IntentGate Operator Manual

Pro v2 — Console User Guide

NetGnarus · IntentGate

May 2026

About this manual

This is the day-to-day user guide for the IntentGate operator console. It walks through every page in the sidebar — what it shows, how to use it, and when you would reach for it. Read it once end-to-end to know what IntentGate can do; come back to a section when you need the details.

If you need the install/deploy side (helm chart, env vars, Postgres setup, OIDC config, day-2 ops), that lives in the **IntentGate Deployment Runbook**. This document assumes IntentGate is already running and you're signed in as an operator or admin.

Who this manual is for

- **Platform / SRE engineers** running IntentGate as a service.
- **Security operations teams** investigating audit events, reviewing approvals, and exporting compliance evidence.
- **Admins** managing roles, tenants, integrations, and identity provider settings.

How IntentGate works in 60 seconds

IntentGate sits between your AI agents and the tool servers they call. Every call goes through a four-check pipeline:

1. **Capability** — is the agent's signed token valid, in TTL, and scoped to this tool?
2. **Intent** — what is the agent actually trying to do, and does the policy allow it?
3. **Policy** — your Rego (OPA) rules decide allow / block / escalate, with a reason.
4. **Budget** — has the agent exceeded its per-hour rate or cost cap?

Every decision is signed, hash-chained into a tamper-evident audit log, and forwarded to your SIEM. Operators view, query, approve, and verify through this console.

The sidebar at a glance

Section	Role required	What it's for
Dashboard	viewer	Live overview of decision volume, allow rate, top blockers.
Tokens	viewer	Mint, revoke, attenuate, inspect agent capability tokens.

Section	Role required	What it's for
Policies	admin	Author and edit Rego policies, test them against fixtures.
Audit	viewer	Browse audit events with filters.
Audit verify	admin	Prove the chain is intact; export evidence.
Approvals	viewer/operator	Approve or deny escalate-to-human decisions.
Elevations	admin	Approve other admins' JIT elevation requests.
Compliance	admin	One-click export packs for SOC 2, ISO, GDPR, AI Act, NIST.
Integrations	admin	Wire up SIEM, webhooks, ChatOps.
Notifications	admin	Channel routing and per-event subscriptions.
Security	viewer	Your account: MFA enrollment, session, API tokens.
Elevation	viewer	Request a time-bounded promotion to admin.
Tenants	admin	Create, scope, and isolate tenants.
Settings	admin	License, OIDC, SCIM, role mapping, gateway URL.

The "role required" column is the **minimum**. Higher roles can always see lower-role items. On the demo lab, prospects sign in as admin so they see everything except Lab Access.

Part 1 — Daily operations

Dashboard

The Dashboard is the home page. Five live metric tiles sit across the top with sparkline charts that update every 15 seconds.

- **Tool calls** — total authorization decisions since the gateway started. The sparkline shows the last hour, sampled every 15s.
- **Allow rate** — share of decisions that returned `allow`. A healthy production deployment runs in the 85–98% band; a sudden drop means a policy change or a misbehaving agent.
- **Top blocker** — the policy reason responsible for the most denies in the last hour. Click through to filter Audit by that reason.
- **P95 latency** — 95th-percentile end-to-end latency of `/v1/mcp`. Above 50ms typically means the extractor or an upstream tool server is slow.
- **Revoked tokens** — count of tokens currently in the deny list. Each revocation is also recorded as an audit event.

Below the tiles, a **Revoked tokens** table shows the most recent revocations with reason, operator, and timestamp. Click any row to open that token's full lifecycle in the Tokens page.

When you'd reach for the Dashboard: every morning, after a deploy, during an incident, while you're on a call with a customer.

Tokens

The Tokens page is the operator's primary tool for managing agent capabilities.

The list shows every minted token with: subject (which agent), scope (which tools), tenant, status (active / revoked / expired), TTL remaining, and the operator who minted it. Filter by tenant, status, or free-text search.

Common operations

Mint a token. Click "Issue new". Fill in subject (any string, but typically `service:role`, e.g. `analytics:invoice-bot`), pick a tenant, tick the tools this agent should access, set a TTL (defaults to 7 days), and optionally a per-hour rate ceiling and daily cost cap. Click "Issue". You'll see the token JWT once — copy it into your agent's deployment secrets. We never store the plaintext; only the JWT id (JTI) and metadata.

