

AMASTAN · PILOT RC1 · USER & OPERATOR GUIDE

Amastan AI-SWG

Self-hosted AI Secure Web Gateway — allow / block / safer remote browse, with verdicts and logs on infrastructure you control.

Beginner → Expert

Updated 2026-08-23

Stack: Pilot RC1

Amastan AI-SWG — User & Operator Guide

Audience: Customer IT admins, security operators, and deployment engineers — from first install through expert operations.

Format: Step-by-step procedures with screenshots. Each chapter lists **Beginner / Intermediate / Expert** depth.

Last updated: 2026-08-23 · **Stack baseline:** Pilot RC1 (self-hosted HA appliance)

What Amastan is

Amastan is a **self-hosted** secure web gateway. Staff web traffic is checked on machines you run. Operators see allow / block / remote-browser decisions in a console you own — without sending browsing logs to a vendor cloud by default.

Allow, block, or open safely

Policy can allow a site, block it, or open it in Remote Browser Isolation (RBI) so malware does not run on the PC.

AI for unknown websites

Known lists + policy first; AI helps judge new or unclear URLs instead of only a static category catalog.

Operator console

Rules, traffic decisions, health, identity, and settings in one UI — plus Grafana / Loki observability.

Directory-aware policy

LDAP / Active Directory users and groups — Finance can have different web rules than contractors.

High availability

HA pair so a single box failure does not take the whole filter offline.

Licensed options

DLP (stop secret uploads) and Shadow IT / unofficial cloud apps when enabled on your contract. **LICENSED**

Pilot RC1 means this is the first pilot-ready release candidate: deep self-hosted SWG for lab evaluation. Full cloud SSE parity, ZTNA, and multi-site fleet remain on the roadmap.

How to use this guide

Track	Who	Start here
Beginner	First-time admin	01-beginner/01-first-login.md → 03-onboarding-wizard.md
Deployer	Installs the appliance	02-deployment/01-prerequisites.md → 05-verify-deployment.md
Day-2 operator	Policies, traffic, appeals	03-admin-console/README.md
Expert	HA, API, SIEM, tuning	05-expert/README.md

Screenshots live under **screenshots/**. Re-capture after each major UI release (see **screenshots/README.md**).

Full table of contents

Part 1 — Beginner (first hour)

- 1. First login & console layout
- 2. Dashboard tour
- 3. First-run onboarding wizard

Part 2 — Deployment (step-by-step)

- 1. Prerequisites & sizing
- 2. Install the appliance OS & clone
- 3. Deploy the HA stack
- 4. Windows / endpoint client proxy
- 5. Verify deployment (QA checklist)

Part 3 — Admin console (every screen)

See [03-admin-console/README.md](#) — one page per sidebar area:

Area	Topics
Monitor	Dashboard, Health, Logs, Traffic
Analytics	Shadow IT (licensed)
Policy	Policies, Categories, Wildcards, Appeals
Compliance	DLP (licensed)
Identity	LDAP/AD, Endpoints, Windows agent
Tools	Test URL, Scripts (superadmin)
Administration	Users, Backup, Integrations, Email security
Settings	Network, Billing/license, General

Part 4 — Operations

- Backup & restore
- Upgrades & deploy manifest
- Troubleshooting
- Alerts & on-call → links [docsV2/RUNBOOK.md](#)

Part 5 — Expert reference

- HA architecture & failover
- Classification & fail-open → [docsV2/customer-classification-paths-and-failopen.md](#)
- Licensing & channels → [docsV2/CHANNEL_AND_LICENSING.md](#)
- Pilot connector (Windows) → [docsV2/PILOT_CONNECTOR.md](#)
- Observability & Grafana → [docsV2/OBSERVABILITY.md](#)
- REST API → `docs/api/openapi.json`

Related internal docs (operators)

Doc	Purpose
docsV2/RUNBOOK.md	On-call procedures

Doc	Purpose
docs/ops/PILOT-ALERTS.md	Alert routing
docs/archive/.../FIRST_RUN_WIZARD.md	Onboarding gate details

