Refine — Deployment & Rollout Guide

Audience: IT / Cloud Operations team owning the deployment, and the executive sponsor approving it.

Purpose: A single document covering everything needed to plan, deploy, and roll out Refine inside your AWS environment — from architecture decisions, to costs, to step-by-step instructions, to onboarding your end-users.

At-a-glance

What is Refine? A self-hosted AWS cost-optimization platform. It runs entirely inside *your* AWS account, reads cost and inventory data from the AWS accounts you tell it to monitor, and shows your team where to save money. **No data leaves your AWS environment.**

What gets deployed? A single Linux EC2 instance running a Docker container, fronted by an Application Load Balancer with HTTPS. Plus per-monitored-account collector resources (an IAM role, an S3 bucket, and a Lambda function) deployed in each AWS account you want Refine to see.

Time to deploy: ~30-60 minutes for the first deployment. ~5-10 minutes per monitored AWS account onboarded thereafter.

Monthly cost (host infrastructure): ~$30 to ~$140 depending on size, exclusive of monitored-account costs (which are negligible — see Cost Breakdown).

Data residency: Everything stays in your AWS account. Refine has zero phone-home, no telemetry, no external services. License is verified locally with a public-key signature.

What Gets Created — Bare Minimum vs Customizable

The CloudFormation template `aws-server-auto.yaml` (included in your delivery package) creates everything below. Each row is marked as either **auto-created** (the template handles it for you) or **bring-your-own** (you point the template at an existing resource you already have).

Resource	Auto-created (default)	Bring-your-own option	Why you might bring your own
EC2 instance	Yes	Use `aws-server-support.yaml` (Topology F, manual deployment) instead	You require a specific golden AMI or instance configuration that doesn't fit the template
EBS root volume	Yes (50 GB, gp3)	Configurable size via `RootVolumeSize`	Compliance / sizing
EC2 IAM role + instance profile	Yes	Yes — set `CreateInstanceProfile=false` + provide `ExistingInstanceProfileArn`	Your security team requires pre-approved IAM roles with corporate boundary policies. The existing profile must include `AmazonSSMManagedInstanceCore` and `s3:GetObject` on the delivery bucket.
EC2 security group	Yes	Yes — set `CreateEc2SecurityGroup=false` + provide `ExistingEc2SecurityGroupId`	Existing baseline SG with corporate ingress / egress rules. The existing SG must allow inbound TCP 8000 from the ALB SG (HTTPS mode) or `AllowedAppCidr` (HTTP mode).
Application Load Balancer (ALB)	Yes (when HTTPS enabled)	Yes — set `CreateLoadBalancer=false` + provide `ExistingLoadBalancerArn`, `ExistingLoadBalancerListenerArn`, and `ExistingLoadBalancerHostHeader`	Existing corporate ALB with WAF / custom listeners / Web ACL. CF adds a host-header listener rule that forwards Refine traffic to a new target group.
ALB security group	Yes	Yes — set `CreateAlbSecurityGroup=false` + provide `ExistingAlbSecurityGroupId` (only valid when ALB is BYO)	Same — your ALB already has its own SG
ALB target group	Yes (always — one per Refine deployment)	n/a	n/a
ALB listener (HTTPS:443)	Yes (when CF creates the ALB)	BYO ALB: CF adds a *listener rule* to your existing listener instead	n/a
HTTP-to-HTTPS redirect listener	Yes (when CF creates the ALB)	n/a	If you BYO ALB, configure the redirect on your own listener
Elastic IP	Yes (when `EnableElasticIp=true`)	Configurable on/off	If you don't need a stable public IP, save $3.60/mo by setting to `false`
S3 bucket for delivery ZIP	Yes (when `UseExistingBucket=false`)	Yes — set `UseExistingBucket=true` and provide bucket name	If you have a corporate S3 governance template (encryption, lifecycle rules)
VPC	No — required input	Always BYO	You always provide the VPC ID

Resource	Auto-created (default)	Bring-your-own option	Why you might bring your own
Subnets (EC2 + ALB)	No — required input	Always BYO	You always provide subnet IDs
Route 53 / DNS records	No — you create after deploy	Always BYO	Your DNS provider may not be Route 53
ACM certificate	No — required input when HTTPS=true	Always BYO	ACM cert must be created in advance and DNS-validated
NAT Gateway (if private subnets)	No — you create separately	Always BYO	Internal-ALB topology needs NAT for the EC2 to reach AWS APIs
VPC Endpoints (if no NAT)	No — you create separately	Always BYO	Topology C / D — see Architecture Decisions
Per-monitored-account: Lambda + S3 + IAM role + IAM user	Yes (separate `aws-setup.yaml` template per account)	No	Standard onboarding flow

What you must have ready before starting

1. **VPC ID** in the AWS account where Refine will be hosted
2. **Subnet IDs** — at least one for the EC2, plus two more in different AZs for the ALB if using HTTPS (recommended)
3. **ACM certificate ARN** for HTTPS — create + DNS-validate this in advance (~10 min)
4. **A domain name** to point at the ALB (e.g., `refine.yourcompany.com`)
5. **Office or VPN CIDR** to allow into the ALB
6. **Optional: an SSH key pair** if you want SSH access (otherwise SSM Session Manager works without one)
7. **The delivery ZIP from Blacktip Solutions** (sent separately)
8. **The activation code** (sent in a separate email — required to start the container)

Architecture Decisions

You'll make four decisions before deploying. Each is a CloudFormation parameter; defaults are noted in **bold**.

