama=0 --scale ollama-init=0
curl -s http://localhost:8080/healthz.php
```

Compose ersetzt Web- und Cron-Container. Während der Web-Container migriert und startet, ist die Anwendung kurz nicht erreichbar (gemessen: etwa 16 Sekunden bei einer kleinen Datenbank, bei größeren Datenbanken und Migrationen länger). Bis `/healthz.php` wieder `ok` meldet, antwortet sie mit 503. Angemeldete Anwender bleiben angemeldet, die Sitzungen liegen im eigenen Volume. Mit der KI: ohne die `--scale` -Angaben, wie bei der Installation.

Weg B: Einzelne VM mit Traefik (ohne Unterbrechung)

```
cd /opt/rescloud-src/deploy
./upgrade_tennants.sh --check      # zeigt, was sich ändern würde
./upgrade_tennants.sh main        # Update
# oder ein bestimmtes Image
./rollout.sh main dockerreg.jannedy.de/cptsolo/rescloud:5.4.0
```

Die Skripte `upgrade_tennants.sh` ziehen standardmäßig vom Docker Hub; für diese Registry setzen Sie vorher `export RC_UPGRADE_IMAGE=dockerreg.jannedy.de/cptsolo/rescloud:5.4.0`.

`rollout.sh` startet eine zweite Replik mit dem neuen Image, wartet, bis ihre Migration fertig ist und sie gesund meldet, nimmt dann die alte aus der Rotation und beendet sie. Wird die neue nie gesund, entfernt das Skript sie und lässt die alte weiterlaufen; es ändert dabei nichts am Kunden. Jedes Update wird in `/opt/rescloud/deployments.log` mit Digest festgehalten.

Weg C: Kubernetes

```
kubectl -n rescloud set image deploy/rescloud-web web=dockerreg.jannedy.de/cptsolo/rescloud:<neue-version>
kubectl -n rescloud set image deploy/rescloud-cron cron=dockerreg.jannedy.de/cptsolo/rescloud:<neue-version>
kubectl -n rescloud rollout status deploy/rescloud-web
```

Der neue Web-Pod migriert das Schema, bevor er *Ready* meldet. Web und Cron immer auf dieselbe Version setzen.

7.3 Nach dem Update prüfen

- `/healthz.php` liefert `{"status":"ok"}`, `docker compose -p rescloud ps` zeigt alle Dienste als gesund bzw. laufend.
- Anmelden, eine Aufgabe öffnen, eine Datei hochladen (S3), eine Mail auslösen.
- Die Lizenzmeldung läuft weiter (`docker compose -p rescloud logs rescloud-cron | grep -i license`).
- Alte Images aufräumen: `docker image prune`.

7.4 Rückfall

Die meisten Migrationen fügen nur Spalten oder Indizes hinzu; das alte Image läuft damit weiter. Das gilt **nicht immer**, die Release-Notes nennen Ausnahmen.

- **Update schlägt beim Start fehl** (Container wird nicht gesund): Das Log des Web-Containers nennt die Ursache. Bei Weg B und C bleibt die alte Version in Betrieb. Bei Weg A setzen Sie das alte Image per Digest zurück: `RC_IMAGE=dockerreg.jannedy.de/cptsolo/rescloud@sha256:...` in der Umgebungsdatei und erneut `up -d`.
- **Fehler zeigt sich erst im Betrieb**: Image per Digest zurücksetzen. Hat das neue Image das Schema verändert und kann das alte damit nicht umgehen, spielen Sie zusätzlich den Dump von vor dem Update zurück (Abschnitt 6.2). Das Ergebnis: die Daten seit dem Update gehen verloren, weshalb ein Update besser außerhalb der Hauptarbeitszeit läuft.