Documentation status

Section	Status
Structure & TOC	✔ This release
Deployment steps	✔ Draft (text); 📷 screenshots pending
Admin console pages	⚠ Outline; per-page drafts next
Screenshot pack	📷 Placeholders — run capture script on lab

First login & console layout

Level: Beginner · **Time:** 10 minutes

Before you start

- Appliance deployed and Admin UI reachable (default lab: `http://<appliance-ip>:8080`)
- Admin username/password from your deploy runbook (lab default: see operator — **FA-003**: do not use shared passwords in production)

Step 1

Open the admin console

1. In a browser, go to `https://<appliance-fqdn>/` or `http://<appliance-ip>:8080` (lab).
2. You should see the **Sign in** page.



Screenshot: Capture the login page at 1440×900. Save as `screenshots/01-beginner/01-login.png` .

Step 2

Sign in

1. Enter your **username** and **password**.
2. Click **Sign in**.
3. On success you land on **Dashboard**.

If login fails with **429**, wait 60 seconds (rate limit) or ask an operator to clear Redis login counters.

Step 3

Learn the layout

Area	Purpose
Left sidebar	Navigation — Monitor, Policy, Identity, Settings
Top bar	Tenant context, user menu, language

Area	Purpose
Main panel	Page content
Grafana embeds	Live metrics on Dashboard (require Grafana reachable)



Step 4

Roles (what you can see)

Role	Typical access
auditor	Traffic, logs, audit — read only
policy_editor	Policies, blocklist, appeals
admin	Endpoints, email, integrations
superadmin	Users, scripts, backup

Your sidebar may hide items you are not entitled to — that is normal.

Next

- **First-run onboarding wizard** if this is a new appliance.
- **Dashboard tour** for daily monitoring.

Dashboard tour

Level: Beginner

1. **Key metrics** — requests, block rate, latency.
2. **Executive KPIs** — Grafana iframes (classify RPS, block rate).
3. **Verdict summary** — allow/block/rbi split.
4. **Topology** — HA node map.



Dashboard

Expert: Panel UUIDs in `admin-ui/src/shared/config/grafanaDashboards.js`.

First-run onboarding wizard

Level: Beginner · **UI:** `/onboarding`

The wizard walks through **license** → **network** → **endpoints** → **client proxy** → **verify**.

When to use it

- Fresh appliance install
- New customer handoff
- After license renewal or network change

Step 1

Open onboarding

1. Sign in as **admin** or **superadmin**.
2. Sidebar → **First-run setup** (or navigate to `/onboarding`).



Step 2

Activate license

1. Paste your **SWG-LIC-v2** token (from vendor).
2. Click **Activate**.
3. Status must show **ACTIVE** or **GRACE**.

Lab appliances: operator runs `scripts/identity/e2e_mint_license_token.py` + admin activate API.

Step 3

Appliance network (KVM / bare metal only)

Skip on Docker-only eval profiles.

1. Configure management + data interfaces.
2. Confirm until `setup_required: false`.

Step 4

Client proxy / PAC

1. Download PAC file or note proxy `host:port` (lab WinVM: `192.168.122.79:8088`).
2. Run **canary test** from the wizard.

Step 5

Pilot ready

Green banner when:

- License active
- Network configured (if applicable)

Endpoints enrolled: optional for pilot ready.

Expert detail

See archived **FIRST_RUN_WIZARD.md** for API gates (`SETUP_REQUIRED` , PAC IP sync).

Deployment prerequisites

Level: Beginner (planning) / Expert (sizing)

Hardware (pilot / lab reference)

Component	Minimum	Lab appliance
CPU	8 vCPU	12 vCPU
RAM	32 GB	52 GB
Disk	200 GB SSD	300 GB
Network	2 NICs (mgmt + data) recommended	<code>192.168.122.79</code> lab

Software on appliance host

Tool	Version
Ubuntu LTS	22.04 / 24.04
Docker Engine + Compose v2	Required
Git	Clone <code>swg-project</code>
Python 3.11+	QA scripts, JWT mint
Node 20+	Admin UI build (optional on appliance)

