

Installing Resourcecloud

From nothing to a working instance with your company, your first user and outgoing mail — on Docker, on a blank Linux VM, or on Kubernetes.

Overview

Resourcecloud ships as one container image that contains the application (Apache + PHP 8.5). Around it you need a MariaDB database, a small scheduler container and an **S3-compatible object store** for uploaded files. Every installation target below starts the same three services; the object store is prepared first:

Service	What it does
<code>rescloud-mysql</code>	MariaDB 12.3 (11.8 is the minimum — the AI index needs <code>VECTOR</code> columns).
<code>rescloud-web</code>	Apache + PHP. Installs and migrates the database schema on start, then serves the application.
<code>rescloud-cron</code>	Same image, runs the scheduler only. Without it no mail is sent and nothing scheduled runs — and the instance still looks healthy.

All configuration comes from environment variables; nothing is edited inside the image. After the containers are up the steps are identical for every target: run the setup wizard, log in, connect the file storage, download the audit key and do the basic configuration.

Docker

Fastest. One machine, plain HTTP, ten minutes. Good for evaluation and small teams behind your own proxy.

Blank Linux VM

Debian/Ubuntu server with automatic HTTPS (Traefik + Let's Encrypt) and a real hostname.

Kubernetes

Manifest for an existing cluster with an Ingress controller.

What you need

	Minimum	Comfortable
CPU / RAM	2 vCPU / 4 GB	4 vCPU / 8 GB (more with local AI models)
Disk	20 GB	50 GB+ (database, uploads, images)
Software	Docker Engine 24+ with the Compose plugin (or a Kubernetes 1.25+ cluster)	
For HTTPS	A DNS name pointing at the server, and ports 80 and 443 reachable from the internet	
For files	An S3-compatible bucket and an access key for it (see below)	
For mail	An SMTP account the application may send through	

Before you start: prepare S3 storage

Everything users upload — task and project attachments, inline images, avatars — is stored through one storage layer. By default that is the local disk of the web container, and Resourcecloud itself warns that a **container loses every uploaded file when it is restarted or rescheduled**. For any Docker or Kubernetes installation, files therefore belong in an S3-compatible object store, and it is best to have that in place *before* the first upload, so nothing has to be moved later.

Prepare four things at your storage provider now; you will type them into Resourcecloud in step 6.1:

1. **A private bucket**, e.g. `rescloud-files`. Never make it public — downloads are handed out as short-lived signed links.
2. **An access key and secret** allowed to read, write and delete objects in that bucket. Give it that bucket only.
3. **The endpoint URL** (empty for Amazon S3) and the **region**.
4. Optionally a **key prefix**, so several instances can share one bucket.

The bucket will also hold the audit log's verification key and its daily anchor copies, under `audit/` (below your prefix). Do not add an expiry or lifecycle rule for that prefix, and keep it in your bucket backup or replication.

Provider	Endpoint	Region	Path-style
Amazon S3	<i>empty</i>	the bucket's region, e.g. <code>eu-central-1</code>	off
Hetzner Object Storage	<code>https://fsn1.your-objectstorage.com</code>	<code>fsn1</code> , <code>nbg1</code> or <code>hel1</code>	on
Wasabi	<code>https://s3.eu-central-1.wasabisys.com</code>	<code>eu-central-1</code>	on
Strato HiDrive	<code>https://s3.hidrive.strato.com</code>	as issued	on
MinIO, Ceph RGW (self-hosted)	<code>https://minio.example.com:9000</code>	the server's region (<code>us-east-1</code> is the usual default)	on

Backblaze B2 and any other S3-compatible service work the same way. The **region must match the endpoint**: it is part of the request signature, so a wrong one is rejected before the bucket is even looked at.

► No provider yet? A throwaway MinIO for testing

1 • Installing with Docker

This path runs the three services with Docker Compose and publishes the web container on a local port over plain HTTP. Put your own reverse proxy in front of it for HTTPS, or use the VM path, which does that for you.

1 Install Docker. Follow docs.docker.com/engine/install for your distribution, then check:

```
docker --version
docker compose version
```

- 2 **Get the two compose files.** They live in the root of the Resourcecloud repository. Copy them to a working directory of your own — nothing else from the repository is needed, the application is inside the image.

```
mkdir -p ~/rescloud && cd ~/rescloud
# from a checkout of the repository:
cp /path/to/rescloud/docker-compose.prod.yml /path/to/rescloud/docker-compose.local
```

- 3 **Create the environment file.** Generate two random passwords and write `rescloud.env`:

```
cat > rescloud.env <<EOF
RC_IMAGE=cptsolo/rescloud:latest
DB_NAME=rescloud
DB_USER=rescloud
DB_PASSWORD=$(openssl rand -hex 16)
DB_ROOT_PASSWORD=$(openssl rand -hex 16)

# Address people will type in the browser (no trailing slash)
BASE_URL=http://localhost:8080
HTTP_PORT=8080
ACCESS_METHOD=http

# Required by the compose file (used for proxy routing); any value works here
TENANT=rescloud
TENANT_HOST=localhost
EOF
chmod 600 rescloud.env
```

Back this file up. It holds the only copy of the database passwords. `BASE_URL` must be exactly the address you will use in the browser, otherwise links in e-mails and some redirects point to the wrong place. On a server, use its hostname or IP instead of `localhost`.

