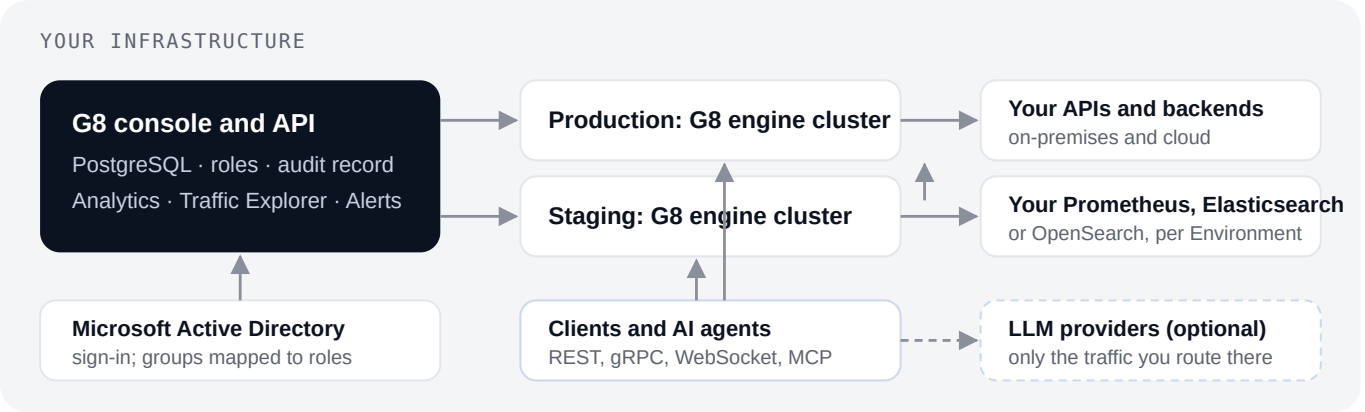
Everything runs in your network: what, where, and how it is secured

What runs where, what you provide, how the installation is secured and what leaves your network. Written for the infrastructure and security teams who review G8 before a proof of concept.

Architecture

G8 is two parts in your network: the console with its management API and PostgreSQL database, and one G8 engine cluster per Environment, managed through the engine's admin interface. Clients and AI agents call the engine; the console never sits in the request path. It reads request counts from your Prometheus and request records from your Elasticsearch or OpenSearch, and it works without internet access.



What runs where

Console and management API	One deployment, several replicas if you want; stateless apart from PostgreSQL. Serves the console, the OpenAPI-documented management API and the Developer Portal.
G8 engine clusters	One per Environment, sized for the traffic of that Environment; add clusters you already run.
Your data stores	PostgreSQL for configuration, users, roles and the audit log; Prometheus and Elasticsearch or OpenSearch per Environment for counts and request records.

What stays and what leaves

Stays in your network

Gateway configuration, users, roles, the audit log, request records, metrics, certificates and secrets. The console reads your stores; it has no vendor connection and no telemetry, and the license is a signed file checked locally.

Leaves only when you route it

Nothing leaves your network except the traffic you route to LLM providers. The AI Plugins that connect to an external service (moderation, Lakera Guard, retrieval) are marked so in the console. Everything else answers inside your network.

Platforms

OpenShift and OKD	One Helm chart for the console and the engine clusters, under the default restricted security profile; Routes for the console and the gateway; the internal database needs the anyuid profile only if you let the chart run it.
Kubernetes	The same chart on any conformant Kubernetes; our build runs it on k3s. Ingress or your load balancer in front.
Docker Compose	For a proof of concept in an afternoon and for small installs; the compose file names every variable.
Virtual machines	Linux hosts with Docker, or the engine installed as a service; the compose file and the installation guide cover both.
Networks without internet access	Mirror the images into your registry; the Helm chart and the compose file take the registry as a value. The license is a signed file checked locally.

Installation, step by step

01

Prepare

A namespace or hosts, PostgreSQL, the registry mirror if the network is closed, and the secrets key you will hold.

02

Install the console

Helm install or docker compose up; the server refuses to start without the secrets key and seeds the roles and permissions.

03

Add the first Environment

Admin API address and key, gateway address, certificate authority, Prometheus and the log store; the connection test reports each one.

04

Map your directory

Bind Microsoft Active Directory, map groups to roles per Environment, sign in once as a Viewer to see the refusal recorded.

05

Apply the index template

Index template and ingest pipeline: records named by Environment and UTC day, stored once per request id. The gateway masks before sending.

