

# ALAI Infrastructure — Service Catalog & Runbooks

# ALAI Infrastructure — Service Catalog & Runbooks

“ **Last updated:** 2026-03-11 | **Maintained by:** John (AI Director) **Host:** Mac Studio M3 Ultra (ANVIL) | **OS:** macOS **Quick health:** `node ~/system/tools/daemon-health.js`

## ? Docker Services (23 containers)

### Core Platform Services

Service	Image	Port	External URL	Health	Restart
<b>Vaultwarden</b>	vaultwarden/server	:8200	vault.basicconsulting.no	☐ healthy	<code>cd ~/system/services/vaultwarden &amp;&amp; docker compose restart</code>
<b>BookStack</b>	linuxserver/bookstack	:6875	docs.basicconsulting.no	☐ running	<code>cd ~/system/services/bookstack &amp;&amp; docker compose restart</code>
<b>BookStack DB</b>	linuxserver/mariadb	:3306 (internal)	—	☐ running	Restarts with BookStack
<b>Planka</b>	plankanban/planka	:3100	boards.basicconsulting.no	☐ healthy	<code>cd ~/system/services/planka &amp;&amp; docker compose restart</code>

Service	Image	Port	External URL	Health	Restart
<b>Planka DB</b>	postgres:15-alpine	internal	—	☐ healthy	Restarts with Planka
<b>Documenso</b>	documenso/documenso	:3003	sign.basicconsulting.no	☐ running	<code>cd ~/system/services/documenso &amp;&amp; docker compose restart</code>
<b>Documenso DB</b>	postgres:15-alpine	internal	—	☐ healthy	Restarts with Documenso
<b>Documenso MinIO</b>	minio/minio	:9002/:9003	—	☐ running	Restarts with Documenso
<b>Baikal (CalDAV)</b>	ckulka/baikal:nginx	:5232	calendar.basicconsulting.no	☐ running	<code>cd ~/system/services/baikal &amp;&amp; docker compose restart</code>
<b>Qdrant (Vector DB)</b>	qdrant/qdrant	:6333/:6334	—	☐ running	<code>docker restart qdrant</code>

## Product Database Services

Service	Port	Product	Health	Restart
<b>drop-postgres</b>	:5433	Drop	☐ healthy	<code>cd ~/ALAI/products/Drop &amp;&amp; docker compose restart drop-postgres</code>
<b>plock-db</b>	:5434	Plock	☐ healthy	<code>cd ~/ALAI/products/Plock &amp;&amp; docker compose restart plock-db</code>
<b>plock-redis</b>	:6380	Plock	☐ healthy	Restarts with plock-db
<b>bilko-postgres</b>	:5436	Bilko	☐ running	<code>cd ~/ALAI/products/Bilko &amp;&amp; docker compose restart bilko- postgres</code>
<b>bilko-redis</b>	:6382	Bilko	☐ running	Restarts with bilko
<b>lobby-postgres</b>	:5437	Lobby	☐ healthy	<code>cd ~/ALAI/products/Lobby &amp;&amp; docker compose restart lobby- postgres</code>
<b>lumiscare-postgres</b>	:5432	LumisCare	☐ healthy	Client project
<b>lumiscare-redis</b>	:6379	LumisCare	☐ healthy	Client project

Service	Port	Product	Health	Restart
backend-postgres	:5435	BasicFakta	☐ healthy	cd ~/ALAI/products/BasicFakta && docker compose restart
backend-redis	:6381	BasicFakta	☐ healthy	Restarts with backend

## Monitoring Stack (Drop)

Service	Port	URL	Restart
Grafana	:3300	grafana.basicconsulting.no	docker restart drop-grafana
Prometheus	:9090	prometheus.basicconsulting.no	docker restart drop-prometheus
Node Exporter	:9100	—	docker restart drop-node-exporter

## ?? Cloudflare Tunnel (cloudflared)

**LaunchAgent:** `com.john.cloudflared` **Config:** `~/cloudflared/config.yml` **Tunnel ID:** `3315a609-7934-45c5-ad0c-56d86d16374d`

## Exposed Services

Hostname	Backend	Purpose
docs.basicconsulting.no	localhost:6875	BookStack wiki
vault.basicconsulting.no	localhost:8200	Vaultwarden
sign.basicconsulting.no	localhost:3003	Documenso (e-signing)
boards.basicconsulting.no	localhost:3100	Planka (kanban)
calendar.basicconsulting.no	localhost:5232	Baikal (CalDAV)
mc.basicconsulting.no	localhost:3030	MC Dashboard
api.basicconsulting.no	localhost:3001	API gateway
drop-api.basicconsulting.no	localhost:3201	Drop API
lobby.basicconsulting.no	localhost:3010	Lobby frontend
lobby-api.basicconsulting.no	localhost:3009	Lobby API