- 4 **Create the edge network** the compose file expects (once per machine; harmless if a proxy is not used):

```
docker network create rescloud-edge
```

5 Start it.

```
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
```

The compose file also defines an optional local embedding server (`ollama`) for AI search. The `--scale` flags leave it out, which saves a ~1 GB model download; see AI if you want it. Repeat the flags on every later `up` .

6 Wait until it is healthy. The first start creates the schema, which takes about a minute.

```
docker compose -p rescloud ps
curl -s http://localhost:8080/healthz.php
```

When `rescloud-web` shows `(healthy)` , `/healthz.php` answers `{"status":"ok"}` . Any other status names the failed check — see Troubleshooting.

7 Open the wizard: `http://localhost:8080/` redirects you to the setup wizard. Continue at First-run wizard.

2 • Installing on a blank Linux VM

For a real deployment on a fresh **Debian or Ubuntu** server. The bundled scripts install Docker, start a Traefik proxy that obtains and renews Let's Encrypt certificates, and deploy Resourcecloud behind it. Nothing is built on the server.

1 Point DNS at the server. Create an `A` record, for example `rescloud.example.com` → `203.0.113.10` , and make sure ports **80** and **443** are open in your provider's firewall. The certificate cannot be issued until the name resolves and port 80 is reachable.

- 2 **Copy the deployment files to the server.** From a checkout of the repository on your machine:

```
ssh root@rescloud.example.com 'mkdir -p /opt/rescloud-src'
scp -r deploy docker-compose.prod.yml docker-compose.localtest.yml \
    root@rescloud.example.com:/opt/rescloud-src/
```

The layout must stay intact: the scripts look for the compose file one directory above themselves.

- 3 **Prepare the server.** Log in and run the bootstrap script. It installs Docker, caps container log size, creates the shared `rescloud-edge` network and starts Traefik. It is safe to re-run.

```
ssh root@rescloud.example.com
cd /opt/rescloud-src/deploy
sudo bash bootstrap-server.sh --acme-email you@example.com --staging
```

`--staging` uses the Let's Encrypt *test* CA: browsers warn about the certificate, but a DNS mistake cannot exhaust the live CA's rate limit. Add `--firewall` to also restrict the machine to SSH, HTTP and HTTPS (off by default so a mistake cannot lock you out).

- 4 **Deploy the instance.** `new-customer.sh` takes an instance name and the hostname. It writes `/opt/rescloud/customers/<name>.env` with random database passwords, starts the containers and waits until they are genuinely healthy.

```
cd /opt/rescloud-src/deploy
./new-customer.sh main rescloud.example.com --no-ai --start
```

- Without `--admin-email` / `--admin-name` / `--company` the instance shows the manual setup wizard — what you want for your own installation. With all three it configures itself and prints an admin password once.
- `--no-ai` skips the local AI services. Leave it out to also start an embedding server and reranker (see AI).
- The instance name must match `[a-z][a-z0-9_]*`; it is also the compose project name.

5 Check it.

```
docker compose -p main ps
curl -sk https://rescloud.example.com/healthz.php
```

Open `https://rescloud.example.com/`. With `--staging` the browser warns about the certificate — accept it for now.

6 Switch to real certificates once the page loads:

```
sudo bash bootstrap-server.sh --acme-email you@example.com # no --staging
docker compose -p traefik up -d --force-recreate
```

Traefik then requests a trusted certificate on the next request and renews it automatically.

7 Back up `/opt/rescloud/customers/*.env` and the Docker volumes — see Operating it. Then continue at First-run wizard.

The complete operations reference — several customers on one server, rolling updates, licensing dashboard, backups — is `docs/DEPLOYMENT.md` in the repository.

3 • Installing on Kubernetes

A ready-made manifest, `install-guide/rescloud-k8s.yaml`, creates a `rescloud` namespace with the database (StatefulSet), the web and scheduler Deployments, their volumes, Services and an Ingress. It mirrors the Docker Compose setup one to one.