Network ports (customer firewall)

Port	Service
80 / 443	HTTP(S) proxy (Envoy/HAProxy)
8080	Admin UI (put behind TLS reverse proxy in production)
8010	Sidecar API (localhost only in HA profile)
8011	Pilot connector API (LAN forward)

Port	Service
8088	Optional LAN proxy forward for VMs

Accounts & secrets before deploy

- [] JWT key pair (`config/jwt/`)
- [] License token (`SWG-LIC-v2`)
- [] Admin password (unique per customer — not lab default)
- [] Grafana / Keycloak passwords if enabled

Next

→ [Deploy the HA stack](#)

Install the appliance OS

Level: Intermediate

Lab (KVM)

1. Import golden QCOW2 or run autoinstall.
2. Static IP on mgmt NIC (lab: `192.168.122.79`).
3. SSH as `swg@<ip>`.
4. Clone repo to `/home/swg/swg-project`.

Production

- Follow hardware guide from vendor.
- Harden SSH, disable password auth, unique admin creds.
- See [docsV2/SECRETS.md](#).

Deploy the HA stack (step-by-step)

Level: Intermediate · **Time:** 30–60 minutes

Step 1

Clone and configure

```
git clone git@github.com:Kabdi-Kalboddine/swg-project.git
cd swg-project
git checkout main
cp .env.example .env    # edit for production
make pilot-bootstrap    # lab helper
```

Step 2

Production env checks

In `.env` for customer pilots:

```
DEV_MODE=false
FAIL_OPEN=false
EDGE_POLICY_ENFORCE=true
ENVIRONMENT=production
SWG_LICENSE_AUTO_ACTIVATE=false    # paste token via UI instead
METRICS_REQUIRE_AUTH=true
```

Step 3

Deploy

```
./scripts/appliance/deploy-lab-appliance-ha.sh
```

For dirty runtime trees during iterative lab work:

```
SWG_ALLOW_DIRTY_DEPLOY=1 ./scripts/appliance/deploy-lab-appliance-ha.sh
```

Step 4

Record manifest

```
./scripts/appliance/record-deploy-manifest.sh
```

Stores git SHA + image digests under `benchmark-results/deploy-manifests/`.
Step 5

Health check

```
./scripts/appliance/verify-appliance-health.sh
curl -sf http://127.0.0.1:8010/api/v1/health
curl -sf -x http://127.0.0.1:80 http://example.com/
```



Screenshot: `docker ps` showing sidecar-a-d, envoy-a/b, redis, mongo healthy.

Step 6

Activate license

Admin UI → Settings → Billing **or** API `PUT /admin/license/activate`.

Step 7

QA sign-off

```
./scripts/appliance/run-lab-qa-manifest.sh
```

Expect **0 FAIL / 0 WARN** on core gates before customer handoff.

Next

→ **Client proxy & Windows agent**

Client proxy & Windows agent

Level: Intermediate

System proxy (Windows lab / pilot)

On the appliance:

```
./scripts/appliance/ensure-pilot-firewall.sh apply
./scripts/appliance/configure-winvm-proxy.sh
./scripts/appliance/verify-winvm-proxy.sh
```

Default lab proxy: `192.168.122.79:8088` → HAProxy `:80`.



Pilot connector (Windows agent)

1. Publish installer from `dist/SWG-Agent-Windows/`.
2. Enroll via `pilot-connector` CLI or GPO.
3. Verify heartbeat: Admin → **Endpoints**.

Expert: [docsV2/PILOT_CONNECTOR.md](#)

Verify deployment

Level: Beginner checklist / Expert QA

Quick smoke (5 min)

Check	Command / URL
Sidecar health	<code>curl http://127.0.0.1:8010/api/v1/health</code>
Proxy allow	<code>curl -x http://127.0.0.1:80 http://example.com/</code>
Admin UI	Browser → <code>:8080</code> login
License	Admin → Settings → Billing → ACTIVE

Full QA manifest

```
./scripts/appliance/run-lab-qa-manifest.sh
```

Results: `benchmark-results/lab-qa-manifest-<id>/results.tsv`

Customer handoff bar: all required rows `PASS`, no `FAIL` / `WARN`.