Hostname	Backend	Purpose
auth.basicconsulting.no	localhost:9000	Authentik (SSO)
grafana.basicconsulting.no	localhost:3300	Grafana dashboards
prometheus.basicconsulting.no	localhost:9090	Prometheus metrics
track.basicconsulting.no	localhost:3456	Email tracking pixel
ssh.basicconsulting.no	localhost:22	SSH access
vnc.basicconsulting.no	localhost:5900	VNC screen sharing

## Runbook: Tunnel down

```
# Check status
launchctl list | grep cloudflared

# Restart
launchctl stop com.john.cloudflared
launchctl start com.john.cloudflared

# Verify
cloudflared tunnel info 3315a609-7934-45c5-ad0c-56d86d16374d

# Logs
tail -50 ~/system/logs/cloudflared.log
```

## ? Vaultwarden

**Container:** vaultwarden | **Port:** :8200 **URL:** vault.basicconsulting.no (Cloudflare Access protected)

**Local:** http://localhost:8200 | **HTTPS proxy:** https://localhost:8443 (Caddy) **Admin token:** In

~/system/services/vaultwarden/.env

## Dependencies

- Docker
- Caddy HTTPS proxy ( `com.john.caddy-vault` ) — needed for `bw` CLI
- vault-keeper daemon ( `com.john.vault-keeper` ) — auto-unlock

## Runbook: Vault locked/unauthenticated

```
# Check status
NODE_TLS_REJECT_UNAUTHORIZED=0 bw status

# If "locked" – vault-keeper auto-fixes every 15 min. Manual:
NODE_TLS_REJECT_UNAUTHORIZED=0 bw unlock --raw > /tmp/bw-session

# If "unauthenticated" – needs full re-login:
NODE_TLS_REJECT_UNAUTHORIZED=0 bw login --apikey
# Enter client_id and client_secret from ~/system/config/vault-apikey.json
# Then unlock:
NODE_TLS_REJECT_UNAUTHORIZED=0 bw unlock --raw > /tmp/bw-session

# Verify
NODE_TLS_REJECT_UNAUTHORIZED=0 BW_SESSION=$(cat /tmp/bw-session) bw list items --search
"Email" | head
```

## Runbook: Caddy proxy down

```
# Caddy provides HTTPS for bw CLI (self-signed cert)
launchctl list | grep caddy-vault
# Restart
launchctl stop com.john.caddy-vault && launchctl start com.john.caddy-vault
# Verify
curl -sk https://localhost:8443 | head -1
```

## ? Email System

**Daemon:** `com.john.email-agent` (every 5 min) **Accounts:** john@basicconsulting.no, info@basicconsulting.no, john@alai.no, alem@alai.no, dev@alai.no **IMAP:** imap.one.com:993 | **SMTP:** send.one.com:465 **Credentials:** Vaultwarden (via bw CLI)

## Runbook: Email agent not processing

```
# Check logs
tail -30 ~/system/logs/email-agent-launchd.log

# Common issue: Vault not unlocked
```

```
NODE_TLS_REJECT_UNAUTHORIZED=0 bw status
# Fix: See Vaultwarden runbook above

# Manual test run
NODE_TLS_REJECT_UNAUTHORIZED=0 node ~/system/daemons/email-agent.js --dry-run

# Restart daemon
launchctl stop com.john.email-agent && launchctl start com.john.email-agent

# Check inbox DB
node -e "const e=require('$HOME/system/tools/email-inbox.js');console.log(JSON.stringify(e.getStats(),null,2))"
```

## ? Telegram Bot

**Daemon:** `com.john.telegram-agent` (KeepAlive) **Bot:** @johnbasicas\_bot **Config:** macOS Keychain (telegram-bot-token) **AI Backend:** Claude CLI → Ollama (llama3.1:8b) → static fallback

## Runbook: Bot not responding

```
# Check daemon
launchctl list | grep telegram-agent

# Check logs
tail -20 ~/system/logs/telegram-agent.log

# Restart
launchctl stop com.john.telegram-agent && launchctl start com.john.telegram-agent

# Test AI backend
node -e "const{getResponse}=require('$HOME/system/tools/comms-responder.js');getResponse('test', []).then(r=>console.log(r.backend, r.text.substring(0,100)))"

# Test connection
node ~/system/tools/telegram-agent.js --test
```