Decision 1: Network topology — where do your users come from?

Choice	Best for	Topology
Internet-facing ALB	Commercial customer; users are remote / on home networks; auth gates access	A — public ALB + public subnet (default)
Internal ALB + VPN	Regulated / gov-leaning; users are on corporate VPN or Direct Connect; no public-internet exposure	B — internal ALB + private subnet + NAT
Internal ALB + VPC endpoints	Highest security; no public-internet egress allowed; same-partition monitoring only	C — internal ALB + private subnet + VPC endpoints
Air-gapped GovCloud	GovCloud customer; SCP blocks public-internet egress; custom AMI required	D — see `AIR_GAPPED_SETUP_GUIDE.md`

CloudFormation parameter: `LoadBalancerScheme` = `internet-facing` (default) or `internal`.

For Topology B/C/D, also set `AutoAssignPublicIp=false` to keep the EC2 off the public internet.

Decision 2: HTTPS — yes or no?

Choice	Best for
HTTPS enabled (default)	Production. Requires an ACM certificate ARN. ALB terminates TLS; EC2 stays on HTTP:8000 internally.
HTTPS disabled	Throwaway / smoke test only. EC2 directly exposes plaintext HTTP:8000. CloudFormation refuses to deploy this with `AllowedAppCidr=0.0.0.0/0` because plaintext credentials would be exposed to the internet.

Always pick HTTPS for any real deployment.

Decision 3: Server size — how many AWS accounts will you monitor?

AWS accounts	Recommended type	RAM	Monthly EC2 cost (us-east-1)
1-10	**`t3.medium`** (default)	4 GB	~$30
10-50	`t3.large` or `m6i.large`	8 GB	~$60-70
50-200	`m6i.xlarge`	16 GB	~$140
200+	`m6i.2xlarge`	32 GB	~$280

CloudFormation parameter: `InstanceType`.

Decision 4: SSH access — bring a key pair, or use SSM only?

Choice	Best for
SSM Session Manager only (recommended)	No SSH key needed. Connect via the AWS console "Connect" button. The browser shell uses outbound HTTPS through SSM Interface endpoints — works in private subnets too.
Bring an EC2 key pair	If your team's tooling assumes SSH. Set `KeyPairName` to your existing key pair name and `AllowedSshCidr` to your office/VPN CIDR.

CloudFormation parameter: `KeyPairName` — leave blank for SSM-only.

Cost Breakdown

All numbers are USD/month, us-east-1 commercial AWS. GovCloud is ~20% higher; other regions vary by ~10-15%.

Host infrastructure (the Refine server)

Component	Default Topology A	Topology B (private + NAT)	Topology C (private + VPC endpoints)
EC2 `t3.medium` (24/7)	$30.40	$30.40	$30.40
EBS gp3 50 GB	$4.00	$4.00	$4.00
Elastic IP (attached)	$3.60	$0 (no EIP needed)	$0
ALB	$16.20 base + $0.008/LCU-hr	$16.20 + LCUs	$16.20 + LCUs
Public IPv4 charge (AWS-wide since 2024)	$3.60	$0 (no public IP)	$0
NAT Gateway	$0	$32.40 + data	$0
VPC Interface endpoints (×6 to ×9)	$0	$0	$42-65
S3 delivery bucket (1 ZIP, ~100 MB)	<$0.01	<$0.01	<$0.01
Subtotal	**~$60/mo**	**~$85/mo**	**~$95/mo**

Per-monitored-account (each account you point Refine at)

Component	Cost
Lambda (24 invocations/day × ~30 sec)	<$0.10
S3 bucket (~10 MB inventory data)	<$0.01
CloudWatch Logs (30-day retention)	<$0.50
Subtotal per account	**~$0.50/mo**

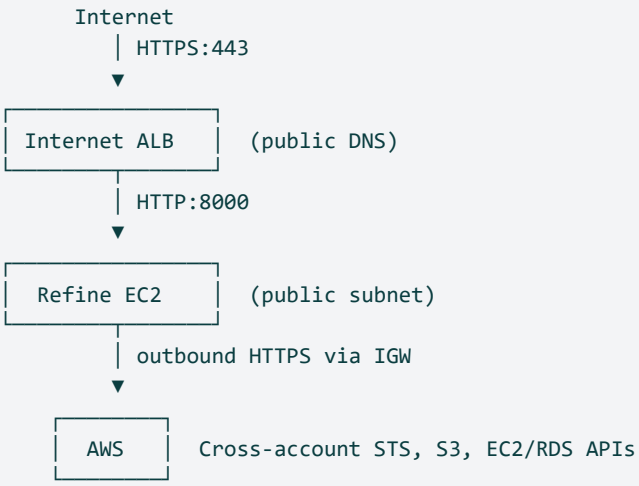
For 50 monitored accounts: ~$25/mo total across all of them.