Prerequisites

- An S3 bucket and key, prepared as described above — pods are ephemeral, so this matters even more here than on a single Docker host.
- A cluster (1.25+) with a default `StorageClass` (or set `storageClassName` in the manifest).
- An Ingress controller (the manifest assumes *ingress-nginx*) and, for HTTPS, *cert-manager* with a ClusterIssuer. Neither is required for a first test — see the port-forward tip below.
- `kubectl` pointing at that cluster, and a DNS name for the Ingress.

- 1 **Copy the manifest and edit it.** Every line that needs your attention is marked `# <--` CHANGE :

Object	Setting	Set to
Secret <code>rescloud-db</code>	<code>DB_PASSWORD</code> , <code>DB_ROOT_PASSWORD</code>	Long random values (<code>openssl rand -hex 16</code>)
ConfigMap <code>rescloud-config</code>	<code>rc_baseurl</code>	The public URL, e.g. <code>https://rescloud.example.com</code>
ConfigMap	<code>rc_mailsender</code>	Sender address for outgoing mail
Deployments	<code>image</code>	Pin a version tag (same on web and cron), or your own registry
Ingress	host, TLS host, class, issuer	Your hostname, ingress class and cert-manager issuer

- 2 **Apply it.**

```
kubectl apply -f rescloud-k8s.yaml
kubectl -n rescloud get pods -w
```

The database becomes `Ready` first; `rescloud-web` then installs the schema (the startup probe allows up to five minutes) and becomes `Ready` when `/healthz.php` answers 200. `rescloud-cron` starts on the same node as the web pod, because both share the uploads volume.

- 3 **Follow the logs if anything is slow:**

```
kubectl -n rescloud logs deploy/rescloud-web -f
```

4

Reach it. Once DNS and the certificate are in place, open

`https://rescloud.example.com/` . To test before that, forward a port and set `rc_baseurl` / `rc_access_method` to `http://localhost:8080` / `http:`

```
kubectl -n rescloud port-forward svc/rescloud-web 8080:80
```

Continue at First-run wizard.

Design notes

- **One web replica, Recreate strategy.** Uploads and sessions live on volumes. Running several web replicas needs a `ReadWriteMany` storage class for `rescloud-userdata` and `rescloud-sessions` ; migrations are serialised by a database lock, so several pods starting together is safe.
- **Exactly one** `rescloud-cron` . Never scale it up.
- **Secrets.** Database passwords are in a `Secret` ; for anything beyond a test, source them from your secret manager instead of committing the manifest.
- **Upgrades.** Change the image tag on both Deployments and apply. The new web pod migrates the schema before it reports Ready.
- **Mail via environment.** Instead of the settings page you can add `rc_smtp_host` , `rc_smtp_port` , `rc_smtp_user` , `rc_smtp_pass` , `rc_smtp_secu` and `rc_smtp_from` to the ConfigMap/Secret.

4 • First-run wizard

On a database that holds no company and no user, the application starts the setup wizard at `/basic_setup` . It runs exactly once: as soon as a team and a user exist, the same address forwards to the application. It has three tabs.

Tab 1 — The first user

This person becomes the first **administrator** and team lead. Enter first name, last name, a password (twice) and an e-mail address; a picture is optional — skip it for now, so no file is written before the storage is connected.



Configure Your Installation

This information is needed to basically set up your ResourceCloud installation.

USER

COMPANY

FINISH

Please insert the first user



CHOOSE PICTURE



First Name (required)



Last Name (required)



Password (required)



Password again (required)



Email (required)

NEXT

The wizard opens on the **User** tab.

Choose a strong password — this account can configure everything. The login name is derived for you: first letter of the first name plus the family name in lower case (*Anna Schmidt* → `aschmidt`).

Tab 2 — Your company

RESOURCECLOUD
DevOps Organisation

Configure Your Installation

This information is needed to basically set up your ResourceCloud installation.

USER **COMPANY** FINISH

Please insert your company data

Company Name*	Shortcode*	
Example Engineering GmbH	EXENG	
Street Name*	Street Number*	
Hauptstrasse	12	
Zip Code*	City*	Country*
10115	Berlin	Germany