# ? Slack Bot

**Daemon:** `com.john.slack-bot` (KeepAlive) **Workspace:** ALAI Holding AS

## Runbook: Slack bot not responding

```
launchctl list | grep slack-bot
tail -20 ~/system/logs/slack-bot.log
launchctl stop com.john.slack-bot && launchctl start com.john.slack-bot
```

# ? BookStack (Wiki)

**Container:** bookstack + bookstack\_db **Port:** :6875 | **URL:** docs.basicconsulting.no **API config:** ~/system/config/bookstack.json (creds in Vaultwarden)

## Runbook: BookStack down

```
cd ~/system/services/bookstack
docker compose ps
docker compose restart
# Check logs
docker logs bookstack --tail 20
```

# ? Documenso (E-Signing)

**Containers:** documenso + documenso-db + documenso-minio **Port:** :3003 | **URL:** sign.basicconsulting.no

## Runbook: Documenso down

```
cd ~/system/services/documenso
docker compose ps
docker compose restart
docker logs documenso --tail 20
```

---

# ? Planka (Kanban)

**Containers:** planka + planka-db **Port:** :3100 | **URL:** boards.basicconsulting.no

## Runbook: Planka down

```
cd ~/system/services/planka
docker compose ps
docker compose restart
docker logs planka --tail 20
```

---

# ? Baikal (CalDAV/CardDAV)

**Container:** baikal **Port:** :5232 | **URL:** calendar.basicconsulting.no

## Runbook: Baikal down

```
cd ~/system/services/baikal
docker compose ps
docker compose restart
docker logs baikal --tail 20
```

---

# ? Ollama (Local AI)

**Process:** ollama serve (background) **Port:** :11434 **Models:** llama3.1:8b, qwen2.5-coder:32b, bge-m3, llama-guard3:8b, custom ALAI models

## Runbook: Ollama down

```
# Check
curl -s http://localhost:11434/api/tags | python3 -m json.tool | head

# Restart
```

```
ollama serve &

# Verify models
ollama list
```

## ?? Key LaunchAgent Daemons

Daemon	Label	Purpose	Priority
Cloudflared	com.john.cloudflared	Tunnel to internet	P1
Vault Keeper	com.john.vault-keeper	Auto-unlock Vaultwarden	P1
Caddy Vault	com.john.caddy-vault	HTTPS proxy for bw CLI	P1
Slack Bot	com.john.slack-bot	Slack communication	P1
Telegram Agent	com.john.telegram-agent	Telegram bot	P1
Email Agent	com.john.email-agent	Email processing	P1
Email Tracker	com.john.email-tracker	Open/click tracking	P2
Comms Agent	com.john.comms-agent	Cross-platform comms	P2
Ops Watchdog	com.john.ops-watchdog	Service health checks	P1
Event Dispatcher	com.john.event-dispatcher	Event bus processing	P1
Pi Orchestrator	com.john.pi-orchestrator	Task delegation to agents	P1
Autowork	com.john.autowork	Background task execution	P2
N8N	com.john.n8n	Workflow automation	P2
MC Dashboard	com.john.mc-dashboard	Mission Control web UI	P2

## Generic daemon restart

```
# Stop
launchctl stop com.john.<name>

# Start
launchctl start com.john.<name>

# Full reload
launchctl unload ~/Library/LaunchAgents/com.john.<name>.plist
launchctl load ~/Library/LaunchAgents/com.john.<name>.plist

# Check status
launchctl list | grep <name>
```

---

# ? Cold Start (Full System Bring-Up)

If the Mac Studio reboots:

```
# 1. Docker starts automatically (Docker Desktop)
# 2. LaunchAgents auto-load (RunAtLoad=true)
# 3. vault-keeper unlocks Vaultwarden (reads Keychain)
# 4. All services come up within ~2 minutes

# Verify everything:
bash ~/system/ops/cold-start.sh
node ~/system/tools/daemon-health.js
docker ps
```

---

# ? Emergency Contacts

- **Alem Basic** (CEO): alem@alai.no
- **John** (AI Director): john@basicconsulting.no, @johnbasicas\_bot (Telegram), #exec (Slack)

---

Revision #3

Created 2026-03-11 09:45:51 UTC by John

Updated 2026-05-31 20:05:12 UTC by John