What's NOT in the cost above

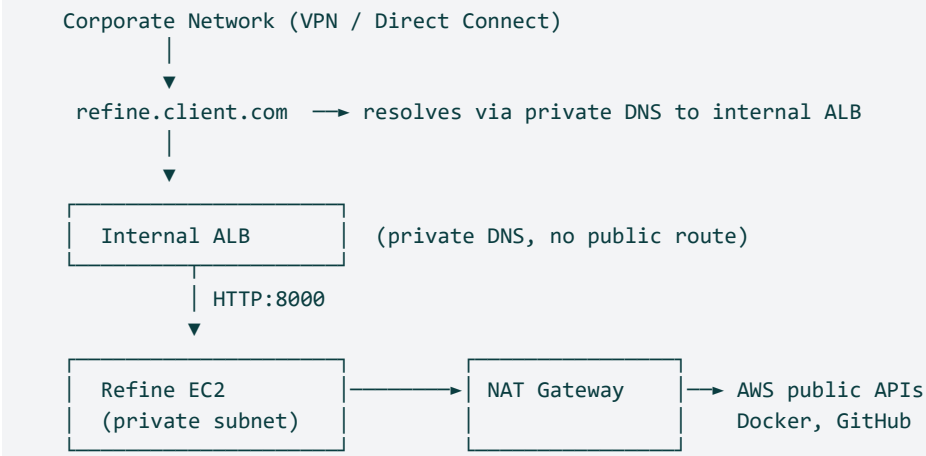
- Your existing VPC, subnets, NAT/IGW (if shared with other workloads — your cost allocation, not Refine-specific)
- Your ACM certificate (free)
- Your Route 53 hosted zone (~$0.50/mo per zone if you create one for Refine)
- The Refine license fee (separate from AWS infrastructure — see your contract with Blacktip Solutions)

Architecture Diagrams

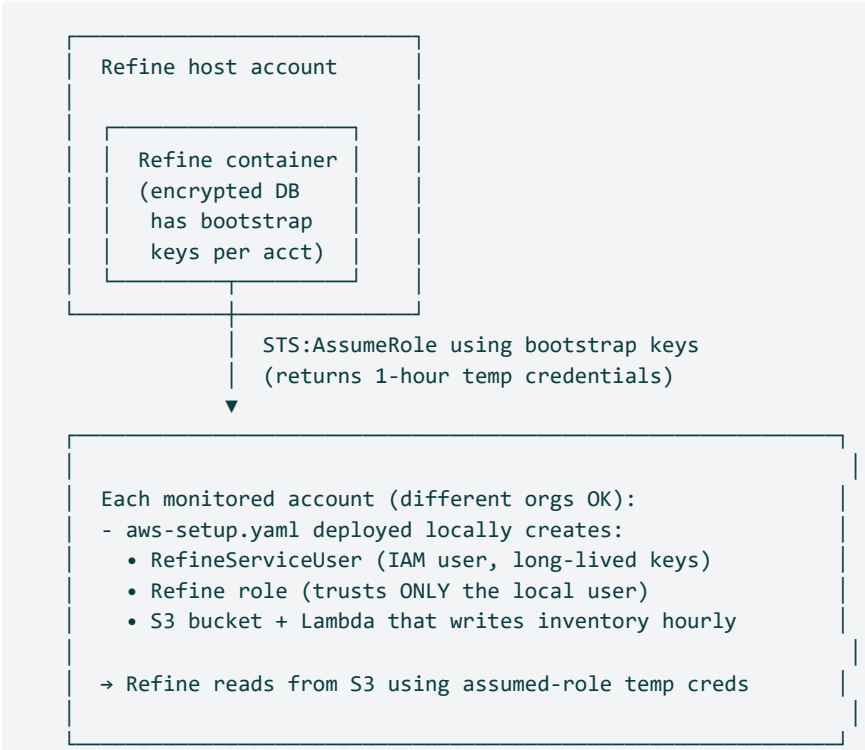
Topology A — Internet-facing (default for commercial)



Topology B — Internal ALB + VPN (recommended for regulated / multi-team)



Cross-account data flow



Important: the monitored-account role trusts an IAM user *inside its own account*, not the Refine host account. There is no cross-account trust to set up. This is why monitoring AWS accounts in different organizations works with the same flow.

For the full set of topology diagrams — including air-gapped GovCloud and cross-partition data flows — see `NETWORK_ARCHITECTURE.md`.

Bring Your Own Resources (Optional, Advanced)

By default the CloudFormation template creates everything it needs. For enterprise customers with corporate baselines (pre-approved IAM roles, hardened security groups, an existing ALB with WAF), you can point the template at existing resources instead.

