

ANVIL DR Bootstrap Runbook (Mac Air)

ANVIL DR Bootstrap Runbook (Mac Air)

When to use

This runbook is for recovering the ALAI AI factory infrastructure when:

- ANVIL (Mac Studio, 100.103.49.98) is dead, stolen, or inaccessible
- Hardware failure requiring complete rebuild on new Mac
- Setting up FORGE (disaster recovery clone) on fresh hardware
- Provisioning a new MacBook Air for Alem with minimal AI factory capabilities

SPOF Context: As of 2026-04-20, ANVIL is the single Mac Studio hosting 112 LaunchAgent daemons, 68 SQLite databases (litestream-replicated), Ollama (8 models), and the entire ~/system + ~/.claude infrastructure. This runbook enables recovery to any fresh Mac with admin access.

Prerequisites

Before starting bootstrap, ensure you have:

1. **Fresh Mac** with admin account (macOS Sonoma or later, Apple Silicon preferred)
2. **Tailscale app** installed + logged into `alembasic@` tailnet (download from tailscale.com/download)
3. **GitHub account** with read access to:
 - `github.com/johnatbasicas/clawd` (~/system repo, auto-backup branch)
 - `github.com/johnatbasicas/claude-config` (~/claude repo)
4. **Bitwarden account unlocked** with master password ready (Alem's personal vault: `alembasic@gmail.com`)
5. **Internet connection** (stable, for 2-3 GB of Homebrew packages + Ollama models)

Step-by-step Bootstrap

Phase 1: Foundation

1. Install Xcode Command Line Tools

```
xcode-select --install
```

Expected: GUI dialog appears. Click "Install" and wait 5-10 minutes. Verify with:

```
xcode-select -p  
# Should output: /Library/Developer/CommandLineTools
```

2. Install Homebrew

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Expected: Homebrew installs to `/opt/homebrew`. Add to shell profile:

```
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile  
eval "$(/opt/homebrew/bin/brew shellenv)"  
  
# Verify:  
brew --version  
# Should show: Homebrew 4.x.x
```

3. Install Bitwarden CLI + unlock vault

```
brew install bitwarden-cli  
  
# Unlock vault (enter master password when prompted):  
bw login alembasic@gmail.com  
export BW_SESSION=$(bw unlock --raw)  
  
# Verify:  
bw status | jq .status  
# Should show: "unlocked"
```

Note: Keep this terminal window open. `BW_SESSION` is needed for bootstrap script.

Phase 2: Clone Infrastructure Repos

4. Clone ~/system (clawd repo)

```
# If using SSH (recommended if SSH keys already set up):
git clone git@github.com:johnatbasicas/clawd.git ~/system

# OR if using HTTPS with GitHub PAT:
git clone https://github.com/johnatbasicas/clawd.git ~/system

# Switch to auto-backup branch (contains latest portability artifacts):
cd ~/system
git checkout auto-backup
git pull
```

Expected:

```
ls ~/system/
# Should show: Brewfile, bootstrap.sh, config/, databases/, tools/, etc.
```

5. Clone ~/.claude (claude-config repo)

```
git clone git@github.com:johnatbasicas/claude-config.git ~/.claude

# Verify:
ls ~/.claude/
# Should show: CLAUDE.md, hooks/, agents/, skills/, projects/
```

Phase 3: Run Bootstrap Script

6. Execute bootstrap (with BW_SESSION active)

```
cd ~/system
bash bootstrap.sh workstation
```

Role options:

- `anvil`: Full primary node (all daemons, Ollama, heavy workloads)
- `forge`: DR clone (continuous restore from Azure, lighter load)
- `workstation`: Minimal setup (SSH relay to ANVIL for heavy ops)

What the script does:

1. Re-checks Xcode CLT + Homebrew (idempotent)
2. Installs ~70 brew packages from Brewfile (15-30 min depending on connection)
3. Copies 112 LaunchAgent plists from `~/system/config/launchagents/` to `~/Library/LaunchAgents/`
4. Rehydrates `BW:<item>` placeholders in plists by calling `bw get password <item>`
5. Loads all LaunchAgents via `launchctl bootstrap`
6. Verifies core services (Ollama, litestream)

Expected output (tail of bootstrap.log):

```
[bootstrap] Bootstrap COMPLETE. Next steps:
[bootstrap]   - Verify SSH: ssh makinja@100.103.49.98
[bootstrap]   - Check MC: node ~/system/tools/mc.js list
[bootstrap]   - Log: /Users/makinja/bootstrap.log

LaunchAgents loaded: 112
Ollama models available: 8
Litestream: RUNNING
```

If BW rehydration fails: You'll see warnings like:

```
WARN: Bitwarden item 'groq-api-key' not found – com.alai.groq-model-benchmark.plist will need manual fix
```

Fix manually after bootstrap completes (see Troubleshooting section).

Phase 4: Database Restore (if DBs lost/corrupt)

When to run: Only if `~/system/databases/` is empty or you need to restore from Azure backups (e.g., ANVIL disk died).