Revoke a token. Click "Revoke" on the row. Confirm with a reason. The revocation propagates to the gateway in under a second via the gateway's `/v1/admin/deny-list` poll, and every

subsequent call fails the capability check.

Attenuate a token. From the row menu, pick "Attenuate". This mints a child token with a narrower scope (subset of tools, lower TTL, smaller budget). The audit chain links child to parent. Useful for handing a limited token to a sub-agent or short-lived task.

Inspect a token. Click the JTI to see its full audit trail — every decision made with this token, both allows and blocks, with the policy reason for each.

Audit

The Audit page is the searchable log of every authorization decision the gateway has made.

Each row is one decision: timestamp, agent identity, tenant, tool, decision (allow / block / escalate), policy reason, latency, and operator elevation id if the decision happened during an elevation.

Filters

The filter bar (top of page) supports:

- **Time range** — last 15 min, 1h, 24h, 7d, custom.
- **Tenant** — when in multi-tenant mode.
- **Decision** — allow / block / escalate.
- **Reason** — free-text match against policy reason strings.
- **Subject** — exact match against the agent identity in the token.
- **Tool** — exact match against the upstream tool name.
- **Elevation** — events that happened during an admin elevation.

Row drill-down

Click any row to expand. You see:

- The full token JTI used.
- The full Rego policy decision tree (which rules matched).
- The intent classification (if extractor is enabled).
- The request body if intent capture is on.
- The hash that links this event to the previous one in the chain.

Export

The "Export" button at the top right writes the current filter as JSON, NDJSON, or CSV. Useful for regulator requests or feeding into an external SOAR.

Approvals

When a policy returns `escalate-to-human` instead of allow/block, the call pauses and an approval request appears here.

Each row: agent identity, tool, intent summary, policy reason, time pending. Click to expand for the full request body and policy reasoning.

Two buttons: **Approve** lets the call through (the agent gets the response, audit chain stamps you as approver). **Deny** blocks it (the agent gets a structured deny with your reason).

If you don't act within the escalation TTL (default 5 min), the call auto-denies and the agent has to retry. The TTL is configurable per-policy.

When you'd reach for Approvals: real-time during business hours, or paged when something high-stakes (production database write, money movement, account deletion) hits an escalate rule.

Notifications

The Notifications page is where you configure where alerts go and what triggers them.

- **Channels** — Slack, Teams, PagerDuty, Discord, email, webhook. Add a channel with name, type, and credentials/URL.
- **Subscriptions** — pick an event type (block, escalate, revocation, elevation start, audit chain break, license expiry warning, etc.) and route it to one or more channels with optional severity and tenant filters.

Each subscription supports rate limiting (don't flood Slack), and a quiet-hours window.

Part 2 — Policies & enforcement

Policies

The Policies page is where authorization rules live. IntentGate uses Rego (Open Policy Agent's policy language).

The page is split: policy list on the left, editor on the right. Each policy has:

- A name and description.
- The Rego source.
- A test fixture (a sample request) and expected decision.
- A status (enabled, disabled, dry-run).

Editing

Click a policy to load it in the editor. The Rego editor has syntax highlighting and inline linting. Save publishes the change to the gateway within one second.

Dry-run mode is critical for safe rollouts. A dry-run policy runs alongside the live decisions but doesn't enforce — instead it logs what it *would* have decided. Compare dry-run output to live decisions on the Audit page (look for `dry_run: true`). When the dry-run output matches what you want, flip it to enabled.

AI-assisted authoring (Pro)

Click "Suggest a policy". Describe what you want in plain English ("block any agent from calling delete_invoice unless it has elevation"). The console returns a Rego draft. Always review and test before enabling.

Audit verify

The Audit verify page proves that no audit event has been silently deleted or modified since it was first written.

Click "Verify chain". The console fetches the per-tenant chain head from Postgres, walks the chain backward, and recomputes each event hash. It reports:

- Chain integrity (OK or first broken event).
- Number of events verified.
- Time range covered.
- Signature of the chain head (matches the last audit-export bundle).