Each option is independent — bring your own ALB but let CF create the IAM role, or vice versa. The defaults remain "create new" so a typical deployment doesn't see any of these parameters.

BYO IAM instance profile

When your security policy requires pre-approved IAM roles with corporate boundary policies / SCPs.

Parameter	Value
`CreateInstanceProfile`	`false`
`ExistingInstanceProfileArn`	`arn:aws:iam::123456789012:instance-profile/corp-refine-profile`

The existing profile must have:

- `AmazonSSMManagedInstanceCore` managed policy (for SSM Session Manager)
- `s3:GetObject` permission on the delivery ZIP key in the bucket Refine reads from
- `s3:ListBucket` on that bucket
- Outbound internet egress permitted by your VPC routing (or VPC endpoints — Topology C/D)

BYO EC2 security group

When your VPC has a baseline SG with corporate ingress / egress rules.

Parameter	Value
`CreateEc2SecurityGroup`	`false`
`ExistingEc2SecurityGroupId`	`sg-0123456789abcdef0`

The existing SG must allow:

- **HTTPS mode:** inbound TCP 8000 from the ALB SG (CF-created or BYO)
- **HTTP mode:** inbound TCP 8000 from `AllowedAppCidr`
- Outbound to AWS API endpoints (NAT, IGW, or VPC endpoints depending on topology)
- If you're using SSH: inbound TCP 22 from your office/VPN CIDR

CF will NOT add ingress rules to your existing SG — you manage the rules yourself.

BYO Application Load Balancer (most common BYO pattern)

When you have a corporate ALB with WAF, custom listeners, and want all corporate traffic — including Refine — to flow through it.

Parameter	Value
`CreateLoadBalancer`	`false`
`ExistingLoadBalancerArn`	`arn:aws:elasticloadbalancing:us-east-1:123:loadbalancer/app/corp-alb/abc...`
`ExistingLoadBalancerListenerArn`	`arn:...:listener/app/corp-alb/.../...` (your HTTPS:443 listener)
`ExistingLoadBalancerHostHeader`	`refine.yourcompany.com`
`ExistingLoadBalancerListenerRulePriority`	`100` (or any 1-50000 not already used)

What CF does instead of creating an ALB:

1. Creates a new target group `refine-tg-<stackname>` with the EC2 registered
2. Adds a listener rule to your existing HTTPS listener: when the host header equals `refine.yourcompany.com`, forward to the new target group
3. All other traffic on your listener is unaffected

What you do separately:

- Make sure your ALB's security group allows inbound 443 from your end-user CIDRs
- Make sure your DNS already points `refine.yourcompany.com` at your ALB
- The ACM certificate is already attached to your existing listener — no new cert needed

When you bring your own ALB, you typically also bring your own ALB security group (`CreateAlbSecurityGroup=false` + `ExistingAlbSecurityGroupId`) so the EC2 SG ingress on port 8000 references your existing ALB SG.

BYO ALB security group

Only valid when `CreateLoadBalancer=false` . CF doesn't create a new ALB, so it doesn't make sense to create a new SG for an ALB CF doesn't own.

Parameter	Value
`CreateAlbSecurityGroup`	`false`
`ExistingAlbSecurityGroupId`	`sg-0fedcba9876543210`

The EC2 security group's port-8000 ingress will reference this SG ID instead of a CF-created one.

Validation

CloudFormation Rules block invalid combinations at stack-creation time with clear error messages:

- `CreateInstanceProfile=false` requires `ExistingInstanceProfileArn` non-empty
- `CreateEc2SecurityGroup=false` requires `ExistingEc2SecurityGroupId` non-empty
- `CreateAlbSecurityGroup=false` requires `ExistingAlbSecurityGroupId` non-empty AND `CreateLoadBalancer=false`
- `CreateLoadBalancer=false` (with HTTPS enabled) requires `ExistingLoadBalancerArn` , `ExistingLoadBalancerListenerArn` , `ExistingLoadBalancerHostHeader` all non-empty

If you mix BYO toggles in incompatible ways, the stack creation aborts before any resources are created, and CF tells you exactly which parameter to fix.

Decision tree for BYO

```
Do you need to use a specific corporate IAM role?
  └─ Yes → CreateInstanceProfile=false + ExistingInstanceProfileArn

Do you need to use a specific corporate security group on the EC2?
  └─ Yes → CreateEc2SecurityGroup=false + ExistingEc2SecurityGroupId

Do you have an existing corporate ALB you must route through?
  ├─ Yes → CreateLoadBalancer=false + 4 Existing* fields
  │        Most likely also CreateAlbSecurityGroup=false + ExistingAlbSecurityGroupId
  │
  └─ No → leave defaults (CF creates a new ALB)
```

For a pilot deployment, we recommend starting with all defaults (CF creates everything). You can migrate to BYO resources later by updating the stack — CF will replace only the affected resources.

Pre-Deployment Checklist

Run through this with your AWS team before scheduling the deployment window.

