EKE Services On-Prem: Installation Guide

Release 2026-11-24 · For the administrator who installs EKE on the customer's own network

This guide installs the EKE Services suite (Data Migration, File Migration, Archive Files) on your own machines with `eke-setup`, the installer that ships in every release bundle. You need administrator rights on each machine and the license key (`lic_…`) EKE sent you.

1. What gets installed

- **One site server.** It runs manage-tenants (the console, users and the license), auth-cred-service (the stored Salesforce connections), manage-auth-creds (the page where you add those connections) and the three product controllers. It can also run PostgreSQL for you.
- **Worker machines** (as many as your license allows per product). They do the transfer work. Each joins the site with a one-time code and gets its program and settings from the site server, so worker machines need no internet access and no EKE download.
- Any mix of **macOS** (Apple silicon), **Linux** (x86_64 with systemd) and **Windows** (x86_64, 10/11 or Server 2019 and later) works for both roles.

2. Before you start

Need	Details
Release	One for the site server's OS: `eke-onprem-<date>-mac.pkg`, `eke-onprem-<date>-windows.msi`, or the bundle `eke-onprem-<date>-<mac|linux|windows>.tar.gz`. The installers just unpack the bundle. It contains the worker programs for all three OSes.
License key	`lic_…`, sent by EKE separately from any download link. You enter it in the console after installing.
PostgreSQL	Either the one in the bundle (`bundled_postgres = true`: simplest; for evaluations and small sites) or your own PostgreSQL 14 or later, recommended for production so your team patches and backs it up. `eke-setup` needs a superuser (or a role with CREATEROLE and CREATEDB) once, to create its databases.
Network	The site server needs a fixed LAN address (reserve it in DHCP). Browsers and workers reach it on TCP 3000, 3002, 3005, 3006, 3012 and 3020, and the site reaches workers on 3051 (Data), 3021 (Archive) and 3001 (File). File Migration workers also reach the database (bundled: port 5433).
Internet	The site server needs outbound HTTPS to `manage-tenants.ekeservices.com` to renew a subscription license (it keeps working for up to 7 days without it). All-In sites need it only until their license file is installed.
Windows	With the bundled PostgreSQL: the Microsoft Visual C++ 2015-2022 Redistributable (x64), which most Windows machines already have.

3. Unpack and check the release

Unpacking installs no service yet. Pick the way that fits the site server:

OS	How	Where the bundle ends up
macOS	Open `eke-onprem-<date>-mac.pkg` (signed and notarized by Apple; works without internet)	`/usr/local/eke-releases/eke-onprem-<date>-mac`
Windows	Run `eke-onprem-<date>-windows.msi` (signed by EKE Services LLC); silently: `msiexec /i eke-onprem-<date>-windows.msi /qn`	`C:\Program Files\EKE Releases\eke-onprem-<date>-windows`
Any	`tar -xzf eke-onprem-<date>-<os>.tar.gz` (Windows 10+: `tar` works in PowerShell too)	where you unpack it

Then, from that folder:

```
cd /usr/local/eke-releases/eke-onprem-2026-11-24-mac    # or your folder
bin/eke-setup verify                                    # EKE's signature, then every file
```

A Mac without internet, using the `.tar.gz`: macOS checks programs downloaded in a browser with Apple the first time they run, and without internet it may block them. Use the `.pkg`, or check the download with minisign (below) and then clear the download flag: `xattr -dr com.apple.quarantine eke-onprem-2026-11-24-mac`.

`verify` must report `signed by EKE release key 328E8BEA42539E79` and every file as checked. If it reports a changed or missing file, or a missing or invalid signature, don't install the bundle: download it again and tell EKE. `server` and `upgrade` run the same check first.

You can also check the download itself before unpacking it, with the standard minisign tool and EKE's release public key, which is also on ekeservices.com/release-signing.html and in Appendix A of your license agreement:

```
minisign -Vm eke-onprem-2026-11-24-linux.tar.gz -P
        RWR5nlNC6ouOMlPnJTCsc9yPAOyH0FqoPwuasK7IkEIHiG4m7kBOMJz7
```

On Windows, every EKE program is also signed by "EKE Services LLC" (Authenticode); right-click → Properties → Digital Signatures shows it.