Use this monthly for SOC 2 evidence, and after any incident where you suspect a breach.

Compliance

The Compliance page is one-click evidence export for regulators and auditors.

Pick a framework: SOC 2, ISO 27001, GDPR, AI Act Article 12, NIST AI RMF, DORA. The console assembles a PDF + JSON pack covering:

- All authorization decisions for the period.
- All operator elevations (who, when, why, what they did).
- All policy versions and the diffs between them.
- The audit chain head signature for the period (proves the bundle hasn't been edited after export).

Pick a date range and click "Generate pack". Pack arrives in your downloads (and is itself an audit event).

Part 3 — Identity & access

Settings → Access (RBAC)

The Access page (under Settings) shows the full RBAC picture for the deployment.

- Who has which role (viewer / operator / admin), grouped by tenant.
- Where the role came from: OIDC claim, SCIM overlay, mock provider, or active JIT elevation.
- When the role was last asserted.

For each user you can see the chain: "OIDC said operator → SCIM overlay said admin → elevation said admin until 14:32". This is the answer to *"why does this user have role X?"* — directly visible, no log-archaeology.

Admins can edit role mappings here too (it writes to the `AUTH_ROLE_MAPPING` env config via a server action).

Security

The Security page is **your own account**, not other users'.

- **MFA enrollment** — scan a QR code into Authenticator / 1Password / Yubikey. Status changes from "not enrolled" to "enrolled" once you successfully verify the first code.
- **Active sessions** — every browser/device with a live session token. Sign-out individual sessions or all-but-this.
- **API tokens** — personal tokens for `igctl` CLI scripts, the SCIM IdP push, or custom automation. Set a TTL on each, scope to specific admin endpoints.

Elevation

The Elevation page is where you (operator) **request** a time-bounded promotion to admin to perform a privileged operation you can't do as operator.

Fill in:

- Why you need elevation (free text, audit logged).
- How long (15 min default, max configurable per-deployment).
- Who's approving (pick an admin who is online).

Submit. The approver gets notified. When they approve, your role flips to admin for the requested window. A countdown banner appears at the top of every page. When it expires, you drop back to operator automatically. Every action you take during the window is stamped with the elevation id in the audit chain.

Elevations

The Elevations page is the **other side** of the same feature: the queue of elevation requests waiting for *you* to approve, plus the history of past ones.

For each pending request: who's asking, what they say they need it for, requested duration. **Approve** flips them to admin and starts the timer. **Deny** drops the request with a reason.

Past elevations are listed below, with full audit detail — who approved, what actions happened during the window, the chain hash that anchors the window.

Tenants

Multi-tenant deployments use this page.

The list shows every tenant: id, display name, plan, monthly volume, last activity. Click a tenant to see its dedicated config — its own audit chain head, its own policy set (or inherits global), its own token quotas, its own integration routing.

Admins create new tenants here. New tenant id auto-propagates to the gateway, and the first decision against it starts a fresh audit chain root for it.

In single-tenant deployments this page is empty (and usually hidden via env config).

Part 4 — Integrations

Settings → Integrations

The Integrations page is where IntentGate talks to the rest of your stack.

SIEM

Three SIEM backends are first-class: **Splunk** (HEC), **Datadog** (Logs Intake API), **Elastic** (Bulk Index API). For each, you supply a URL and a token. The gateway streams every audit event to that backend with at-least-once delivery and a local buffer for outages.

You can wire multiple SIEMs simultaneously (e.g. Splunk for security ops + Datadog for the SRE team).

Webhooks

For anything else (SOAR, custom ETL, your data platform), add a generic webhook. Each event POSTs as JSON to a URL of your choice with an HMAC signature header.

ChatOps

Slack and Teams integrations both (a) route notifications (see Notifications page) and (b) optionally let you respond to approval requests inline — clicking an Approve button in Slack does the same thing as the Approvals page.

Source-of-truth caveat

Integrations *fan out* events; they're never the authoritative copy. The audit chain in Postgres is the source. If a SIEM delivery fails, the audit event is still recorded — you'll see a `siem_delivery_failed` event in Audit.

