DataSitr — Deployment Options

Generated: 2026-05-17T05:18:20Z

Source document: docs/deployment-options.md

Git commit: 460842ce

Generator: [operator-tooling]

Benchmark artifact: docs/generated/pii_benchmark_latest.json (generated 2026-04-29T14:40:52Z, Arabic NER loaded, 1K p95 47.92 ms)

For: Technical champions, IT operations, infrastructure teams

---

Deployment models

DataSitr's live pilot runs on a Saudi-hosted stack. All PII processing, vault storage, and compliance logging happen in-Kingdom. The only routine outbound application traffic is detector-sanitized green-lane text sent to approved external AI providers plus any optional tenant-configured webhook deliveries.

Option 1: DataSitr-hosted pilot (fastest start)

We provision and manage the Saudi-hosted pilot stack.

Aspect	Detail
Setup time	1-2 days
Your responsibility	Provide AI provider API keys, integrate your application
Our responsibility	Stack provisioning, deployment, TLS, monitoring, updates
Access	Dashboard + API over HTTPS; no SSH access to the managed runtime
Cost	Infrastructure cost covered during pilot; production pricing per request

Option 2: Customer-hosted VPS / bare-metal deployment

Your team deploys DataSitr on your own Saudi infrastructure using our deployment guide.

Aspect	Detail

Setup time	1-3 days (depending on infrastructure readiness)
Your responsibility	VPS provisioning, deployment, TLS, backup, monitoring
Our responsibility	Deployment guide, support during setup, software updates
Access	Full control of infrastructure including SSH
Requirements	Ubuntu 22.04+, Docker or Python 3.12+, nginx

See Pilot Deployment Guide for step-by-step instructions.

Option 3: Guarded shared-state / Helm rollout

For teams that want closer parity with the shared-state production posture,

DataSitr also has a guarded Helm-based path used for ACK-style environments.

Treat this as an operator-assisted rollout path, not a self-serve quickstart.

Aspect	Detail
Setup time	Case-by-case; depends on cluster and state-service readiness
Your responsibility	Cluster/network prerequisites, private PostgreSQL + Redis, approved egress/proxy posture
Our responsibility	Guarded chart rollout, values review, smoke checks, rollback procedure
Access	Dashboard + API over HTTPS; cluster access stays operator-scoped by agreement
Requirements	Kubernetes-compatible cluster, Helm, private state services, deployment-specific network policy

Option 4: On-premises

Same as customer-hosted, but on your own datacenter hardware. DataSitr runs as a Docker container or systemd service — no cloud dependency required beyond the external AI provider APIs.

Runtime options

Runtime	Best for	Notes
Docker + Compose (recommended)	Most deployments	Single command, includes dashboard build, health checks, named volume persistence
systemd (bare-metal)	Environments where Docker is not available	Requires Python 3.12+ and Node.js 20+ on the host

Helm / guarded shared-state	Clustered environments that need closer production parity	Uses the shared-state chart/operator path; not positioned as a general self-serve quickstart

Both paths use nginx as a reverse proxy for TLS termination, rate limiting, and security headers.

Authentication options

DataSitr supports two authentication methods that can be used independently or together:

API key authentication (default)

Aspect	Detail
How it works	Bearer token in Authorization header
Key format	prefixed + random string
Storage	Keys are SHA-256 hashed before storage
Roles	`tenant`, `tenant_admin`, `super_admin` (plus separate read-only `regulator` role when provisioned)
Management	Create, revoke, list via API or dashboard
Best for	Service-to-service integration, operator access, API-first workflows

OIDC SSO (optional, additive)

Aspect	Detail
How it works	Authorization Code + PKCE flow via corporate IdP
Tested with	Keycloak (dev), mapped for Entra ID, Okta, Auth0
Session management	Server-side sessions with HttpOnly Secure SameSite cookies
Role mapping	IdP claims map to DataSitr roles via configurable claim names
Best for	Corporate dashboard users, SSO-mandatory environments
Note	API-key auth continues to work when OIDC is enabled

See OIDC SSO Guide for configuration and IdP onboarding.

Environment architecture

Pilot (current)

[See source document for diagram/code]

Staging (available)

A separate staging instance runs on its own VPS with:

- Separate domain and TLS certificate
- Separate `.env` configuration
- Dashboard instance badge showing "Staging"
- 5 safeguard layers preventing staging from affecting production
- OIDC can be tested on staging before enabling on production

Production scaling path

Scale tier	Current posture
Saudi-hosted shared-state pilot stack	**Live now.** This is the current public pilot posture.
Single-host multi-worker	**Demonstrated.** Canonical `worker=2` evidence is captured on the pilot host.
Single-host two-instance shared-state	**Demonstrated.** Two independent API instances served traffic through an evidence proxy against the same PostgreSQL + Redis state.
Two-host shared-state	**Demonstrated.** The first two-host rehearsal passed across the pilot host and the test host over private transport.
ACK public API workload / broader Kubernetes posture	**Live in limited scope.** The current public API workload runs on ACK in Riyadh, and broader Kubernetes continuity proof remains narrower than a blanket HA claim.
In-Kingdom multi-region warm-standby	**Drill-rehearsed (scoped).** GCP Dammam (`me-central2`) warm-standby reachable at `standby.gcp.datasitr.com`; 2026-05-16 signed drill validates DNS, GKE Ingress, TLS routing, and Cloud Armor WAF only.
Cross-cloud DB replication, auto-failover, full-region tolerance	**Not demonstrated.** Future direction; gated on RDS maintenance-window prerequisite and operator approval.

