SIDE KICK LAB

AI-POWERED DATA INTELLIGENCE

DEPLOYMENT & SETUP GUIDE

Everything needed to stand up Sidekick inside your environment — infrastructure, installation, configuration, licensing and first discovery.

IN THIS GUIDE

- What you are deploying, and the supported deployment models
- Prerequisites: sizing, networking, accounts and roles
- Step-by-step setup: Windows/SQL, Linux/Docker, configuration
- Licensing, connecting sources, and running your first discovery
- Day-2 operations: updates, backups and troubleshooting

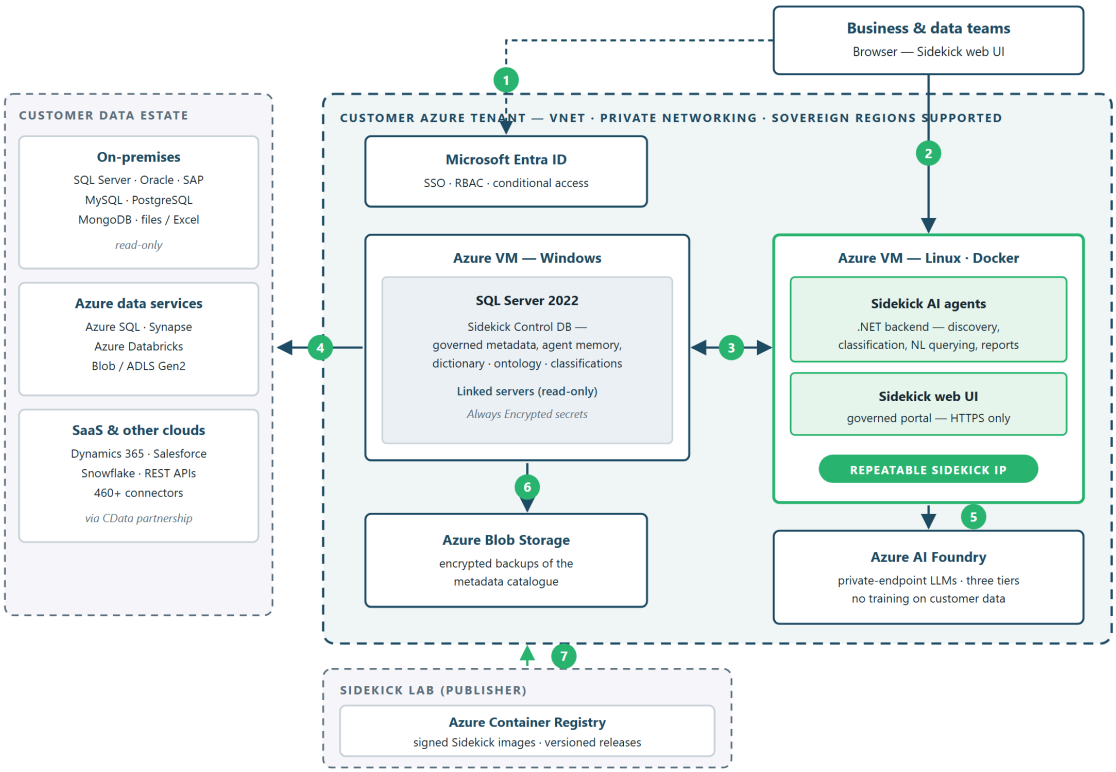
V2.0

JULY 2026

1. WHAT YOU ARE DEPLOYING

Sidekick runs entirely inside your environment — on-premises, private cloud, or your own (sovereign) Azure region. A deployment consists of five parts:

COMPONENT	WHAT IT IS
Windows server	SQL Server 2022 hosting the Sidekick application database (governed metadata, agent memory) **and** the linked servers that provide read-only access to your source systems.
Linux server	Docker host running two containers: the Sidekick backend (.NET agents) and the web front-end, managed with Docker Compose.
Configuration	A single environment file (`sidekick.env`) holding database and LLM connection details — the container images ship with no embedded configuration.
LLM endpoints	Three model tiers (simple / intermediate / complex reasoning), typically deployed in your Azure AI Foundry, or partner-hosted for air-gapped sites.
License	A per-instance license activated through a challenge/response exchange with Sidekick Lab. Discovery is gated until it is applied.



Sidekick reference architecture (Azure deployment)

INFO

Core principle: your data never leaves your boundary. Sidekick is read-only to source systems, transfers only schema and configurable sampled rows into its own database, and its deliverable is metadata, documentation and classification — never copies of raw data.

DEPLOYMENT MODELS

MODEL	SHAPE	NOTES
On-premises	Both servers inside your network	Most secure; no internet access required; LLMs run on a local enterprise AI cluster. Physical (non-VM) servers supported for highly sensitive sites.
Private / sovereign Azure	Both VMs in your Azure tenant	Private endpoints only; sovereign regions supported; LLMs on Azure AI Foundry.
Hybrid	Discovery on-premises, metadata optionally synced to private cloud	No raw data leaves the environment.