1 — AWS account access

- ☐ You have admin access (or sufficient IAM permissions to deploy CloudFormation, create EC2/ALB/S3/IAM resources) in the account that will host Refine
- ☐ You're deploying in a region that supports all required services (any commercial AWS region; for GovCloud see `AIR_GAPPED_SETUP_GUIDE.md`)
- ☐ Your AWS account doesn't have a Free Tier-only restriction (Refine recommends `t3.medium`, which isn't free-tier-eligible)

2 — Networking

- ☐ You've picked your network topology (A, B, C, or D)
- ☐ VPC and subnets exist:
  - At least one EC2 subnet (public for A; private for B/C/D)
  - At least two ALB subnets in different AZs (HTTPS only)
- ☐ If Topology B: NAT Gateway is in a public subnet, with private-subnet routes to it
- ☐ If Topology C/D: VPC Interface endpoints for STS, Lambda, SSM, SSMMessages, EC2Messages are created; S3 Gateway endpoint is created
- ☐ DNS plan: which domain will point at the ALB (e.g., `refine.yourcompany.com`)
- ☐ You have your office or VPN CIDR ready (for `AllowedAppCidr` and `AllowedSshCidr`)

3 — Certificates

- ☐ ACM certificate created in the same region as the deployment
- ☐ Certificate covers the domain you'll point at the ALB
- ☐ Certificate is DNS-validated (status: Issued)

4 — Delivery package

- [ ] You've received the delivery ZIP from Blacktip Solutions
- [ ] You've received the **activation code** (sent in a separate email — keep it secure; you'll be prompted for it during setup)
- [ ] The license expiration date in the email matches your contract

5 — Stakeholders

- [ ] You've identified the **Root admin** (the person who'll be the Refine super-admin — this is the most powerful role in the system)
- [ ] You've identified any **Admin** users (who can manage other users and AWS accounts)
- [ ] You've identified the initial set of **end users** who'll get accounts (or planned to use directory integration to provision them automatically — see Directory Integration)

6 — Directory integration (optional, can be added later)

- [ ] Decided whether to use LDAP / Active Directory, Azure AD, or local accounts
- [ ] If LDAP: you have the server URL, bind DN, bind password, search base ready
- [ ] If Azure AD: you've registered an App in Entra ID and have the Tenant ID, Client ID, Client Secret ready
- [ ] If Azure Government: you know the customer is on Azure Gov tenant — Refine v2.19+ supports this via the Cloud dropdown

---

Deployment Walkthrough

This is a 30-60 minute sequence. Have the pre-deployment checklist complete before starting.

Step 1 — Extract the delivery ZIP

On your laptop, unzip `refine-CUST-XXX-vX.Y.Z.zip`. You'll see:

- `aws-server-auto.yaml` — main CloudFormation template (host server)
- `aws-server-support.yaml` — helper template (S3 + IAM + SG only, used in Topology F)
- `aws-setup.yaml` — per-monitored-account template (deployed later, in each account you want Refine to see)
- `setup.sh`, `setup.bat`, `setup.py` — **local install** wizard (Mac/Linux/Windows laptop with Docker Desktop). Use for laptop/workstation demos and the "local" deployment path.
- `start-refine.sh` — **EC2 server install** wizard, invoked by CloudFormation UserData on the EC2 (or manually if you SSH in). Use for the AWS hosted deployment path.
- `docker-compose.yml`, `.env.example` — container config
- `refine.license` — your signed license file
- `docs/` — full customer documentation, including this guide

Step 2 — Deploy `aws-server-auto.yaml`

1. AWS Console → CloudFormation → **Create stack** → **With new resources**
2. **Template source:** Upload a template file → pick `aws-server-auto.yaml`
3. **Stack name:** something like `refine-prod` or `refine-pilot`
4. **Parameters:** fill in based on your architecture decisions (defaults are sensible)
5. Click through Next → Next → review → check the IAM acknowledgment → **Create stack**

6. Wait for `CREATE_COMPLETE` (~3-5 minutes)

If anything fails, the **Events** tab shows the precise reason. Common issues:

- "AllowedAppCidr=0.0.0.0/0 in HTTP mode" → set `EnableHttps=true` or restrict the CIDR
- "AcmCertificateArn required when HTTPS=true" → fill in the ARN
- "Subnets must be in different AZs" → pick subnets in different AZs

Step 3 — Upload the delivery ZIP to the S3 bucket

Do this in parallel with stack creation (the EC2 will poll for the file and pick it up automatically).

1. AWS Console → **S3** → find the bucket named `refine-delivery-<stackname>-<accountid>`
2. **Upload** → drag in the entire delivery ZIP
3. **Rename the object** to exactly `refine-delivery.zip` (the EC2 looks for this exact name)

Step 4 — Connect to the EC2

Use **SSM Session Manager** (recommended — no SSH key needed):

1. EC2 console → Instances → click `refine-server-<stackname>`
2. **Connect** → **Session Manager** tab → **Connect**
3. You'll land in a browser shell as `ssm-user`. Switch to the EC2's user:

   ```
   sudo su - ec2-user
   ```

   (Use `ubuntu` if you picked `OperatingSystem=Ubuntu2204`.)