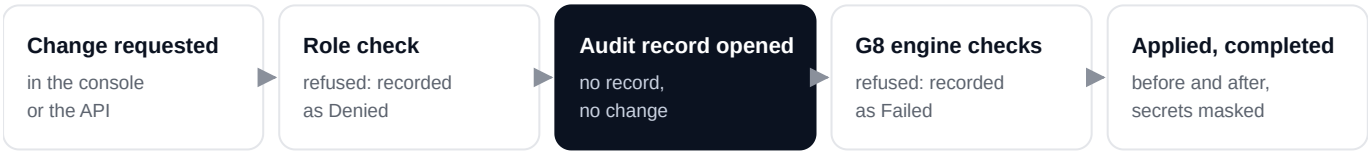
What you provide

A place to install	An OpenShift or Kubernetes cluster, a Docker host or Linux virtual machines. Networks without internet access install from a local registry.
PostgreSQL	One database for every Environment. Tested with PostgreSQL 18.
Prometheus	Optional. For Analytics and Alerts, per Environment or shared with a label per Environment.
Elasticsearch or OpenSearch	Optional. For the request records of Traffic Explorer, per Environment.
Redis	Optional. For rate-limit counters shared across the nodes of a gateway cluster.
Microsoft Active Directory	Optional. For console sign-in and for mapping directory groups to roles.
A mail relay	Optional. For alert emails; alerts can also go to webhooks.
External services	Only the LLM providers you route AI traffic to, and the AI Plugins marked as connecting to an external service.

Security of the installation

Container image	Minimal image, non-root user; no shell tools the service does not need.
OpenShift profile	Runs under the default restricted security context constraint, without extra privileges.
Stored secrets	Engine admin keys and directory bind passwords are sealed with AES-256-GCM under a key you hold (G8_SECRETS_KEY); the server refuses to start without it or with a row sealed under a key it does not hold, and a previous key reseals rows at startup for rotation.
Certificates	Add your organization's certificate authority for every HTTPS connection of an Environment; verification cannot be turned off.
Console sign-in	Local accounts or Microsoft Active Directory over LDAP; directory groups map to roles; every mounted route of the management API declares who may call it, and the server refuses to start otherwise.
Masking	Headers, query parameters and body fields you name are masked before a record is stored; authorization, cookie and API-key headers by default. Viewing a recorded body takes its own permission and is audited.

How a change is applied



Every change is written as an audit record before it is applied; a refused change is a record too. The record is opened before the G8 engine is called; if it cannot be written, nothing is applied. A change a role forbids is a Denied record, one the engine refuses a Failed record with the engine's reason, and a completed record carries the actor, the request id, the IP address and the object before and after, secrets masked.

Versions and limits

Stores	PostgreSQL, tested with 18; Prometheus 3.15; Elasticsearch 8.15 or 9.5, or OpenSearch 2.18 or 3.9. Every build is tested against Prometheus 3.15, Elasticsearch 8.15 and OpenSearch 2.18.
Prometheus	Per Environment, or one Prometheus shared with a label per Environment; request records in one index per Environment and UTC day.
Retention	G8 deletes its own daily indices of request records after 1 to 30 days per Environment and records each deletion; copies your log pipeline takes are not touched. Audit records: 365 days by default, or kept for ever. Alert history: 90 days.

Operations

Upgrades
Install the new image with the same chart or compose file; Enterprise support includes help with every upgrade.

Backup and restore
The database is the state: back it up and restore it with the tools G8 ships; gateway configuration stays in the engine clusters.

Diagnostics
A support bundle collects the configuration, versions and logs a support engineer needs, with secrets left out. Engineering support in Turkish and English is included in Enterprise licenses.

Sizing
Agreed with you during the proof of concept, on the traffic of each Environment.

How we verify

The tests run against the real products, not stubs; a build that cannot reach them fails.

The real engine
Every build runs the test suite against a real G8 engine, Elasticsearch, OpenSearch and Prometheus; a test that cannot reach them fails, none is skipped.

Analytics equal to Prometheus
Each number Analytics shows is compared with Prometheus's own answer for the same window.

Environments kept apart
Every Environment-scoped read is walked against two Environments on two engines and two stores.

Audit on every route
Every route that changes something is walked to prove it leaves a completed audit record.

A real domain controller
Sign-in and group mapping are tested against a Windows domain controller in our lab.

Forms follow the engine
The engine's schema is kept as a snapshot; when an engine version differs, the build goes red until a person reviews the form.

Read more

g8.dev/deployment	What runs where, platforms, security of the installation, operations.
g8.dev/why-g8	Where it runs, change control, one console, your data stores, support, and how we verify.
g8.dev/compare	G8 beside Kong, Apigee, Tyk, Gravitee, Azure API Management, AWS API Gateway and IBM API Connect, from each vendor's own documentation.
g8.dev/downloads	The datasheets of each product area, the evaluation checklist, the comparison and the editions one-pager.

NEXT STEP

Send this brief to your security team and bring their questions to the demo. Request a demo at g8.dev/request-a-demo or write to hello@g8.dev. Read next: API Gateway Evaluation Checklist (1 page), at g8.dev/downloads.

g8.dev/request-a-demo · hello@g8.dev