7.5 Datenbank-Update (MariaDB)

Die Version der Datenbank wird unabhängig vom Anwendungs-Image geführt (`DB_IMAGE` , Standard `mariadb:12.3`). Verlangt ein Release eine neuere Version, steht das in den Release-Notes; das Rollout-Skript weigert sich außerdem, ein Image zu starten, das eine neuere Datenbank braucht. **Ändern Sie die Versionsnummer nicht einfach in der Umgebungsdatei:** MariaDB unterstützt den direkten Sprung nur über eine Hauptversion, und 12.3 auf einem 10.11-Datenverzeichnis wird entweder abgelehnt oder schreibt es unumkehrbar um. Nutzen Sie das mitgelieferte Skript:

```
./upgrade_mariadb.sh main --check      # zeigt, was passieren würde
./upgrade_mariadb.sh main              # Dump, Tausch, Rückspielen, Prüfung
```

Es stoppt die Anwendung, sichert per Dump und als Archiv des Volumes, startet die neue Datenbank auf einem leeren Verzeichnis, spielt den Dump ein, vergleicht die Tabellenzahl und lässt erst dann die Anwendung wieder zu. Schlägt das fehl, geht es selbstständig zurück. Beide Sicherungen liegen unter `/opt/rescloud/backups/<instanz>/` .

8 • Fehlersuche

Symptom	Ursache und Abhilfe
<code>/healthz.php</code> → 503 <code>db_unreachable</code>	Datenbank noch nicht oben oder falsche <code>DB_*</code> -Werte. Ein nach dem ersten Start geändertes Passwort kennt die Datenbank nicht: entweder in MariaDB ändern oder das Volume neu anlegen (zerstört die Daten).
503 <code>schema_outdated</code> / <code>schema_missing</code>	Migration läuft noch oder ist fehlgeschlagen; Log des Web-Containers lesen. Beim ersten Start meist nach einer Minute vorbei.
503 <code>draining</code>	Die Replik wird gerade bei einem Rollout ersetzt. Kein Fehler.
Web-Container beendet sich sofort	Eine Pflichtvariable fehlt; das Log nennt sie (<code>DB_PASSWORD</code> , <code>TENANT</code> ...).
Weiterleitungen und Links in Mails zeigen auf eine falsche Adresse	<code>BASE_URL</code> und <code>ACCESS_METHOD</code> müssen zur Adresse im Browser passen (http/https, Host, Port). Beim Proxy zusätzlich <code>X-Forwarded-Proto</code> setzen.
Keine Mails	Der Reihe nach: läuft <code>rescloud-cron</code> ? SMTP-Einstellungen und Testmail, Log des Cron-Containers, Spam-Ordner.
Datei-Upload scheitert oder Dateien verschwinden	S3-Test in den Einstellungen wiederholen; Region zum Endpunkt, Pfad-Stil, Berechtigungen des Schlüssels prüfen.

Lizenz erscheint als „nicht lizenziert“ / „keine Meldung“	Drei <code>rc_license_*</code> -Zeilen im Cron-Container? (<code>printenv</code>). Container <i>neu erzeugt</i> statt neu gestartet? Ausgehendes HTTPS zu <code>licensing.resourcecloud.io</code> frei?
Assistent oder Suche antworten nicht	KI-Schalter eingeschaltet? Basis-URL vom Server aus erreichbar (<code>docker exec ... curl</code>)? Bei <code>AI_LOCAL_ONLY=1</code> wird eine Funktion mit externem Ziel nicht eingeschaltet.
Image lässt sich nicht laden: <code>denied</code> / <code>unauthorized</code>	<code>docker login dockerreg.jannedy.de</code> mit einem neuen Token wiederholen; das alte ist abgelaufen oder gesperrt, dann bei Ontaric melden. Die Rollout-Skripte brechen dann bewusst ab, statt ein altes Image weiterzubetreiben.
Einrichtungsassistent erscheint nicht	Die Datenbank enthält schon eine Firma und einen Benutzer (oder die Instanz wurde über <code>rc_seed_*</code> eingerichtet). Normal anmelden.
Port 8080 belegt (Weg A)	<code>HTTP_PORT</code> und Proxy-Ziel ändern, <code>up -d</code> erneut.

9 • Anhang

A • Wichtige Variablen der Umgebungsdatei

Variable	Bedeutung
<code>RC_IMAGE</code>	Anwendungs-Image, z. B. <code>dockerreg.jannedy.de/cptsolo/rescloud:5.3.0</code> ; immer eine Versionsnummer oder einen Digest eintragen. (Ohne Angabe verwendet die Compose-Datei den Docker Hub, auf den Sie keinen Zugriff haben.)