Part 5 — Lab features (demo only)

These pages only appear when `INTENTGATE_LAB_MODE=true`, which is set on the hosted demo lab. Customer self-hosted deployments don't see them.

Lab Install

`/lab/install` is a ten-step animated replay of how the demo lab was built — secrets, image pull, Postgres init, sidecars, gateway, smoke test, console-pro, observability, Caddy edge, SSO wiring. Each step shows the exact shell command and a one-line explanation.

For prospects: it's the answer to *"OK, how would I install this against my environment?"* — every command is the same as the real install. Click "Start walkthrough" to play it through.

Lab Demo

`/lab/demo` runs the four-control demo against the live lab gateway. It fires a sequence of agent requests that exercise each of the four checks (capability, intent, policy, budget), and streams the gateway's decision feed in real time.

Use it to show a prospect *"here is the four-check pipeline turning these requests into decisions"* without needing them to set up agents themselves.

Lab Access (admin only, hidden from prospects)

`/lab-access` is where you, the lab operator, mint per-prospect credentials. Each row is one prospect: username, email, status (active / revoked / expired), expiry, last login. Click "Issue new credential" to add a new prospect with a name and email; the modal shows the username and password once.

This page is **hidden** from prospect-admins on the demo lab — they're admin in IntentGate but never see other prospects' credentials.

Part 6 — Settings

The Settings page is the deployment-level configuration entry point. Each section below corresponds to a subsection.

Settings → General

- Console version, gateway version, license key status (active / expiring in N days / expired).
- Public URL (where browsers reach this console).
- Theme (light / dark / system).

Settings → Notifications

(Same content as the top-level Notifications page — duplicated under Settings for navigation.)

Settings → Integrations

(Same content as the top-level Integrations page — duplicated under Settings.)

Settings → Security

(Same content as the top-level Security page — duplicated.)

Settings → Elevation

Per-deployment configuration for JIT elevation: max duration, required-approver count, who can approve whom, allowed reasons (free-text vs. dropdown from a fixed list).

Settings → Access

(Same content as the top-level Access page — duplicated.)

Appendix A — Keyboard shortcuts

Shortcut	Action
`/`	Focus the search/filter bar on the current page.
`g d`	Go to Dashboard.
`g t`	Go to Tokens.
`g a`	Go to Audit.
`g p`	Go to Policies.
`?`	Show this shortcut list.
`Esc`	Close any open modal.

Appendix B — When something looks wrong

- **Dashboard shows zeros across the board.** No traffic has hit the gateway. Check `gateway /healthz` and that an agent is actually pointed at it.
- **Allow rate suddenly dropped.** Look at Top blocker — usually a recent policy change. Compare the policy's current vs. previous version on the Policies page.
- **P95 latency spike.** Check the extractor's `/healthz` first (intent classification is the slowest step). Then check Postgres CPU.
- **A token "I just minted" doesn't work.** Confirm tenant scope, and that the agent's clock is within 60s of the gateway (JWT clock-skew tolerance).
- **Audit verify reports a chain break.** Don't panic. The most common cause is a Postgres restore where the chain head pointer rolled back. Snapshot the broken state, file an incident, and follow the Runbook §"Audit chain repair".
- **Console signin works locally but not over the load balancer.** `AUTH_URL` env var has to match the public URL exactly. See the Runbook §"OIDC behind a load balancer".

Appendix C — Glossary

- **Capability token** — signed JWT the agent presents on every call. Encodes subject, tenant, tools, TTL, budget.
- **JTI** — the unique id of a token instance. Lives in the deny-list when revoked.
- **Attenuation** — minting a child token strictly narrower than a parent.
- **Intent classification** — short summary of what the agent is trying to do, produced by the extractor.
- **Escalate-to-human** — a policy decision that pauses the call and creates an Approval.
- **Audit chain** — per-tenant hash-linked sequence of decisions. Tamper-evident.

- **Elevation** — time-bounded promotion of an operator to admin for a documented reason.
- **Dry-run policy** — a policy that logs what it *would* decide without enforcing it.
- **SCIM overlay** — IdP-pushed role/active overlay that wins over OIDC claims on a per-user basis.