Refine — Air-Gapped Setup Guide

For an architectural overview, see `NETWORK_ARCHITECTURE.md` (specifically Topology D).

Directory Integration in Air-Gapped Deployments

What: Refine supports LDAP / Active Directory in air-gapped deployments. Azure AD is **not supported** in air-gapped because it requires public internet egress to Microsoft endpoints.

Why:

- **LDAP / on-prem AD:** server-to-server bind. No public-internet calls. Works if your AD is reachable via VPN, Direct Connect, peering, or Transit Gateway.
- **Azure AD:** OAuth flow requires HTTPS to `login.microsoftonline.com` (or `.us` for Government), which is unreachable from a VPC with no NAT.

How:

- Configure LDAP via the in-app **Directory Settings** page after first login.
- Verify your VPC can route to your AD server (Direct Connect / peering / Transit Gateway).
- Verify DNS resolution for the AD hostname (Route 53 Resolver outbound endpoint to on-prem DNS, or use IP directly).
- See `LDAP_SETUP_GUIDE.md` for detailed configuration steps.

Same-Partition Limitation

A fully air-gapped Refine instance can monitor accounts ONLY in the same AWS partition. VPC endpoints cannot span partitions, and no public NAT egress is available.

If you need to monitor BOTH commercial AND GovCloud accounts, you have two options:

1. **Run Refine in a non-air-gapped VPC** (with NAT Gateway), use stored IAM keys for the cross-partition accounts. See `NETWORK_ARCHITECTURE.md` (Topology B).
2. **Run TWO Refine instances** — one per partition, both air-gapped. No cross-instance data sharing. Out of scope for v2.19.

Required VPC Endpoints

For Topology D (fully air-gapped GovCloud), create these VPC endpoints in your VPC before deploying the CF stack. Substitute your region for `us-gov-west-1` below.

Service	Type	Endpoint name	Used by
S3	Gateway	`com.amazonaws.us-gov-west-1.s3`	Delivery ZIP download, S3 reads from monitored accounts
STS	Interface	`com.amazonaws.us-gov-west-1.sts`	Cross-account assume-role
Lambda	Interface	`com.amazonaws.us-gov-west-1.lambda`	Refine invokes the customer's collector Lambda
SSM	Interface	`com.amazonaws.us-gov-west-1.ssm`	Admin access via Session Manager
SSM Messages	Interface	`com.amazonaws.us-gov-west-1.ssmmessages`	Session Manager data plane
EC2 Messages	Interface	`com.amazonaws.us-gov-west-1.ec2messages`	Session Manager data plane
CloudWatch Logs	Interface	`com.amazonaws.us-gov-west-1.logs`	Optional
ECR API	Interface	`com.amazonaws.us-gov-west-1.ecr.api`	Only if hosting the Refine image in your private ECR
ECR DKR	Interface	`com.amazonaws.us-gov-west-1.ecr.dkr`	Same

Cost: ~$7/mo per Interface endpoint. Gateway endpoints (S3) are free.

Custom AMI Build (Required for Topology D)

In a fully air-gapped VPC, the EC2 cannot reach `get.docker.com`, `awscli.amazonaws.com`, or any apt/dnf package repos. You must build a custom AMI with all required tools pre-installed.

Required tools in the AMI

- Docker Engine + systemd unit
- Docker Compose v2 plugin (`/usr/local/lib/docker/cli-plugins/docker-compose`)
- AWS CLI v2 (`/usr/local/bin/aws`)
- `python3`, `unzip`, `ca-certificates`, `curl`
- `amazon-ssm-agent` (already on AL2023; install via deb on Ubuntu)
- (Optional) The Refine Docker image pre-loaded via `docker load` so the EC2 can skip even the S3 download

Recommended base AMI

Amazon Linux 2023 — ships with `amazon-ssm-agent` pre-installed. Build steps:

```
# Launch a temporary EC2 from a stock AL2023 AMI in a non-air-gapped VPC
sudo dnf update -y
sudo dnf install -y docker python3 unzip awscli
sudo systemctl enable docker

# Docker Compose v2
sudo mkdir -p /usr/local/lib/docker/cli-plugins
sudo curl -SL https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86_64 \
  -o /usr/local/lib/docker/cli-plugins/docker-compose
sudo chmod +x /usr/local/lib/docker/cli-plugins/docker-compose

# Optional: pre-load the Refine image so the air-gapped EC2 skips the S3 download too
# (Only do this if you have the refine-image-*.tar from your delivery)
# sudo docker load -i refine-image-vX.Y.Z.tar

# Stop the instance and create an AMI from it
# Then copy that AMI to your air-gapped GovCloud account
```

In the CF stack, set `OperatingSystem=Custom` and `CustomAmiId=ami-xxx` (your hardened AMI). The UserData script in `aws-server-auto.yaml` v2.19+ skips installs when tools are already present, so it boots cleanly with no public-internet calls.