<code>DB_IMAGE</code>	Datenbank-Image (Standard <code>mariadb:12.3</code>)
<code>DB_NAME</code> , <code>DB_USER</code> , <code>DB_PASSWORD</code> , <code>DB_ROOT_PASSWORD</code>	Datenbankzugang; Pflicht
<code>BASE_URL</code> , <code>ACCESS_METHOD</code>	Öffentliche Adresse und Protokoll (<code>https</code>)
<code>HTTP_PORT</code>	Veröffentlichter Port bei Weg A
<code>TENANT</code> , <code>TENANT_HOST</code>	Instanzname und Hostname für die Weiterleitung durch Traefik; Pflicht, bei eigenem Proxy ohne Bedeutung

<code>SESSION_TIMEOUT</code>	Inaktivitätsgrenze in Sekunden (Standard 30000)
<code>MAIL_SENDER</code> , <code>ADMIN_MAILS</code>	Standard-Absender, Adressen für Administrator-Benachrichtigungen
<code>rc_license_customer</code> , <code>rc_license_token</code> , <code>rc_license_report_url</code>	Lizenzmeldung (5.3); Schreibweise in Kleinbuchstaben beibehalten
<code>RC_CRON_INTERVAL</code> , <code>RC_CRON_PING_URL</code> , <code>RC_CRON_JOB_TIMEOUT</code>	Takt (Standard 300 s), Totmannschalter, Zeitgrenze je Job (900 s)
<code>OLLAMA_EMBED_MODEL</code> , <code>AI_EMBED_BASE_URL</code> , <code>COMPOSE_PROFILES</code> , <code>AI_RERANK_BASE_URL</code> , <code>AI_LOCAL_ONLY</code>	Lokale KI (5.4)
<code>rc_seed_admin_*</code>	Vorgegebener erster Administrator (Abschnitt 4)

Eine Änderung der Umgebungsdatei wirkt erst, wenn der Container **neu erzeugt** wird (`up -d`), nicht durch `restart` .

B • Netzwerkverbindungen der Instanz

Von	Nach	Zweck	Notwendig?
Browser der Anwender	Ihr Proxy, 443	Zugriff auf die Anwendung	ja
Proxy	Server, Port <code>HTTP_PORT</code>	Weiterleitung (Weg A)	ja
Cron-Container	<code>licensing.resourcecloud.io</code> , 443	Lizenzmeldung: Kennung, Platzzahl	laut Vertrag
Web / Cron	Ihr S3-Endpunkt	Dateien	ja (empfohlen)
Web / Cron	Ihr SMTP-Server, 587/465	Mailversand	ja

Cron	Ihr IMAP-Server	Postfachabruf für Eingangs-E-Mail, falls genutzt	nein
Web / Cron	<code>login.microsoftonline.com</code> , <code>graph.microsoft.com</code>	Microsoft 365 (5.5)	nur bei Nutzung
Web	Ihr Identitätsanbieter	Single Sign-On (5.6)	nur bei Nutzung
Web / Cron	KI-Anbieter oder interner Modellserver	KI (5.4)	nur bei Nutzung
Server (Docker)	Docker-Registry bzw. Ihre Registry	Images beim Update	beim Update
Reranker (einmalig)	huggingface.co	Modelldatei, etwa 440 MB, danach nichts	nur bei lokaler KI, entfällt bei vorab kopierten Modellen
Ollama (einmalig)	ollama.com	Einbettungsmodell, danach nichts	nur bei lokaler KI, entfällt bei vorab kopierten Modellen
Traefik (nur Weg B)	Let's Encrypt	Zertifikate	nur Weg B

Die Datenbank und der Scheduler sind nur im privaten Netz der Instanz erreichbar und werden nicht veröffentlicht.

C · Weiterführende Anleitungen

- Kalender-Anbindung (Calendar Integration Guide) · Single Sign-On und Provisioning (SSO Guide) · Von Jira wechseln · Trust-Paket (technische und organisatorische Maßnahmen, Auftragsverarbeitung) · Betriebsrats-Paket
- Deployment-Referenz `docs/DEPLOYMENT.md` mit mehreren Instanzen auf einem Server, Zertifikaten und Betrieb

Ontaric GmbH · Dieser Entwurf beschreibt Version 5.3.0. Die Abläufe stützen sich auf die Installationsanleitung, die Deployment-Referenz und die Skripte der Version; die Wege B und C und

die Rückspielung einer Sicherung wurden für dieses Dokument nicht neu durchgespielt.