7. Set Azure auth environment variables

```
export AZURE_CLIENT_ID="1a0b3018-0c31-474b-918f-531b0a29a669"
export AZURE_CLIENT_SECRET=$(bw get password alai-backup-writer-secret)
export AZURE_TENANT_ID="cd0a7929-1d14-4f81-820d-b36e45f72cf7"
```

8. Restore P0 critical databases

```
mkdir -p ~/system/databases

# Mission Control:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/mission-control.db

# HiveMind:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/hivemind.db

# Tasks:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/tasks.db

# Costs:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/costs.db

# Events:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/events.db
```

9. Restore P0 financial databases

```
# Fiken (accounting cache):
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/fiken.db

# Invoices:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/invoices.db

# Contracts:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/contracts.db

# Leads:
litestream restore -config ~/system/config/litestream.yml -if-replica-exists
```

```
~/system/databases/leads.db
```

Note: The `-if-replica-exists` flag prevents overwriting local DB if it's newer than Azure backup. Omit this flag to force restore.

Bulk restore all 68 DBs (if needed):

```
for db in mission-control hivemind tasks costs events fiken invoices contracts leads \
    orchestrator-queue orchestrator-workers durable-runner session-index knowledge \
    emails email-inbox alem-directives agent-routing bee-index companies contacts \
    deploy-registry design-reviews distill documents drafts drift email-audit \
    email-briefing email-index email-tracking escalations facts flywheel goals \
    guardrails-audit health-events hivemind-archive master-control mc minions \
    observability orchestrator-events pipeline projects routing-outcomes skill-
improvements \
    skill-registry sprint-pipeline strategy-tracker teams tenders tickets tool-audit \
    tool-registry trace-events applications-tracker baikal-caldav prompt-cache \
    prompt-metrics semantic-reuse-index stbs telemetry token-cost usage vcr bih-tenders
browser-tasks; do
    echo "Restoring $db..."
    litestream restore -config ~/system/config/litestream.yml -if-replica-exists
~/system/databases/$db.db || echo "WARN: $db restore failed or skipped"
done
```

Verify restores:

```
ls -lh ~/system/databases/*.db | wc -l
# Should show: 68 (or close, depending on which DBs had replicas)

# Check specific DB integrity:
sqlite3 ~/system/databases/mission-control.db "PRAGMA integrity_check;"
# Should output: ok
```

Bitwarden Items Required

The following Bitwarden vault items **MUST** exist in Alem's vault before running bootstrap. These are referenced as `BW:<item>` placeholders in LaunchAgent plists:

Item Name	Used By	Purpose
-----------	---------	---------

alai-backup-writer-secret	litestream, Azure backups	Azure SP client secret for Storage Blob write access
cf-access-client-secret	BookStack sync, CF-protected APIs	Cloudflare Access client secret for docs.basicconsulting.no
groq-api-key	Groq model benchmark daemon	Groq API key for LLM model testing
slack-app-token	Slack integration	Slack app-level token for socket mode
slack-bot-token	Slack integration	Slack bot user OAuth token (xoxb-...)

How to verify items exist:

```
bw get item alai-backup-writer-secret --session $BW_SESSION
bw get item cf-access-client-secret --session $BW_SESSION
bw get item groq-api-key --session $BW_SESSION
bw get item slack-app-token --session $BW_SESSION
bw get item slack-bot-token --session $BW_SESSION
```

If missing: Contact Alem or check Vaultwarden (<https://vault.basicconsulting.no>) for backup credentials. These secrets are also in ANVIL's Keychain if ANVIL is still accessible.

Post-Bootstrap Verification

10. Check LaunchAgents loaded

```
launchctl list | grep -E "com\.(alai|john)\." | wc -l
# Expected: ~110-112 (depending on role)
```

11. Verify Ollama running

```
curl -s http://localhost:11434/api/tags | jq -r '.models[].name'
# Expected (ANVIL): qwen2.5-coder:32b, llama3.3, deepseek-r1, etc.
```

12. Verify litestream replication

```
pgrep -fl litestream
# Should show: litestream replicate -config /Users/makinja/system/config/litestream.yml

# Check logs:
tail -f ~/system/logs/litestream.log
# Should show periodic sync messages (every 1-30s depending on DB tier)
```

13. Test Mission Control

```
node ~/system/tools/mc.js stats
# Should show task counts, agents, recent activity

node ~/system/tools/mc.js list --limit 5
# Should show recent tasks
```

14. Test SSH to original ANVIL (if still alive)

```
ssh makinja@100.103.49.98 "hostname && uptime"
# Expected: ANVIL + uptime if machine is reachable
```

Troubleshooting

Error: "brew: command not found" after install

Cause: Homebrew not in PATH.

Fix:

```
eval "$(/opt/homebrew/bin/brew shellenv)"
echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> ~/.zprofile
```

Error: "bw: command not found"

Cause: Bitwarden CLI not installed or not in PATH.