4. Write the answers file

Create `site.toml` next to the bundle (example; only the first two lines are required):

```
site_address = "192.168.1.10"        # the site server's fixed LAN address
admin_email  = "it-admin@example.com" # the first console administrator
license_mode = "subscription"         # or "all_in", as on your license
bundled_postgres = true               # or remove it and use your own server:
# postgres_admin_url = "postgresql://postgres@db.example.local:5432/postgres"
worker_slots = 20                     # the most workers per product that can ever join

# Optional, for password-reset email:
# [smtp]
# host = "smtp.example.com"
# username = "eke@example.com"
# password = "…"
# from_address = "eke@example.com"
```

With your own PostgreSQL, keep its password out of the file: set `EKE_POSTGRES_ADMIN_URL` in the shell that runs the install (`sudo --preserve-env=EKE_POSTGRES_ADMIN_URL …` on macOS/Linux).

5. Install the site server

Run on the site server, from the unpacked bundle (section 3):

OS	Command
Linux	`sudo bin/eke-setup server --answers ../site.toml`
macOS	`sudo bin/eke-setup server --answers ../site.toml`
Windows	In an **Administrator** PowerShell: `bin\eke-setup.exe server --answers ..\site.toml`

It creates a service account (`eke` on Linux, a hidden `_eke` on macOS, one virtual account per service on Windows), generates the site's keys once, creates the databases (each with its own login), writes each service's settings (readable only by its account), installs everything as system services that start at boot, starts them in order and checks each one. At the end it prints:

```
Master admin: it-admin@example.com / <a one-time password>  (shown once; it must be changed at the first sign-in)
Console: http://192.168.1.10:3012
```

Write the password down now; it is not shown again. Running `eke-setup server` again later is safe: it keeps the keys and the data and only refreshes the installation.

6. Sign in and enter the license key

1. Open `http://<site_address>:3012`, sign in with the admin email and the one-time password, and choose a new password.
2. In the **Site License** card, paste the `lic_…` key and save. The card shows the license state, products, limits and expiry within a few seconds.
3. Create your tenant (the tenant ID on your license) and invite its users. Without email set up, the console shows each new user's one-time password until you click **Done**: copy it and give it to the user yourself. Each user must sign in to this console (port 3012) first and choose a new password; until then the other pages refuse their sign-in and say so.

Until the key is entered, only the master admin can sign in and no worker is accepted.

Add your Salesforce connections

Each tenant adds its orgs at `http://<site_address>:3002` (the installer prints this address too), signed in as a tenant user with the Member role or higher. Add one connection per org (production or sandbox), with the Connected App details your Salesforce administrator set up. Secrets are stored encrypted and are never shown again; to change one, enter it again. The products list these connections when you start a job.

Map users before migrating data

Data Migration writes each record's owner and its Created By / Last Modified By users only when it can find the same user in the target org. It matches users by the field `Legacy_SF_Id__c` on **User** in the target org, holding the source user's Id. Without that field (or with users not filled in), every migrated record is owned by

the connection's integration user, and the job report lists those fields as `unmapped_user`. Ask your Salesforce administrator to create the field on User in the target org and fill it in before the first real migration. Created Date and Last Modified Date are set by Salesforce on insert and can't be copied.

7. Firewall

- **Windows:** `eke-setup` adds the inbound rule "EKE site server" (Domain and Private networks).
- **macOS:** if the firewall is on, macOS asks once per program to accept incoming connections; allow them.
- **Linux:** `eke-setup` prints the commands; it does not change the firewall. With `ufw` (on by default on some distributions):

```
sudo ufw allow from 192.168.0.0/16 to any port 3000,3002,3005,3006,3012,3020,5433 proto tcp
```

Use your LAN's range. Leave 5433 out when you use your own PostgreSQL. The installer prints these commands every time, even when the ports are already open; if you opened them beforehand, there is nothing more to do.

8. Add worker machines

On the **site server**, create a code for each worker (single use, valid 24 hours):

```
sudo /opt/eke/bin/eke-setup join-code --product data      # or: file, archive
```

(The installed `eke-setup` is in `/opt/eke/bin` on Linux, `/usr/local/eke/bin` on macOS and `C:\Program Files\EKE\bin` on Windows.)