Or use SSH if you set a `KeyPairName`:

```
ssh -i <your-key>.pem ec2-user@<server-ip>
```

Step 5 — Run the setup wizard

```
./start-refine.sh
```

The wizard will:

1. Verify Docker, Compose, Python, AWS CLI are present (skips installs if pre-baked)
2. Download and extract the delivery ZIP if it hasn't already
3. Prompt for:
   - **Activation code** — paste the code you received separately
   - **Root admin email** — your designated super-admin's email
   - **Root admin password** — at least 12 characters
   - **Directory integration** — pick LDAP, Azure AD, or skip
4. Pull the Refine Docker image and start the container

When complete, you'll see:

```
URL      : http://localhost:8000
Username : <admin email>
Password : ************
Role     : Root (server administrator)
```

Step 6 — Configure DNS

Point your chosen domain at the ALB:

1. Get the ALB DNS name from CloudFormation stack outputs (`AlbDnsName`)
2. In your DNS provider, create a CNAME record: `refine.yourcompany.com` → `<alb-dns-name>`
3. Or in Route 53, create an alias A record pointing at the ALB
4. Wait 1-5 minutes for DNS to propagate

Step 7 — Open Refine in your browser

```
https://refine.yourcompany.com
```

You should see the Refine login page (HTTPS, certificate valid, browser shows the lock icon). Log in with the Root admin email and password you just set.

Step 8 — Verify the deployment

After login, you'll be at the Setup Onboarding page. Confirm:

- [ ] You're logged in as Root (top-right shows your email + "Root" badge)
- [ ] The dashboard loads with no $-amount data (empty state — this is correct for a fresh install)
- [ ] You can navigate to AWS Accounts, User Management, Settings without errors

If you see fake data ($87K/month, etc.), you have a stale demo build — contact Blacktip Solutions support.

Adding AWS Accounts

This is the per-account onboarding flow. You'll do it once per AWS account you want Refine to monitor.

From the Refine UI

1. **AWS Accounts → Add Account**
2. Enter:
   - **Account ID** (12-digit AWS account number)
   - **Account Name** (friendly name, e.g., "Production")
   - **Deployment Region** (where Refine's collector Lambda will run in that account — typically `us-east-1` for commercial, `us-gov-west-1` for GovCloud)
   - **Account Type** — `Commercial` or `GovCloud`
3. Click **Create**. Refine generates an External ID and shows you a "Launch Stack" link.

In the target AWS account

1. Click the "Launch Stack" link — opens AWS CloudFormation in that account
2. The template (`aws-setup.yaml` or `aws-setup-govcloud.yaml`) is pre-filled with the External ID. The **CustomerEmail** parameter is also pre-filled with the email you provided to Blacktip when ordering (used for sync-failure SNS alerts). Edit if you want notifications to go to a different address — e.g., a shared DL like `aws-alerts@yourcompany.com` .
3. Click through to create the stack (~2 minutes)
4. Open the stack's **Outputs** tab. Copy these five values:
   - `RefineServiceAccessKeyId`

- `RefineServiceSecretAccessKey` (NoEcho — click "Show" to reveal)
- The IAM Role ARN (e.g., `arn:aws:iam::123456789012:role/blacktip_refine-role-prod`)
- `CostDataBucketName`
- `CostDataBucketRegion`

Back in Refine

5. Paste the five values into the **Account Setup** modal
6. Click **Save & Validate**
7. Wait ~10 seconds — Refine assumes the role, lists the bucket, and confirms read access
8. You'll see a green check mark when the account is fully validated

Day-2 hardening (recommended after first validation)

The CloudFormation stack output included the secret access key. Even though it's masked with `NoEcho`, it's safest to rotate it once after first use:

1. AWS Console (in the monitored account) → IAM → Users → `RefineServiceUser-<stack>`
2. Security credentials → **Create access key**
3. Copy the new keys into Refine (AWS Accounts → click the account → **Update Credentials**)
4. Click **Test Connection** — confirm success
5. Back in IAM, deactivate the OLD access key
6. After 24 hours of confidence, delete the OLD key

Multi-User Setup

Refine has three roles. Pick them deliberately during rollout.

Role	Can do	Cannot do
Root	Everything: deploy, configure system settings, manage users + groups + AWS accounts, change directory integration, regenerate External IDs	n/a
Admin	Manage users + groups, add/remove AWS accounts, assign users to account groups	Change directory integration, modify Root settings, access platform admin
User	View dashboards, recommendations, savings reports for the AWS accounts in groups they're assigned to	Add/remove accounts, manage other users, see accounts they're not assigned to

Creating users

1. Log in as Root (or Admin)
2. **User Management** → **Add User**
3. Enter email, name, role
4. Optional: assign to an Account Group (see below)
5. Set initial password — share it with the user via your secure channel

Account groups (RBAC)

Account groups let you give Users access to specific AWS accounts only.