PREVIOUS NEXT

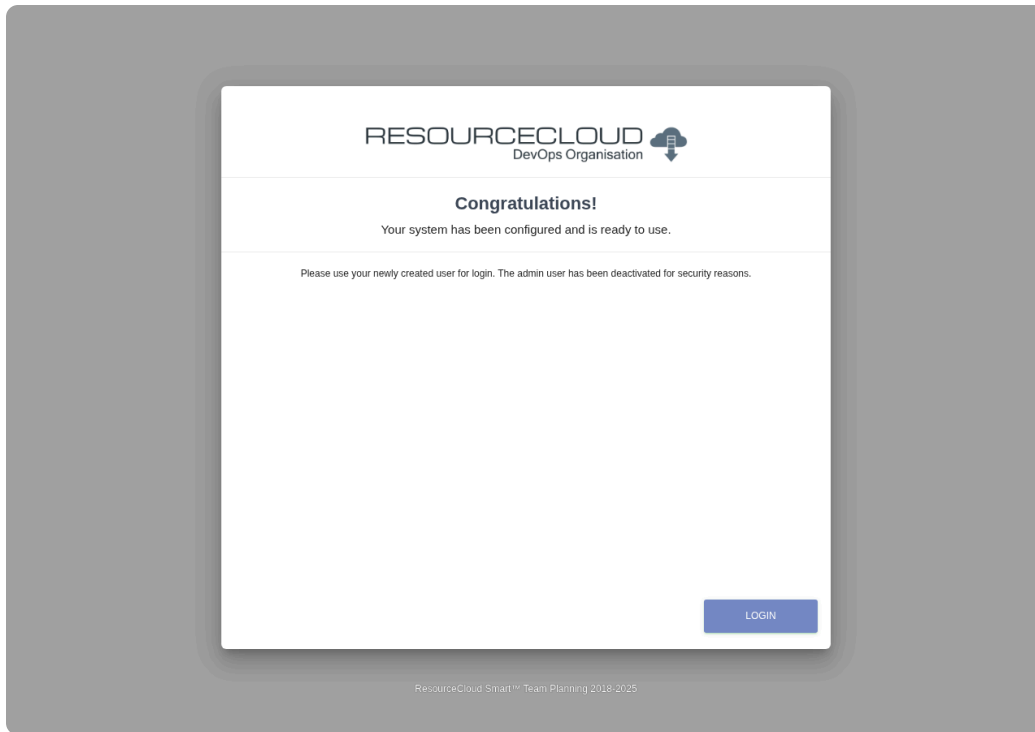
ResourceCloud Smart™ Team Planning 2018-2025

Company name, a shortcode of at most six letters (used to name the default area, department and team), address and country.

From this the wizard builds the starting organisation: your company, a default area, department and team (`<SHORT>_area` , `<SHORT>_dep` , `<SHORT>_team`), with the first user as team leader, plus a welcome task. Everything can be renamed and extended later in *Org Management*.

Tab 3 — Finish

Click **Next**, read the note, then **Finish**. The wizard writes the configuration and shows a confirmation:



The built-in `admin` account is deactivated for security; use the user you just created.

Skipping the wizard. Automated deployments can pre-seed the first administrator and company through `rc_seed_*` environment variables (this is what `new-customer.sh --admin-email ... --company ...` does). The wizard is then never shown.

5 • First login

Click **Login** and sign in with the login name from the wizard (the field is labelled “Email” but takes the login name) and the password you chose.

Resources & Inspiration

Smart Team Organisation

Do more with less people. Reliable commitments and happy colleagues!

Smart teamplanning, cool project planing, virtual teams, resource planning, vacation planning and much more...

"Look mum: Scrum, Kanban, Waterfall... all in one place!!"

More information?

[Find out more](#)

Sign in as **aschmidt** (or whatever your first user's login name is).

You land on **My Tasks** with one welcome task in your team. The navigation on the left is your map: everything you need for setup is under **Admin**, near the bottom in the *Settings* group.

RESOURCECLOUD
DevOps Organisation

PROJECTS:
Recent projects

Load: 0.4 % | Role: Teamlead | Week: 39

Find Tasks and more

AS

WORK

Tasks

My Tasks

Taskpool

Backlog

Team Overview

Kanban View

Scheduled Tasks

Worktime

Documents

Extra

SERVICE

Incidents

Service Requests

My Requests

Customer Requests

DELIVER

Projects

Systems

INSIGHT

Dashboards

Watchlist

Search

Messages

SETTINGS

Home / My Task Overview

Select a solution!

R&S

Time

Description

Send

My Tasks

Everything assigned to you, most urgent first

Mon 21 Sep 2026 · KW 39

OVERDUE

0

0:00 h of work left

DUE TODAY

0

0 meetings

THIS WEEK

0

0:00 h planned

WEEK LOAD

0%

0:10 of 40:00 h booked

UNREAD COMMENTS

1

on 1 task

Search my tasks...

All 1

Overdue 0

Replan 0

Today 0

This week 0

Problems 0

Later 1

LATER 1

Next weeks and unplanned

Relevant

All

Σ 0:10 / 1:00 h

COMMENTS

TASK

PLAN

EFFORT

VALUE

FLAGS

1 new

Welcome To Resourcecloud™

RC-1 Example Engineering GmbH opened 0 d ago

Common Tasks

R&S

EXENG_dep

No week

UNPLANNED

0:10 / 1:00

4

☆

Edit

Done

...

ResourceCloud 5.1.0 - Smart Team Planning © BJ 2014-2025 - Get support here... Last account activity 7 sec ago

First view after login. The Admin menu at the bottom of the navigation holds Org Management and Global Settings.