Current evaluator-safe summary:

- Saudi-hosted shared-state is live
- single-host `worker=2` is demonstrated
- same-host two-instance shared-state is demonstrated
- first two-host shared-state behavior is demonstrated
- the current public API workload runs on multi-AZ ACK in Riyadh; the 2026-05-04 customer-route cutover bundle (signed, 4-hour soak) proves multi-AZ ingress HA at the workload layer, while data-tier failover (PostgreSQL/Redis primary failover) remains operator-directed

- an in-Kingdom multi-region warm-standby is live on GCP Dammam (`me-central2`) at `standby.gcp.datasitr.com`; the 2026-05-16 signed scoped drill rehearses DNS, GKE Ingress, TLS routing, and Cloud Armor WAF only
- local non-production Kubernetes continuity appendices also exist for service-path, ingress-path, planned restart, planned node-drain, and separate API-only worker-stop behavior
- cross-cloud database replication, automatic database/authentication failover, and unplanned full-region failure tolerance are not yet demonstrated

Current Kubernetes chart guardrail:

- multi-replica or autoscaled chart renders now fail closed unless PostgreSQL + Redis shared-state is configured and pod-local persistence is disabled
- rehearsal and production chart examples now pin the API container to a read-only root filesystem with explicit writable `/tmp` and `/var/tmp` mounts
- rehearsal values now include a real `NetworkPolicy` example for ingress-nginx ingress plus namespace-scoped state-service egress
- production values now model an explicit proxy-first `NetworkPolicy` shape, but the exact CIDRs, selectors, and approved outbound proxy/gateway targets still remain environment-specific per deployment

Current Kubernetes production egress model:

- app pods should receive traffic only from the chosen ingress controller
- app pods should have direct egress only to DNS and private PostgreSQL / Redis endpoints, plus any optional private OIDC / Keycloak backchannel
- external AI-provider and tenant webhook traffic should traverse an operator-controlled outbound proxy or egress gateway
- raw SaaS FQDN allowlists in `NetworkPolicy` should not be treated as a portable enforcement story

---

Infrastructure requirements

Minimum (pilot)

Resource	Requirement
CPU	2 vCPU
RAM	4 GB

Disk	20 GB SSD
OS	Ubuntu 22.04+
Network	Public IP, ports 80/443 open
Region	Saudi Arabia (Jeddah, Riyadh, or Dammam)

Recommended (production)

Resource	Requirement
CPU	4+ vCPU
RAM	8+ GB
Disk	50+ GB SSD (compliance logs grow over time)
Backup	Off-host replication to cloud storage or remote server
Monitoring	External uptime probe + host-side health monitor

Operational tooling included

Tool	Purpose
`[operator-tooling]`	Guarded workstation-to-VPS deployment with rollback
`[operator-tooling]`	Local encrypted backup (vault + logs + config)
`[operator-tooling]`	Off-host replication to Alibaba Cloud OSS
`[operator-tooling]`	Off-host replication via SSH
`[operator-tooling]`	Restore from backup (interactive or automated)
`[operator-tooling]`	Non-destructive backup validation
`[operator-tooling]`	Post-deploy verification
`[operator-tooling]`	Host-side health and freshness monitor
`[operator-tooling]`	Provider catalog and pricing sync
`[operator-tooling]`	Basic load test scaffold
`[operator-tooling]`	Master key rotation with dry-run

Network and firewall

Inbound

Port	Protocol	Source	Purpose
443	HTTPS	Public	API and dashboard access
80	HTTP	Public	Redirect to HTTPS only
22	SSH	Operator IP only	Server management (customer-hosted)

Outbound

Destination	Purpose
OpenAI API (`[external-provider]`)	Green-lane AI processing
Anthropic API (`[external-provider]`)	Green-lane AI processing
Google AI API (`[external-provider]`)	Green-lane AI processing
STC SambaNova API (`[in-kingdom-provider]`)	Amber/red-lane in-Kingdom AI processing
Webhook endpoints (tenant-configured)	Async result delivery
Certbot / Let's Encrypt	TLS certificate renewal

On the single-VPS pilot these destinations are direct host egress. For shared-state Kubernetes, the recommended production shape is proxy-first: app pods egress directly only to DNS and private state services, while AI-provider and tenant webhook traffic traverses an operator-controlled outbound proxy or egress gateway that owns the approved upstream allowlist.

Related docs

- Pilot Deployment Guide — Step-by-step deployment instructions
- Config Reference — All environment variables
- Operations Runbook — Day-to-day operations procedures
- OIDC SSO Guide — SSO configuration and IdP onboarding
- Disaster Recovery Summary — Backup, restore, and failure scenarios
- Enterprise Readiness FAQ — Common enterprise buyer questions
- Kubernetes Egress Proxy Runbook — Operator procedure for proxy-first outbound control in shared-state Kubernetes
- Kubernetes Readiness Roadmap — Conservative stepwise path from current shared-state pilot to a future cluster rehearsal
- Multi-Host Topology Decision — Chosen first topology for the two-host shared-state rehearsal

---

Version: 0.1.1 | **Last updated:** 2026-05-17

This document describes technical design intent and current operational posture. It does not constitute a warranty, service-level agreement, legal guarantee, or certification of regulatory compliance. DataSitr is designed to support PDPL alignment; it does not itself grant compliance. For the canonical list of safe and unsafe claims, contact gov@datasitr.com.