1. **User Management → Account Groups → Create Group**
2. Name the group (e.g., "EU Region", "Marketing Team", "Production")
3. Add the AWS accounts that belong in this group
4. Assign Users to the group

A User can be in multiple groups. They'll see data only for accounts in their groups. Admins and Roots see everything.

Bulk onboarding via directory integration

If you have LDAP / Active Directory or Azure AD, you can skip manual user creation and let Refine provision users automatically when they first log in. See Directory Integration.

Directory Integration

Refine can authenticate users against your existing LDAP / Active Directory or Azure AD / Entra ID. Set this up either during initial setup (`./start-refine.sh` prompts you) or later from **Directory Settings**.

Option 1 — LDAP / On-Prem Active Directory

Best for: Customers with existing Windows AD / OpenLDAP infrastructure.

What you need:

Field	Example
Server URL	`ldaps://ad.yourcompany.com:636`
Bind DN (service account)	`CN=svc-refine,OU=Service Accounts,DC=yourcompany,DC=com`
Bind password	(kept encrypted at rest in Refine)
User search base	`OU=Users,DC=yourcompany,DC=com`
User domain	`yourcompany.com`
TLS verification	Default **on** (CERT_REQUIRED)

Network requirements:

- Refine's EC2 must be able to reach your LDAP server on TCP 389 or 636
- The LDAP server can be a private IP — works in any topology
- VPC must resolve the LDAP hostname (Route 53 Resolver outbound endpoint to on-prem DNS, or use IP directly)

TLS verification: v2.19 defaults to verifying the LDAP server's TLS certificate against the system trust store. If your AD uses a self-signed corporate CA cert, either install the cert in the EC2's trust store, or set `tls_verify=false` in the LDAP config (with a security risk acknowledgment).

For details, see `LDAP_SETUP_GUIDE.md` .

Option 2 — Azure AD / Entra ID

Best for: Microsoft 365 customers, Azure-native organizations.

What you need:

Field	Where to get it
Azure Tenant ID	Entra ID → Overview
Client ID	The App Registration you create for Refine
Client Secret	Generated in the App Registration (kept encrypted at rest)
User domain	`yourcompany.com`
Cloud (commercial / **government** / china)	Pick "government" if your tenant is `*.onmicrosoft.us` (Azure Government)
Redirect URI	`https://refine.yourcompany.com/auth/azure/callback` (configure in App Registration)

Network requirements:

- Refine's EC2 must reach `login.microsoftonline.com` + `graph.microsoft.com` (or `.us` for Azure Gov) over public internet
- **This means Topology A or B (with NAT) only.** Topology C (no NAT, VPC endpoints only) and Topology D (air-gapped) cannot use Azure AD because Microsoft endpoints aren't reachable. Use LDAP instead in those topologies.

For details, see `AZURE_AD_SETUP_GUIDE.md`.

Option 3 — Local accounts only

Skip directory integration and create users manually in Refine's User Management. Fine for small teams (<10 users) or as a starting point — you can add directory integration later without losing users.

Group-based access via directory groups

Both LDAP and Azure AD support automatic group sync. Map a directory group (e.g., `refine-admins`) to a Refine Account Group, and any user in that directory group automatically gets access to the AWS accounts in that group when they log in.

Day-2 Operations

Updating Refine to a new version

Blacktip Solutions ships new versions periodically. To update:

```
cd /opt/refine
sudo docker compose pull
sudo docker compose down
sudo docker compose up -d
sudo docker logs -f refine-cust-XXX  # confirm "Application startup complete"
```

If the changelog notes `requires_cf_update: true`, also update your CloudFormation stack with the new template version (Blacktip Solutions will provide updated `aws-server-auto.yaml` if needed).

Rotating bootstrap IAM keys

Per AWS best practice, rotate the per-monitored-account IAM access keys every 60-90 days. The flow is in Adding AWS Accounts → Day-2 hardening.

Backing up the Refine database

The container's data volume (mounted at `/opt/refine/data`) contains:

- The Refine SQLite database (encrypted credentials, user accounts, account groups, directory config)
- License file
- Per-account inventory snapshots (regenerated by Lambdas anyway)

Snapshot the EC2's EBS root volume nightly via your existing backup tooling (AWS Backup, Veeam, etc.). Or, if you prefer container-level: `sudo cp /opt/refine/data/refine.db /backup/refine-$(date +%Y%m%d).db`.

Replacing a failed EC2

The EC2 is treated as a cattle, not a pet. To rebuild:

1. Snapshot the EBS root volume (if you want to keep current data)
2. Update the CloudFormation stack — change `InstanceType` or `OperatingSystem`, or just delete + redeploy
3. CloudFormation will replace the EC2; the new instance bootstraps from `aws-server-auto.yaml`'s UserData
4. Restore the data volume (if applicable) or re-onboard the AWS accounts

Monitoring the health of Refine itself

- **Health endpoint:** `GET /health` returns 200 if Refine is up. Hit it from your monitoring tool.
- **CloudWatch:** ALB target group health checks fire every 30 seconds. If Refine fails, the ALB returns 5xx and your monitoring catches it.
- **Container logs:** `sudo docker logs -f refine-cust-XXX` for live tailing.

