

# Pillar #9 LITE — DR-only bridge & backup

Build records and runbooks for Pillar #9 LITE: DR-only bridge deployment on vm-alai-support.

- [Build A: Telegram Bot on vm-alai-support](#)

# Build A: Telegram Bot on vm-alai-support

<b>MC Reference</b>	#99247
<b>Build Date</b>	2026-05-05
<b>Builder</b>	Kelsey Hightower (FlowForge)
<b>Validator</b>	Angie Jones (Proveo) — PASS 5/5 ACs
<b>Related</b>	MC #99290 (response quality fix, separate workstream)

## Runbook: ALAI Telegram Agent on vm-alai-support

**MC:** #99247 | **Date:** 2026-05-05 | **Agent:** Kelsey Hightower (FlowForge)

**Status:** COMPLETE — service active+enabled on vm-alai-support

### What Was Deployed

The Telegram bot bridge (John AI) was migrated from ANVIL (laptop LaunchAgent) to `vm-alai-support` (Azure VM) as a systemd service. The bot now runs 24/7 independent of ANVIL uptime.

### File Locations on VM

File	Path
Main daemon	<code>/opt/alai/system/tools/telegram-agent.js</code>
AI responder	<code>/opt/alai/system/tools/comms-responder.js</code>
Intake logger	<code>/opt/alai/system/tools/intake.js</code>
ALAI config stub	<code>/opt/alai/system/lib/alai-config.js</code>
Config mock	<code>/opt/alai/system/tools/alai-config-mock.json</code>
Prompt files	<code>/opt/alai/system/prompts/extracted/comms-responder/*.md</code>

File	Path
npm packages	<code>/opt/alai/system/tools/node_modules/</code>
Systemd unit	<code>/etc/systemd/system/alai-telegram-agent.service</code>
Environment file	<code>/etc/alai-telegram.env</code> (mode 600, owned root)
Log	<code>/opt/alai/system/logs/telegram-agent.log</code>
Intake DB	<code>/opt/alai/system/databases/intake.db</code>
State	<code>/opt/alai/system/config/telegram-bot-state.json</code>

## Environment Variables (all in `/etc/alai-telegram.env`)

- `TELEGRAM_BOT_TOKEN` — Bot token (from Bitwarden: Telegram Bot item)
- `TELEGRAM_ALLOWED_USERS` — JSON array of allowed Telegram user IDs (e.g. [8454016834])
- `ANTHROPIC_API_KEY` — From Bitwarden: Anthropic's key john@basicconsulting.no (notes field)
- `GROQ_API_KEY` — From Bitwarden: Groq api key
- `HOME=/opt/alai` — Critical: makes all path resolutions resolve correctly under `/opt/alai/system/`
- `NODE_ENV=production`
- `COMMS_OLLAMA_HOST=http://localhost:11434` — Ollama not available on VM; dead-letter fallback

## Code Change Applied to `telegram-agent.js`

**Reason:** On the VM, macOS `security` Keychain command does not exist, causing `loadConfigFromKeychain()` to fail, then falling through to `require('./vault')` which also does not exist.

**Fix (MC #99247, 2026-05-05):** Added env-var first check at top of `loadConfig()` function (lines 164-195 of updated file). When `TELEGRAM_BOT_TOKEN` env var is present (set via systemd EnvironmentFile), it loads config from environment without touching Keychain or vault. This is additive — existing macOS/vault behavior unchanged.

## VM-Specific Stubs Created

# /opt/alai/system/lib/alai-config.js

Lightweight stub replacing the full ALAI SDK (which requires `~/ALAI/internal/packages/alai-config-ts/` — not present on VM). Reads from `alai-config-mock.json`. Uses `process.env.HOME` (not `os.homedir()`) to resolve paths correctly.

# /opt/alai/system/tools/intake.js

Minimal intake stub providing `enqueue()` function backed by `better-sqlite3`. Creates `intake.db` at `HOME/system/databases/intake.db` on first use.

## npm Packages Installed

Location: `/opt/alai/system/tools/node_modules/`

- `@anthropic-ai/sdk` — Claude API backend
- `better-sqlite3` — intake.db SQLite writes

## Systemd Unit

```
/etc/systemd/system/alai-telegram-agent.service
User: alai-admin
WorkingDirectory: /opt/alai/system
ExecStart: /usr/bin/node /opt/alai/system/tools/telegram-agent.js
EnvironmentFile: /etc/alai-telegram.env
Restart=always, RestartSec=10
NoNewPrivileges=true, PrivateTmp=true
```

## ANVIL Rollback

The ANVIL daemon was disabled — NOT deleted:

- Plist renamed: `~/Library/LaunchAgents/com.john.telegram-agent.plist.disabled-2026-05-05`
- To reactivate on ANVIL: `mv .plist.disabled-2026-05-05 .plist` then `launchctl load` the plist
- Remember to stop VM service first via `az vm run-command invoke` before reactivating ANVIL

# Operational Commands

Check status:

```
az vm run-command invoke -g rg-alai-support -n vm-alai-support --command-id RunShellScript --scripts "systemctl status alai-telegram-agent --no-pager"
```

View logs:

```
az vm run-command invoke -g rg-alai-support -n vm-alai-support --command-id RunShellScript --scripts "cat /opt/alai/system/logs/telegram-agent.log | tail -20"
```

Restart service:

```
az vm run-command invoke -g rg-alai-support -n vm-alai-support --command-id RunShellScript --scripts "systemctl restart alai-telegram-agent"
```

Check intake.db (last 10 messages):

```
az vm run-command invoke -g rg-alai-support -n vm-alai-support --command-id RunShellScript --scripts "sqlite3 /opt/alai/system/databases/intake.db 'SELECT created_at, sender, content FROM messages ORDER BY created_at DESC LIMIT 10;'"
```

## SSH Note

SSH (port 22) to vm-alai-support requires source IP whitelisting in NSG rule AllowSSH. Currently whitelisted: 46.46.247.202, 92.221.168.61, 46.46.248.119

For deployment/maintenance from ANVIL (current IP 46.46.253.58), either add ANVIL IP to NSG rule (az network nsg rule update) or use az vm run-command invoke (no IP restriction, used throughout this deployment).

## Proveo E2E Test Instructions (AC2 + AC4 validation)

1. CEO sends a Telegram message to the bot from their phone
2. Verify: bot responds within 30s (AI response via Claude API)
3. Verify on VM: sqlite3 intake.db SELECT query — should show the message
4. Verify: systemctl status alai-telegram-agent shows active (running)
5. Verify: log at /opt/alai/system/logs/telegram-agent.log shows Message received and AI response sent entries