6 · Basic configuration

A fresh installation runs, but five things are worth doing before you invite anyone — and the first one before anybody uploads a file. Right after it, download the audit key. They all need the administrator role.

6.1 File storage (S3) — do this first

Open **Admin** → **Global Settings** → **Data Management** and find the *External Storage* card. Until it is switched on, uploads go to the container's local disk.

External Storage

Where uploaded files are stored

By default, all attachments are written to the local filesystem of this server (`userdata/attachments`). That is fine for a classic installation, but a **container loses every uploaded file when it is restarted**.

Switch this on to store all attachments in an external, S3-compatible object store instead. Any provider works — Amazon S3, MinIO, Ceph, Wasabi, Backblaze B2, Hetzner, Strato, or your own self-hosted store. Files already on the local disk stay readable, and can be copied into the bucket with the migration button below.

Use external S3 storage



Endpoint URL

Leave **empty** for Amazon S3 — the region determines the endpoint there.

MinIO: `https://minio.example.com:9000` | Wasabi: `https://s3.eu-central-1.wasabisys.com`

Hetzner: `https://fsn1.your-objectstorage.com` | Strato: `https://s3.hidrive.strato.com`

Bucket

Region

Must match the endpoint. The region is part of the request signature, so a wrong one is rejected before the bucket is ever looked at.

Hetzner: `fsn1`, `nbg1` or `hel1` — the same prefix as the endpoint host.
AWS: the bucket's own region. MinIO / Ceph: any value, as long as it matches what the server was configured with (`us-east-1` is the usual default, and is what Resourcecloud sends when this is left empty).

Access Key

Secret Key

Key Prefix (optional)

All objects are stored below this prefix, so several instances can share one bucket.

Path-style addressing



On (`host/bucket/key`) for MinIO, Ceph and most self-hosted stores. Switch off for Amazon S3, which uses `bucket.host/key`.

Create the bucket if it is missing



Applies when you press *Test Connection*. You can also leave this off — if the bucket turns out to be missing, the test offers to create it.

Connected to bucket 'rescloud-files' - write, read and delete all succeeded.

Move existing files into the bucket

There are **0** file(s) on the local disk of this server. Copying them into the bucket makes them survive the next container restart. The local copies are kept, so this can be repeated safely.

Hint: Save the settings before migrating. Downloads always check the local disk first, so older attachments keep working after the switch.

Global Settings → Data Management → External Storage, after a successful *Test Connection*.

1. Switch on **Use external S3 storage**. The fields appear.

2. Enter **Endpoint URL** (leave empty for Amazon S3), **Bucket**, **Region**, **Access Key** and **Secret Key** — the values you prepared above. Optionally set a **Key Prefix**.
3. Set **Path-style addressing** on for MinIO, Ceph and most self-hosted or European providers; off for Amazon S3.
4. Press **Test Connection**. It writes, reads and deletes a probe object and reports “*Connected to bucket ... write, read and delete all succeeded.*” If the bucket does not exist, the test offers to create it (or tick *Create the bucket if it is missing* beforehand).
5. Press **Save Storage Settings**. The secret is stored in the database and never shown again; leave the field blank on later saves to keep it.

From now on every attachment and avatar goes to the bucket. Downloads check the local disk first, so files uploaded earlier stay reachable.

If files were uploaded before you switched storage on, the card shows how many, with a **Copy local files to S3** button that copies them across (safe to repeat; local copies are kept). A half-filled configuration silently falls back to local storage rather than losing uploads — so always run the connection test and check that the card reports success.

To verify end to end, attach a file to the welcome task and look in the bucket: a new object named `<hash>/<filename>` appears (below your prefix, if you set one).

Then: download the audit key

Resourcecloud keeps an audit log of who changed what. Nobody can edit it, and a hash chain protected by a secret key makes any later change detectable. Resourcecloud creates that key itself and keeps it in the bucket you just connected — but **you need a copy of your own**. Without it the log cannot be verified after a restore, a move to another bucket or a replaced key. As RC Admin, you are responsible for that copy.

- 1 **Open Admin → Audit Log**. A yellow notice says *Download the audit key and keep it safe*.
- 2 **Press Download key** and store the file where you keep important secrets — a password manager or a vault, not only on the server and not in the database backup.
- 3 **Press “I have stored it.”** The notice disappears and the page shows who confirmed the copy. Only an RC Admin can confirm; the Auditor role may download too.

Do this now, while nothing depends on it. If you set `RC_AUDIT_KEY` yourself in the deployment, no notice appears — keep that value with your other secrets instead. More: *Resourcecloud audit log guide* ([docs/auditlog/index.en.html](https://docs.resourcecloud.io/docs/auditlog/index.en.html) , also a PDF).