---

1. What's in the Package

File	Purpose
`setup.bat` / `setup.ps1`	Windows installer
`setup.sh`	Linux / Mac installer
`refine.license`	Your signed license file — **keep this private**
`docker-compose.yml`	Container configuration
`.env.example`	Configuration reference
`refine-image-*.tar`	Docker image (bundled for offline use)
`aws-setup.yaml`	AWS CloudFormation template (pre-filled with your info)
`CHANGELOG.md`	What changed in this version

You will also receive an **activation code** (`RF-XXXX-XXXX-XXXX-XXXX`) through a separate channel.

What You Get

Refine analyzes **14 AWS services** across all regions and provides:

- **Rightsizing recommendations** with ready-to-run CLI commands for EC2, RDS, and more

- **Exemptions** — mark resources as intentionally sized; exempt resources are excluded from future recommendations
- **Active Savings Plans tracking** — monitor utilization, coverage, and expiration across all plans with guardrails for underutilization
- **Savings Report** — unified view of realized and projected savings across your accounts
- **Cost Cleanup** — identify orphaned EBS snapshots, unused volumes, idle NAT gateways, and other waste
- **System Events** — audit log of syncs, exemptions, and configuration changes

All data stays in your AWS account. Refine has read-only access to an S3 bucket in your environment — no data leaves your network.

---

2. Prerequisites

Install these **in order** before running the setup wizard.

2a. WSL 2 (Windows only — required before Docker)

Open **PowerShell as Administrator**:

```
wsl --install
```

Restart your computer after WSL finishes installing.

2b. Docker Desktop

Platform	Download
Windows	https://desktop.docker.com/win/main/amd64/Docker%20Desktop%20Installer.exe
Mac — Apple Silicon (M1/M2/M3/M4)	https://desktop.docker.com/mac/main/arm64/Docker.dmg
Mac — Intel	https://desktop.docker.com/mac/main/amd64/Docker.dmg
Linux	https://docs.docker.com/desktop/install/linux/

After installing, **open Docker Desktop** and wait for the engine to fully start before proceeding.

> ***Air-gapped note:*** *Download Docker Desktop on a machine with internet, copy via USB, and install offline. Docker Desktop does not require internet after installation.*

2c. Python 3.8+

Platform	Download
Windows 64-bit	https://www.python.org/ftp/python/3.13.2/python-3.13.2-amd64.exe
Mac (Universal)	https://www.python.org/ftp/python/3.13.2/python-3.13.2-macos11.pkg
Linux	`sudo apt install python3` or `sudo dnf install python3`

> ***Windows:*** *Check* ***"Add Python to PATH"*** *on the first screen of the Python installer.*

---

3. First-Time Installation

Step 1: Extract the ZIP

Extract to a permanent location — this is where Refine lives.

- **Windows:** `C:\Refine`
- **Linux / Mac:** `/opt/refine` or `~/refine`

> *Do not extract to a temporary folder or move it after setup.*

Step 2: Run the Setup Wizard

Make sure Docker Desktop is running, then:

Windows — double-click `setup.bat`

Linux / Mac:

```
chmod +x setup.sh && ./setup.sh
```

The wizard will:

1. Verify Docker is running
2. Ask for your activation code
3. Prompt for an admin email and password (this becomes the ROOT account)
4. Ask for a port (default: 8000)
5. Generate a secure configuration (`.env` file)
6. Load the bundled Docker image (no internet needed)
7. Start Refine and open your browser

---

4. First Login

Open **http://localhost:8000** (or your custom port). Log in with the admin email and password you chose during setup.

You'll land on the **Setup** page to connect your AWS account. You are logged in as the **ROOT** user — the server administrator who can manage all users, groups, and AWS accounts.

> ***Note:*** *To add more users, go to* ***User Management*** *in the sidebar after connecting your first AWS account. See the User Guide — User Management for details.*

---

5. Connect Your AWS Account

Step 1 — Add Your AWS Account

Enter your **12-digit AWS Account ID** and select the region for your CloudFormation stack. Click **Add Account**.