On the **worker machine**, copy `eke-setup` from the bundle's `workers/<os>/` folder (or `bin/` for the site server's own OS) and run it as root or Administrator:

```
sudo ./eke-setup worker --join eke1-…
```

It downloads the worker program from the site server, refuses it unless its checksum matches the one carried inside the code, installs and starts it as a service, and confirms the controller has registered it. **Treat codes like passwords**: anyone holding one can join a worker to your site until it is used or expires.

- The license sets how many workers each product accepts; a worker beyond it is refused, and the installer says so.
- **File Migration workers** store their job state in the site's database. With the bundled PostgreSQL this is already allowed from the site's own subnet. With your own PostgreSQL, allow the role `<prefix>file_migration_app` to reach the database `<prefix>file_migration` from the workers (`listen_addresses` and `pg_hba.conf`).
- On Linux worker machines, open the worker's port in the firewall too (3051 Data, 3021 Archive, 3001 File).
- To reinstall a worker machine on the same place: `join-code --product data --slot sfdm-2`.

9. Day to day

Task	Command (site server unless noted)
See what runs	`eke-setup status` (any machine)
Logs	Linux: `journalctl -u eke-<service>`; macOS: `/Library/Logs/EKE/`; Windows: `C:\ProgramData\EKE\logs\` (the bundled PostgreSQL logs to the Windows Event Log, Application)
Back up (bundled PostgreSQL)	`eke-setup backup`: the site stops for the copy (usually under a minute) and starts again. The last 5 backups are kept in the data folder's `backups/`; copy them elsewhere too.
Restore	`eke-setup restore` lists backups; `eke-setup restore <name>` puts one back (the current data is kept aside first).
Upgrade	Unpack the new release (its `.pkg`, its `.msi`, which replaces the older release's folder, or its `.tar.gz`) and run **its** `bin/eke-setup upgrade` on the site server. It checks the bundle, backs up (bundled PostgreSQL), installs the new release, checks every service and rolls back by itself if one doesn't come up. It then prints a code per worker machine: run `eke-setup worker --join <code>` there to upgrade it.
Uninstall	On a worker machine or the site server, `eke-setup uninstall` removes the services and programs and keeps the data (on Windows, the **EKE Services On-Prem site server** or **worker** entry in Installed apps does the same; the release's own entry, from the `.msi`, removes only the unpacked release); `--remove-data` deletes the settings, keys, logs and the bundled database too. With your own PostgreSQL, `--drop-databases site.toml` also drops EKE's databases.

Where things are: programs in `/opt/eke` (Linux), `/usr/local/eke` (macOS), `C:\Program Files\EKE` (Windows); settings and data in `/etc/eke` + `/var/lib/eke`, `/Library/Application Support/EKE`, or `C:\ProgramData\EKE`. `eke-setup paths` prints them.

10. Troubleshooting

Symptom	What to do
`port … is in use by another program`	Something else uses that port; stop it, or install on another machine.
A service "did not answer within 90 s"	Read its log (section 9). The installer says the services may be stopped; fix the cause and run the same command again.
"Your password must be changed before you can use this app"	The user still has the one-time password. Sign in to the console (port 3012), choose a new password, then sign in to the other page again.
"Site license is not active" at sign-in	Open the Site License card as the master admin; check the key and the site server's internet access and clock.
A worker is installed but "its controller hasn't listed it"	The license's worker limit is reached, or the site can't reach the worker's port (firewall).

Symptom	What to do
"the site doesn't know this code"	The code was used, expired, or is from another site: create a new one.
`verify` reports a changed file, or "not signed by EKE"	Download the bundle again; don't install a bundle that fails the check.

11. Security notes

- The site's keys are in `site-secrets.env` in the settings folder, readable only by root/Administrators. Losing it means the stored Salesforce credentials can't be decrypted and EKE sees the site as a new install. Include it in your backups.
- Traffic between the site server, workers and browsers is plain HTTP on your LAN. Use a trusted network segment. (HTTPS on the LAN is planned as a separate option.)
- Join codes carry the site's worker keys; pass them only to the person installing the worker.
- EKE receives no Salesforce data or credentials from your site: license renewals send only the license key, an installation identifier, a timestamp and the release in use.