6.2 Global settings

Admin → Global Settings (`/rc_settings`) has three tabs: *Configuration*, *External Services* (mail, AI, video, proxy) and *Data Management* (file storage, see 6.1). The Configuration tab controls behaviour for the whole installation:

The screenshot shows the ResourceCloud web interface. The top navigation bar includes the ResourceCloud logo, a 'PROJECTS' dropdown, a 'Recent projects' list, a 'Load' indicator (0.4%), a 'Role' indicator (Teamlead), a 'Week' indicator (39), a search bar, and user avatars. The left sidebar contains a menu with categories: WORK (Tasks, Worktime, Documents, Extra), SERVICE (Incidents, Service Requests, My Requests, Customer Requests), DELIVER (Projects, Systems), INSIGHT (Dashboards, Watchlist, Search, Messages), and SETTINGS (Work Orga, Admin, Org Management, Task Management, Global Settings). The main content area is titled 'Settings' and 'Global Settings for ResourceCloud'. It features three tabs: 'Configuration' (selected), 'External Services', and 'Data Management'. The 'Configuration' tab is divided into four sections: 'User Tracking' (with 'Enable User Reports' and 'Enable User Load Tracking' toggles), 'Free Edit' (with 'Everyone can edit tasks' toggle), 'Max Prio Tasks per Team' (with a numeric input for 'Number of allowed prio tasks per team' set to 4), and 'Capacity Forecast' (with 'External contractor rate for the Project → Capacity Forecast tool' input set to 0). A yellow attention box is present in the 'User Tracking' section.

Global Settings → Configuration.

Setting	What it does / recommendation
Enable User Reports	Lets team leads with the "reports" role generate performance reports. Off by default — in many countries this needs works-council approval, so enable it deliberately.
Enable User Load Tracking	Team leads see each worker's percentual workload.
Free Edit / Open Restrictions	Whether people may edit or create tasks outside their own team. Leave the defaults unless you have a reason.
Max Prio Tasks per Team	Cap on tasks flagged as priority per team (default 4). Keeps "everything is urgent" in check.
Default Support Team	The team that receives tickets from new customers. Pick your team here; it defaults to the one the wizard created.
Capacity Forecast	Hourly rate used when a project's capacity gap is sized in external contractors.
Meeting Times	Daily window in which meetings can be planned, and an optional blocked noon break.
Global Event Types	Calendar event types offered to everybody (absence, on-call, ...). Add your own.

Press **Submit** in each card after changing it.

6.3 Mail server (SMTP)

Resourcecloud sends notifications, ticket replies and password e-mails. The scheduler container drains the mail queue, so the mail setup needs both a working SMTP account *and* a running `rescloud-cron`.

Open **Global Settings** → **External Services** and find *Mailserver Configuration*:

Mailserver Configuration

Global SMTP Configuration

If you don't fill in anything here, Resourcecloud will use the default SaaS SMTP server to send notification emails. You need to provide this information in order to send mails from within Resourcecloud when you have an On-Prem installation. Make sure you use a functional account, because this account is used by all outgoing mail traffic!

SMTP Server

e.g. smtp.gmail.com

Port

e.g. 587

Username

e.g. fu.bar@gmail.com

Password

Your smtp password here...

From Address

Your sender address here...

Brand Name

Resourcecloud

Use TLS / SSL



Use Authentication



Save the settings first - the test mail is sent with the stored configuration.

Send Test Mail

Submit

Hint: Many SMTP Servers require Username and From Address to be the same!
It may take one or two minutes for the test mail to arrive in your inbox.

Global Settings → External Services → Mailserver Configuration.

1. Enter **SMTP Server** and **Port** (587 with STARTTLS is typical, 465 for implicit TLS).
2. Enter **Username** and **Password**, and switch on **Use Authentication** and **Use TLS / SSL** as your provider requires.
3. **From Address** — many providers insist that it equals the username or belongs to the authenticated domain.
4. **Brand Name** is shown in outgoing mail.
5. Press **Submit**, then **Send Test Mail**. It can take a scheduler cycle (up to five minutes by default) to arrive.

If you leave this blank the instance falls back to the SaaS default server — not what you want on-premises, where nothing would be delivered. Automated setups can supply the same values as `rc_smtp_*` environment variables.

6.4 Teams, roles and users

Admin → Org Management (`/org_management`) is where the organisation lives: users, virtual teams, teams, departments, areas, your company, access roles and mail-in addresses.