Pipeline self-test

```
PYTHONPATH=. python3 scripts/appliance/test-lab-pipeline.py --skip-winvm
```

Expect **26+ PASS**.

Document handoff

- ☐ Deploy manifest JSON committed path recorded
- ☐ License ID + expiry noted
- ☐ Admin accounts created
- ☐ This client guide linked in welcome email

Admin console guide (all features)

One chapter per screen — draft status: outline + starter pages; expand with screenshots each sprint.

Monitor

Page	Route	Guide	Screenshot
Dashboard	/	KPIs, Grafana embeds, verdict summary	screenshots/03-admin-console/dashboard.png
Health	/health	Sidecar/envoy/redis/mongo status	health.png
Logs	/logs	Searchable event log	logs.png
Traffic	/traffic	Per-request decisions, delete row	traffic-decisions.png

Traffic — common tasks

1. Filter by user, verdict, or domain.
2. Confirm **JWT user** and **client IP** columns populated.
3. Delete test rows (policy_editor+) — confirm API toast.

Analytics (license: shadow_it_analytics)

Page	Route
Shadow IT	/analytics/shadow-it

Policy

Page	Route	Beginner task
Policy manager	/policies	Add/remove blocklist domain
Categories	/policies/categories	Map site category
Whitelist bulk	/policies/whitelist-tools	CSV import

Page	Route	Beginner task
Wildcards	<code>/wildcards</code>	Learner suggestions
Appeals	<code>/appeals</code>	Approve/deny user block appeal



Compliance (license: dlp_scanning)

Page	Route
DLP	<code>/compliance/dlp</code>

Identity

Page	Route
LDAP / AD	<code>/identity</code>
Endpoints	<code>/endpoints</code>
Windows agent	<code>/endpoints/windows-agent</code>

Tools

Page	Route	Role
Test URL	<code>/tools/test-url</code>	Classify single URL
Scripts	<code>/scripts</code>	superadmin

Administration

Page	Route
Users	<code>/administration/users</code>
Backup	<code>/backup</code>
Integrations	<code>/integrations/log-export</code>

Page	Route
Email security	<code>/email-security</code>

Settings

Page	Route
General	<code>/settings</code>
Billing / License	<code>/settings/billing</code>
Network	<code>/settings</code> → Network tab

Role matrix (expert)

See `admin-ui` RBAC tests: `e2e/rbac-api.spec.ts` + OpenAPI security schemes.

Backup & restore

UI: `/backup` (superadmin)

1. Schedule automated backup via UI or `scripts/appliance/run-automated-backup.sh`.
2. DR drill: `scripts/appliance/pilot-dr-backup-restore-drill.sh`.



Screenshot: backup page after successful run.

Troubleshooting

Symptom	Fix
Proxy empty reply	<code>./scripts/appliance/repair-envoy-proxy.sh</code>
WinVM no proxy	<code>./scripts/appliance/configure-winvm-proxy.sh</code>
License INVALID	Re-sync <code>config/license/vendor_ed25519.pub</code> with private key; re-activate
VM paused (lab host)	Free disk on hypervisor; <code>virsh start swgProject</code>

Full runbook: [docsV2/RUNBOOK.md](#)

Expert track

Deep dives for platform engineers and MSSPs.

Topic	Doc
HA & DR	01-ha-architecture.md + docsV2/data-plane-ha-dr (archive)
Classification	02-classification.md
Licensing	03-licensing.md
Windows agent	04-pilot-connector.md
Observability	05-observability.md
REST API	06-api-reference.md

API quick start

```
# Admin JWT
PYTHONPATH=. python3 scripts/identity/generate_jwt.py --sub ops --roles admin

# OpenAPI
open docs/api/openapi.json
```

Automation

- `scripts/ops/swg-admin.py` — operator CLI
- `scripts/appliance/run-lab-qa-manifest.sh` — regression gate
- `make production-gate` — engineering hardening

Complete guide generated 2026-07-11 22:56 UTC.