Fix:

```
brew install bitwarden-cli
hash -r # Refresh shell PATH cache
```

LaunchAgent fails to load

Symptoms: `launchctl bootstrap` returns error code 119, 122, or 125.

Debug:

```
# Check specific agent status:
launchctl print gui/$(id -u)/com.alai.litestream

# Look for "state = waiting" or "last exit code"

# Check agent logs:
tail -f ~/system/logs/litestream.log
tail -f ~/Library/Logs/com.alai.*.log
```

Common exit codes:

- 119: Invalid plist XML (malformed after sed replacement)
- 122: Path not found (e.g., /Users/makinja hardcoded but new user is /Users/alem)
- 125: Permission denied (env var secret not readable)

Secret rehydration failed

Symptoms: Bootstrap log shows "WARN: Bitwarden item 'X' not found".

Fix manually:

```
# Get secret from Bitwarden:
SECRET=$(bw get password groq-api-key --session $BW_SESSION)

# Edit plist:
vi ~/Library/LaunchAgents/com.alai.groq-model-benchmark.plist

# Replace BW:groq-api-key with actual value in <string> tag

# Reload:
launchctl bootout gui/$(id -u)/com.alai.groq-model-benchmark
launchctl bootstrap gui/$(id -u) ~/Library/LaunchAgents/com.alai.groq-model-benchmark.plist
```

Hardcoded /Users/makinja path mismatch

Cause: LaunchAgent plists contain hardcoded paths to /Users/makinja, but new Mac has different username (e.g., /Users/alem).

Fix (bulk replace):

```
NEW_USER=$(whoami)
cd ~/Library/LaunchAgents

for plist in com.alai.*.plist com.john.*.plist; do
    sed -i.bak "s|/Users/makinja|/Users/$NEW_USER|g" "$plist"
done

# Reload all:
launchctl bootout gui/$(id -u)
launchctl bootstrap gui/$(id -u) ~/Library/LaunchAgents/*.plist
```

Ollama models missing

Cause: Fresh install has no models cached. Models are NOT in git repos (too large).

Fix (pull from Ollama registry):

```
ollama pull qwen2.5-coder:32b
ollama pull llama3.3:70b
ollama pull deepseek-r1:32b
ollama pull deepseek-r1:70b
ollama pull devstral:24b
ollama pull mistral-small
ollama pull llama3.2-vision:90b
ollama pull qwq:32b

# Verify:
ollama list
```

Expected download size: ~150 GB total for all models. This takes 2-6 hours on good connection.

Database restore fails with "replica not found"

Cause: Azure credentials invalid, or DB was never replicated (new DB created after litemstream setup).

Debug:

```
# Test Azure auth:
az login --service-principal \
```

```
--username $AZURE_CLIENT_ID \  
--password $AZURE_CLIENT_SECRET \  
--tenant $AZURE_TENANT_ID  
  
# List backups:  
litestream snapshots -config ~/system/config/litestream.yml ~/system/databases/mission-control.db  
  
# Should show timestamps of snapshots in Azure Blob Storage
```

If no snapshots: DB is new or replication was broken. Accept data loss or restore from other source (e.g., Time Machine if on ANVIL).

Known Limitations

- **Untested end-to-end:** bootstrap.sh has NOT been tested on a completely fresh Mac. Code paths for Xcode install prompt, Homebrew first-run, and BW unlock flow are based on best practices but unverified in production DR scenario.
 - **User rename not handled:** If new Mac username != "makinja", LaunchAgent plists will fail due to hardcoded /Users/makinja paths. Manual sed replacement required (see Troubleshooting).
 - **npm install layer incomplete:** ~/system/tools/ contains 1,310 scripts, some requiring `npm install` in subdirs. bootstrap.sh does NOT auto-install these deps. Expect some tools to fail until deps are installed manually.
 - **Ollama models not in backup:** Models are fetched from Ollama registry on first use. Expect 2-6 hour delay to repopulate model cache (~150 GB).
 - **GitHub auth assumed:** Script assumes SSH keys or PAT for GitHub already configured. If not, git clone will prompt interactively.
 - **No Keychain sync:** macOS Keychain items (SSH keys, app passwords, etc.) are NOT part of this backup. Alem must re-enter credentials for Mail.app, Calendar, etc.
 - **No ~/felles or ~/Documents:** User data directories are NOT backed up by this system. Rely on Time Machine or iCloud for personal files.
-

Testing Recommendations

Before trusting this runbook in a real disaster:

1. Spin up a fresh Mac VM (UTM or Parallels) with macOS Sonoma
2. Run through Steps 1-6 end-to-end without looking at ANVIL

3. Verify LaunchAgent load count matches expected (~112)
4. Verify DB restore works for at least mission-control.db and hivemind.db
5. Document any new errors or missing secrets in this runbook

Assigned to: Petter Graff (CodeCraft) — MC task #8534

Last updated: 2026-04-20 | MC Task: #8534 | Tags: status=draft-untested, type=runbook, severity=critical

Revision #3

Created 2026-04-20 20:00:54 UTC by John

Updated 2026-07-05 20:05:05 UTC by John