The screenshot displays the 'Organisation Management' page in the ResourceCloud application. The left sidebar contains a navigation menu with sections: WORK (Tasks, Worktime, Documents, Extra), SERVICE (Incidents, Service Requests, My Requests, Customer Requests), DELIVER (Projects, Systems), INSIGHT (Dashboards, Watchlist, Search, Messages), and SETTINGS (Work Orga, Admin, Org Management, Task Management, Global Settings). The main content area is titled 'Organisation Management' and includes a sub-header 'Here you can change worker, teams, departments, etc.'. Below this is a tabbed interface with 'Users' selected. The 'All Users' section features an 'Add User' button, a 'Show 10 entries' dropdown, and a search bar. A table lists user details with columns: ID, Username, Team, Company, Name, Primary Role, Notifications, h/W, Last Login, Login Count, State, and Edit. The table contains one entry for 'Anna Schmidt' (Teamlead, Active). At the bottom, there is a 'Bulk Import' button and a footer with version information and support links.

ID	Username	Team	Company	Name	Primary Role	Notifications	h/W	Last Login	Login Count	State	Edit
1	aschmidt	EXENG_team	Example Engineering GmbH	Anna Schmidt	Teamlead	On	40	21.09.2026 15:17	1	Active	Edit

The Users tab. After the wizard it lists just the first user.

The structure is a ladder from the company down to the team, with a tab for every level (*C-Levels, Areas, Departments, Teams*). Look at the **Teams** tab to see what the wizard created:

RESOURCECLOUD
DevOps Organisation

PROJECTS:
Recent projects

Load: 0.4 % | Role: Teamlead | Week: 39

Find Tasks and more

AS

WORK

- Tasks
- Worktime
- Documents
- Extra

SERVICE

- Incidents
- Service Requests
- My Requests
- Customer Requests

DELIVER

- Projects
- Systems

INSIGHT

- Dashboards
- Watchlist
- Search
- Messages

SETTINGS

- Work Orga
- Admin
 - Org Management
 - Task Management
 - Global Settings

Organisation Management

Here you can change worker, teams, departments, etc.

UsersVirtual TeamsTeamsDepartmentsAreasC-LevelsCompanyAccess RolesMail In

All Teams

Add new Team

The CEO Office reports to the company, not to an Area, and is managed on the Company tab.

No.	Company	Area	Department	Short Name	Team	Team Leader	Rate	Bearbeiten
4	Example Engineering GmbH	EXENG_area	EXENG_dep	EXENG_team	Default Team EXENG_team	Anna Schmidt	100 €	Edit

ResourceCloud 5.1.0 - Smart Team Planning © BJ 2014-2025 - Get support here...Last account activity 6 sec ago

One default team, led by the first user. Use **Add new Team** for more, and the Departments / Areas tabs to rename or add levels.

To add people, use **Users → Add User**:

Name, e-mail, team, role and weekly work hours. Tick *Create another* to enter several people in a row.

- Resourcecloud generates a **one-time password** and shows it to you after submitting. The new user is forced to choose their own on first login. Hand it over through a safe channel.
- **Role** decides what the person may do (Worker, Teamlead, and more). The *Access Roles* tab lists them all with descriptions.
- **Workhours** per week feed the capacity and load planning — set them realistically (part-timers!).
- Many users at once: use **Bulk Import** at the bottom of the Users tab.

6.5 AI features (optional)

Two features use an AI service: *semantic task-duration estimation / document search* and the *AI chat assistant*. Both are off until you enable them in **Global Settings → External Services → AI Service Configuration**.

AI Service Configuration

Help

Connects ResourceCloud to an OpenAI-compatible AI service for task estimation, the chat assistant and document search. Without one, estimation falls back to keyword matching. Provider examples and details are under [Help](#).

Enable AI-based Estimation

☐

Enable AI Chat Assistant

☐

API Key

API Base URL

Used for chat and, unless overridden below, for embeddings.

Embeddings Base URL (optional)

Only when embeddings run on a different service than chat (e.g. chat on OpenRouter).

Embeddings API Key (optional)

A local Ollama needs no key, but enter any value (e.g. `ollama`).

Embedding Model

Changing the model or embeddings endpoint re-embeds all 0 passages (about 1 minute without a GPU).

Chat Model

Proxy (optional)

Leave empty for no proxy (or the global proxy, if configured).

Reranker Base URL (optional)

Improves the order of retrieved passages before the assistant answers. Leave empty for no reranking.

Reranker API Key (optional)

Reranker Model (optional)

A local llama.cpp reranker ignores this.

Local-only mode

☐

When on, every AI endpoint must be on your own network; others are refused on save.

Document Search — “Ask your Documents”

Lets people ask questions about uploaded documents; answers only use files they can access. **Document content is sent to the AI endpoint configured above.**

Enable Document Search

☐

Index documents from the last

days

☒ indexes everything.

Documents per run

The indexer runs every five minutes.

Index status: 0 searchable, 0 waiting, 0 without readable text, 0 failed — 0 passages stored.

Re-index everything discards every stored passage and costs a full run against the AI service.