2. PREREQUISITES

SIZING

SERVER	MINIMUM	RECOMMENDED (AZURE)	RUNS
Windows	4 vCPU / 16 GB RAM	`D4s_v5`	SQL Server 2022 + SSMS
Linux	2 vCPU / 8 GB RAM	`D4s_v5` (4 vCPU / 16 GB)	Docker + Docker Compose

NETWORKING

Sidekick needs no internet access in normal operation. Required internal routes:

FROM → TO	PORTS	PURPOSE
Windows → Linux	80, 5000 (22 optional)	Front-end / backend access, optional SSH admin
Linux → Windows	1433	SQL Server (application database)
Users → Linux	80 / 443	Sidekick web UI
Windows → sources	per source	Linked-server connections (read-only)
Linux → registry	443 (outbound, optional)	Container image pulls from Azure Container Registry — customer-initiated; can be replaced by offline image transfer for air-gapped sites

ACCOUNTS AND ACCESS TO HAVE READY

- **Local admin / RDP** on both servers for the installation session.
- **A read-only account on every source system** you want Sidekick to catalogue — scoped, least-privilege; no write or admin rights are ever needed.
- **Microsoft Entra ID details** for single sign-on (tenant, app registration — see §6).
- **LLM endpoints and keys** for the three model tiers (see §6).
- **The Sidekick deployment pack** from your partner or Sidekick Lab: `SidekickCreatePopulateSchema.sql`, `SidekickCreateUser.sql`, `LinkedServers.sql` (per-client), `compose.yaml`, `sidekick.env`.

WHO DOES WHAT

ROLE	RESPONSIBILITY
Client internal team	Provisions infrastructure, runs the SQL and Docker setup with the provided scripts, owns all credentials.
Partner	Provides scripts, guidance and support; primary interface during initial deployment.
Sidekick Lab	Technical deployment support, AI model provisioning guidance, license issuance, platform updates.

> **TIP**
>
> RDP access for the partner or Sidekick Lab during initial setup is **optional** — the entire installation can be performed by your own team under guidance.

3. STEP 1 — WINDOWS SERVER (SQL SERVER)

The Windows server hosts SQL Server 2022 with the Sidekick application database and the linked servers to your sources. Use default settings unless noted.

1. Install **Microsoft SQL Server 2022** (basic settings), then the latest **cumulative update**.
2. Install **SQL Server Management Studio (SSMS)** with standard settings.
3. Open SSMS, connect to `localhost`, and open a New Query window.
4. Run the provided `SidekickCreatePopulateSchema.sql` to create the application database.
5. Run the provided `SidekickCreateUser.sql` to create the `sidekick` SQL login — first set a strong, unique password in the script header.
6. In server **Properties** → **Security**, select **"SQL Server and Windows Authentication mode"** (mixed mode — the `sidekick` login uses SQL authentication).
7. In **SQL Server Configuration Manager**, enable the **TCP/IP** protocol.
8. Add a Windows **firewall rule** allowing inbound TCP **1433**.
9. **Restart SQL Server** for the changes to take effect.

> **RULE**
>
> Mixed authentication mode, TCP/IP and the firewall rule are all manual post-install steps, and none apply until SQL Server restarts. Skipping the restart is the classic cause of "the backend can't connect".

LINKED SERVERS — YOUR SOURCES

Every source system Sidekick should discover must exist as a **linked server** on this instance. Add them with the provided per-client `LinkedServers.sql`, using the **read-only account** created on each source.

- Supported directly: SQL Server, Azure SQL, Oracle, PostgreSQL, MySQL/MariaDB, Databricks, MongoDB and more.
- Via the CData connectivity partnership: Dynamics 365, Salesforce, Snowflake, REST APIs and 460+ further sources.

> **DONT**
>
> Never use an admin or read-write account for a linked server. Sidekick requires — and should only ever be given — read-only access to sources.

4. STEP 2 — LINUX SERVER (DOCKER)

The front-end and backend run as Docker containers managed by Docker Compose.

1. Install **Docker** following the official process for your distribution (Ubuntu or RHEL), including the post-installation steps (manage Docker as a non-root user).
2. Create the compose folder: `mkdir -p ~/compose/sidekick`.
3. Copy the provided **`compose.yaml`** and **`sidekick.env`** into that folder, and copy `sidekick.env` to **`.env`**.
4. Edit `.env`: fill in the **database details** (Windows server address, database name, `sidekick` login and password) and the **LLM endpoints and keys**. Leave the `REPO` and `PORT` settings unchanged.
5. Start the stack: `docker compose up -d`, then watch the logs (`docker compose logs -f`) for errors.

On first start against the fresh database, the backend automatically applies its schema migrations and provisions an **encrypted secrets store** (SQL Server Always Encrypted) for sensitive configuration — no manual database preparation or certificate installation is needed.

> **INFO**
>
> The container images contain **no configuration**. If the backend starts with binding or connection errors, the `.env` file is missing values or the SQL Server steps in §3 are incomplete.

Once the containers are healthy, the Sidekick front-end is reachable on the Linux server's address. For production, put your standard TLS termination (e.g. nginx) and DNS hostnames in front — your partner will provide the reverse-proxy template.