Step 2 — Deploy CloudFormation

1. Click **Download CloudFormation Template** in Refine
2. In the **AWS Console** → **CloudFormation** → **Create Stack** → **Upload a template file**
3. Upload the template, name the stack (e.g., `blacktip-refine`), and click through
4. Check the IAM acknowledgment checkbox → **Create stack**
5. Wait for **CREATE_COMPLETE** (~2-3 minutes)

Step 3 — Copy Outputs into Refine

From the CloudFormation **Outputs** tab, copy these values into Refine:

CloudFormation Output	Refine Field
RoleArn	Role ARN
CostDataBucketName	S3 Bucket Name
RefineServiceAccessKeyId	Service Access Key ID
RefineServiceSecretAccessKey	Service Secret Access Key

> *The `RefineServiceSecretAccessKey` is only shown at stack creation. If you miss it, delete and recreate the stack.*
>
> ***Per-account credentials:** Each AWS account stores its own Access Key ID and Secret Access Key in Refine's database. There are no global AWS credentials in the `.env` file. Adding or changing credentials for one account does not affect any other account.*

Step 4 — Validate

Click **Validate Setup**. Refine tests IAM role assumption and S3 access. On success, an initial data sync triggers automatically.

Multiple AWS Accounts

Go to **AWS Accounts** in the sidebar → **Add Account**. Each account gets its own CloudFormation stack and its own set of service credentials — accounts are fully independent. Repeat Steps 1-4 for each additional account.

> ***GovCloud:** AWS GovCloud accounts are supported. Select the appropriate GovCloud region when adding the account.*

After Setup

- **Automatic sync:** Daily at 2 AM UTC (configurable), collecting 14 AWS services in parallel across all regions
- **Manual sync:** Click **Sync Now** on any page

- **Data stays in your AWS account** — Refine has read-only access to your S3 bucket

---

Multi-User Access

Refine supports multiple users on a single server. All AWS accounts are loaded onto one instance, and access is controlled through user roles and account groups.

- The ROOT user (the person who ran the setup wizard) creates users from the **User Management** page and assigns each user a role and one or more account groups.
- **Account groups** are named collections of AWS accounts (e.g., "Production Team", "Dev Team"). Users see only the accounts in their assigned groups.
- Three roles: **ROOT** (sees all accounts, manages all users and groups), **ADMIN** (manages users within their assigned groups, full actions within group accounts), and **USER** (full actions within group accounts, cannot manage users).

See the User Guide — User Management for step-by-step instructions.

---

6. Updating

When you receive a new delivery ZIP:

1. Stop Refine (optional): `docker compose down`
2. Extract the new ZIP **into the same folder**, overwriting when prompted
3. Re-run the setup wizard — it detects your existing config and enters **update mode**:
   - Skips activation code
   - Loads the new Docker image
   - Preserves all configuration and data

Updating CloudFormation

If the changelog mentions CloudFormation changes (such as new Lambda metrics added in v1.4.0), update your existing stack — do not delete and recreate it. Updating preserves your S3 bucket and all collected data.

The new `aws-setup.yaml` is included in your updated delivery ZIP.

1. Go to **AWS Console** → **CloudFormation** → select your stack → **Update**
2. Choose **Replace current template** → upload the new `aws-setup.yaml` from the delivery ZIP
3. Click through all pages — your existing parameters (ExternalId, Email, SyncSchedule) are preserved automatically
4. Acknowledge the IAM capabilities checkbox → **Update stack**
5. Wait for status **UPDATE_COMPLETE**

> ***Why update?*** *The v1.4.0+ Lambda collects EC2 running hours across 3-month and 1-year windows. Without a stack update, those columns in the EC2 table will be empty.*

Updating Your License

Replace `refine.license` and restart:

```
docker compose down && docker compose up -d
```

---

7. Backup & Recovery

What to Back Up

Item	Contains
`data/` folder	Database, VERSION file
`.env` file	Configuration, JWT secret
`refine.license`	Your signed license

Backup

```
cp -r data/ data-backup-$(date +%Y%m%d)/
cp .env .env.backup
cp refine.license refine.license.backup
```

Restore

```
cp -r data-backup-YYYYMMDD/ data/
cp .env.backup .env
cp refine.license.backup refine.license
docker compose down && docker compose up -d
```

Moving to a New Machine

1. Install Docker Desktop and Python on the new machine
2. Copy the entire Refine folder (including `data/`, `.env`, `refine.license`)
3. Load the image: `docker load -i refine-image-*.tar`
4. Start Refine: `docker compose up -d`

---

8. Troubleshooting

Problem	Solution
Refine won't start	Check Docker is running: `docker info`. Check logs: `docker compose logs --tail 50`
License expired/invalid	Contact Blacktip Solutions for a renewed `refine.license` file
Port already in use	Edit `PORT` in `.env` (e.g., `PORT=8080`), then restart
CloudFormation failed	Check the Events tab for details. Common: region not supported, missing IAM permissions
No data after sync	Check Lambda logs in AWS Console. Check S3 for `inventory/` folders. Click **Sync Now** in Refine

Securing Your Configuration

After first startup, restrict `.env` file permissions:

```
chmod 600 .env
```

Need Help?

Contact Blacktip Solutions at **support@blacktipsolutions.com**