Meeting Protocols, Agendas and Blog Entries

Lets the assistant answer from meeting protocols, agendas and blog entries, with the usual visibility rules. **The text is sent to the AI endpoint configured above.** Switching this off deletes what has been indexed.

Enable Protocol, Agenda and Blog Search

☒

Record index: 0 meeting protocols, 0 meeting agendas, 0 blog entries — 0 passages stored.

Save AI Settings

AI Service Configuration. With nothing configured, a keyword-based fallback is used.

- Any OpenAI-compatible endpoint works: OpenAI, Gemini, a local Ollama, or Anthropic for chat. Enter API key, base URL, embedding model and chat model, switch the features on and **Save AI Settings**.
- Anthropic offers no embeddings — use OpenAI, Gemini or the local embedding server for that part.
- The compose file's `ollama` service (and the optional `reranker`) provide embeddings on your own hardware, so no document text leaves the server. Start them by omitting the `--scale ollama=0 --scale ollama-init=0` flags; the first start downloads the model (roughly 0.3–1.7 GB depending on the model). Then use `http://ollama:11434/v1` as base URL.

You are done with the basics. Create your teams and users, then start with the welcome task in *My Tasks*.

7 · Operating it

Back up

What	Docker / VM	Kubernetes
Database	<code>docker exec <project>-rescloud-mysql-1 mariadb-dump -uroot -p"\$DB_ROOT_PASSWORD" rescloud > backup.sql</code>	<code>kubectl -n rescloud exec sts/rescloud-mysql -- sh -c 'mariadb-dump -uroot -p"\$MARIADB_ROOT_PASSWORD" rescloud' > backup.sql</code>
Uploads & avatars	The S3 bucket — use your provider's versioning or replication. The <code>userdata</code> volume/PVC then only holds a local cache and anything written before S3 was switched on.	
Secrets	The <code>.env</code> file	The <code>rescloud-db</code> Secret
Audit key	Download it once (6.1) and keep it with your secrets. It also lives in the S3 bucket under <code>audit/</code> , so back up database and bucket together.	

Use `mariadb-dump` — MariaDB 12 no longer ships the `mysqldump` name. Sessions need no backup (losing them logs everyone out once).

Update

```
# Docker (same command as install, after pulling)
docker compose -f docker-compose.prod.yml -f docker-compose.localtest.yml \
  --env-file rescloud.env -p rescloud pull
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

# Kubernetes
kubectl -n rescloud set image deploy/rescloud-web web=cptsolo/rescloud:<new-tag>
kubectl -n rescloud set image deploy/rescloud-cron cron=cptsolo/rescloud:<new-tag>
```

The web container applies pending database migrations on start and reports unhealthy until the schema matches. Take a database backup first. On a VM, `deploy/rollout.sh` updates with near-zero downtime.

Where to look

- Logs: `docker compose -p rescloud logs -f rescloud-web` · `kubectl -n rescloud logs deploy/rescloud-web`
- Health: `/healthz.php` (200 only when config, database and schema are all fine)
- Scheduler heartbeat: if the scheduler stops for 15 minutes, every administrator sees a red banner in the application.

8 · Troubleshooting

Symptom	Cause and fix
<code>/healthz.php</code> → 503 <code>db_unreachable</code>	Database not up yet, or wrong <code>DB_*</code> values. Check <code>docker compose ps</code> / the mysql pod. Changed a password after first start? The database keeps the old one — either change it inside MariaDB or recreate the volume (destroys data).
503 <code>schema_outdated</code> / <code>schema_missing</code>	Migration still running or failed — read the web container log. Usually resolves itself within a minute on first start.
Web container exits at once	A required variable is missing; the log names it (<code>DB_PASSWORD</code> , <code>TENANT</code> , ...). For compose, the message reads "set ...".
Redirects go to the wrong address, links in mails are wrong	<code>BASE_URL</code> / <code>rc_baseurl</code> and <code>ACCESS_METHOD</code> must match the URL in the browser (http vs https, host, port).
No mails arrive	Check in order: <code>rescloud-cron</code> running, SMTP settings and <i>Send Test Mail</i> , the log of the cron container, and the spam folder.
Wizard does not appear; you land on a login page	The database already holds a company and a user (or the instance was seeded through <code>rc_seed_*</code>). Log in normally.
Browser certificate warning on a VM install	Expected while on <code>--staging</code> . If it persists after switching, check DNS and that port 80 is reachable, then <code>docker compose -p traefik logs traefik</code> .
Port 8080 already in use	Change <code>HTTP_PORT</code> and <code>BASE_URL</code> in the env file and run <code>up -d</code> again.

Screenshots were taken from a real installation of version 5.1.0 following the Docker steps above, with MinIO as the object store. More detail: `docs/DEPLOYMENT.md` and `docker/README.md` in the repository.