5. STEP 3 — LICENSE ACTIVATION

A fresh install starts with **zero discovery credits** — nothing can be discovered until a license is applied. Activation is a challenge/response exchange:

1. Sign in as an administrator and open **Admin** → **License** in the Sidekick UI.
2. Click **Generate challenge** and copy the challenge token.
3. Send the token to Sidekick Lab (or your partner), who sign it and return a license response for the agreed term and discovery volume.
4. Paste the response into **Admin** → **License** to apply it.

- Each discovered database consumes **one discovery credit**; renewals repeat the same exchange.
- The license is bound to the deployment machine and to time — generate the challenge **on the deployment host, shortly before requesting the signed response**, and keep the server clock synchronised.

6. STEP 4 — CONFIGURATION

MICROSOFT ENTRA ID (SINGLE SIGN-ON)

Provide your Entra ID details so users sign in with their corporate identity:

- An **app registration** in your tenant with the Sidekick redirect URI — your partner will supply the exact values and DNS/TLS requirements for your hostnames.
- The **first user to sign in** is bootstrapped as the Sidekick administrator, and can then grant access and roles (Admin / Analyst / Viewer) to others.

LLM ENDPOINTS

Sidekick uses three model tiers for tasks of varying complexity — simple, intermediate and complex reasoning.

- **Option 1 — your Azure AI Foundry**: deploy the models in your own tenant behind private endpoints (recommended on Azure).
- **Option 2 — partner-hosted AI factory**: for fully locked-down or air-gapped environments.

Add the endpoint URLs, keys and model names under the admin settings in the Sidekick front-end. Models can be swapped later as better or cheaper ones become available — no redeployment required.

7. STEP 5 — CONNECT SOURCES AND RUN YOUR FIRST DISCOVERY

1. In the Sidekick UI, open **Data Sources**.
2. Click **Add Linked DB**, select the linked server (from §3) and the database to catalogue, and click **Discover**.
3. Repeat per database — or request discovery across all configured sources.

Discovery runs autonomously: structure scan, sampling, profiling, then the AI passes that generate the data dictionary, ontology, relationships, PII/sensitivity classification and use-case surfacing.

Once discovery completes you can:

- Explore outcomes per source and estate-wide (catalogue, dictionary, ontology, quality and sensitivity views).
- Ask questions in plain English — governed, role-controlled, validated SQL runs behind the scenes.
- Build reports and dashboards, and share access with other users.

8. DAY-2 OPERATIONS

CONCERN	HOW IT WORKS
Updates	Pull new signed images from Azure Container Registry and `docker compose up -d`. Customer-initiated; no impact on the database or agent memory. Offline image transfer available for air-gapped sites.
Backups	Back up the Sidekick application database with your standard SQL Server backup regime (on Azure: encrypted backups to Blob Storage). The containers are stateless.
Monitoring	Container logs via `docker compose logs`; security-auditable events (access, discovery activity, query metadata) in the platform audit trail, SIEM-exportable.
License renewals	Repeat the challenge/response exchange from §5.

9. TROUBLESHOOTING

SYMPTOM	LIKELY CAUSE
Backend cannot connect to the database	Mixed auth mode, TCP/IP, firewall rule or the SQL Server **restart** from §3 was missed; or wrong credentials in `.env`.
Backend starts with binding/connection errors	`.env` missing or incomplete — the images carry no configuration.
UI shows no data sources / nothing to discover	Sources were not added as linked servers on the Windows SQL instance (§3), or the linked-server account lacks read access.
Login does nothing / loops	Entra ID app registration, redirect URI or hostname/TLS mismatch (§6).
Discovery refuses to start	No license credits — apply or renew the license (§5).
Discovery fails against one source	Test the linked server directly in SSMS with the read-only account; verify network path and per-source firewall.

10. DEPLOYMENT CHECKLIST

- [ ] Windows server provisioned (≥ 4 vCPU / 16 GB) and Linux server provisioned (≥ 2 vCPU / 8 GB)
- [ ] Network routes open: Win→Linux 80/5000, Linux→Win 1433, users→UI, Win→sources
- [ ] SQL Server 2022 + CU + SSMS installed
- [ ] Application database and `sidekick` login created (provided scripts), strong unique password set
- [ ] Mixed authentication enabled, TCP/IP enabled, firewall 1433 open, **SQL Server restarted**
- [ ] Read-only accounts created on every source system
- [ ] Linked servers added and tested for every source
- [ ] Docker installed; `compose.yaml` + `.env` in place with DB and LLM details
- [ ] Containers up and healthy (`docker compose up -d`)
- [ ] TLS / hostnames in front of the UI for production
- [ ] Entra ID SSO configured; first admin signed in; roles granted
- [ ] LLM endpoints (three tiers) configured in the admin settings
- [ ] License challenge generated, signed response applied — credits visible
- [ ] First discovery completed and outcomes reviewed

> **TIP**
>
> **Typical elapsed time:** hours, not months — most of it waiting on source read-only accounts and network approvals. Have those in flight before installation day. Questions? **info@sidekicklab.ai · sidekicklab.ai**