Security & Data Handling

What data does Refine store?

Data	Where	Encrypted at rest?
Per-monitored-account IAM access keys (bootstrap)	Refine SQLite DB	Yes — Fernet (AES-128 + HMAC-SHA256)
LDAP / Azure AD credentials	Refine SQLite DB	Yes — same
User passwords (local accounts)	Refine SQLite DB	Yes — bcrypt hashed
Directory user passwords	Never stored	n/a
Inventory data (EC2 instances, RDS, costs)	Customer's own S3 buckets in monitored accounts	AES-256 (S3 SSE)
Activation code	Hashed (SHA-256) in license file	n/a

What data leaves your AWS account?

None. Refine has zero phone-home:

- License verification: purely local Ed25519 signature against an embedded public key
- No telemetry, no analytics, no Sentry / Datadog / Auth0
- The only HTTPX dependency is for customer-configured LDAP / Azure AD endpoints (which you control) — never to Blacktip-hosted endpoints

If your security team needs proof, the audit trail:

- `requirements.txt` shows only `httpx` (used for OAuth + LDAP), `boto3` (AWS), FastAPI, SQLAlchemy
- A grep across the codebase for `requests.post`, `httpx.post`, `urllib` returns only AWS API calls and customer-configured directory endpoints
- The Refine Docker image can be inspected: `docker run --rm ghcr.io/eichenbergerb/refine:latest sh -c 'find / -name "*.py" 2>/dev/null | xargs grep -l "post"'`

IAM permissions — what does Refine need?

On the host EC2's instance role (created by `aws-server-auto.yaml`):

- `s3:GetObject` on the specific delivery ZIP key only (least privilege)
- `s3:ListBucket` on the delivery bucket
- `ssm:UpdateInstanceInformation`, `ssmmessages:*`, `ec2messages:*` (for SSM Session Manager — managed via the AmazonSSMManagedInstanceCore policy)

In each monitored account (created by `aws-setup.yaml`):

- `RefineServiceUser` (IAM user) — has only `sts:AssumeRole` permission on the Refine role in the same account
- Refine role — read-only access to EC2, RDS, S3, EBS, NAT Gateway, Load Balancer, Lambda, CloudWatch Logs, Cost Explorer (commercial only), Fargate, ElastiCache, Lightsail, Data Transfer
- The role does NOT have any write permissions in the monitored account

Data flow trust model

The architectural choice that lets Refine work across orgs and partitions: each monitored account's Refine role trusts an IAM user inside that same account, NOT the Refine host account. There is no cross-account trust to set up between host and monitored. The host's identity never appears in any monitored account's CloudTrail.

Troubleshooting

Symptom	Likely cause	Fix
"Invalid rule description" on CF upload	Older template version	Re-extract the delivery ZIP — you may have a stale template
Stack creates but EC2 never reaches `CREATE_COMPLETE`	UserData script error	Check `/var/log/refine-userdata.log` on the EC2
`./start-refine.sh` fails with "unable to open database file"	Data directory permissions	`sudo chown -R 1000:1000 /opt/refine/data && sudo docker compose up -d`
Login page won't load	DNS not propagated yet	Wait 5 minutes; verify CNAME with `dig refine.yourcompany.com`
Login fails with "Invalid credentials"	Initial admin password not set or typo	Re-run `./start-refine.sh` after `sudo rm /opt/refine/.env`
"Service credentials not saved" on first Add Account submit	Pre-v2.19 — race between PATCH and validate	Update to v2.19+ (fixed)
Azure AD test connection fails with "Connection failed"	Network issue or wrong cloud type	v2.19+ shows specific error: "no internet" vs "wrong cloud" vs "bad credentials"
Dashboard shows fake $87K/mo data	Pre-v2.19 demo data leak in commercial install	Update to v2.19+ then `sudo rm -rf /opt/refine/data && ./start-refine.sh`

For other issues, gather:

```
sudo cat /var/log/refine-userdata.log     # bootstrap log
sudo docker compose ps                      # container status
sudo docker compose logs --tail=200         # app logs
```

And contact Blacktip Solutions support (see below).

Support & Escalation

Issue type	Contact
Setup / deployment questions	support@blacktipsolutions.com
Bugs / unexpected errors	support@blacktipsolutions.com — include the logs above
Sales / contract / license	sales@blacktipsolutions.com
Security / vulnerability disclosure	security@blacktipsolutions.com
GovCloud / air-gapped engagement	sales@blacktipsolutions.com — these typically need a paid engagement for first-customer validation

Response times (for active contracts):

- Setup blocker (you can't deploy): same business day
- Bug affecting production: 1 business day
- Feature request / enhancement: weekly product review

Rollout Plan — Recommended Sequence

For a pilot deployment, we recommend this 2-3 week sequence:

Week 1 — Single-team pilot

1. **Day 1:** Deploy `aws-server-auto.yaml`, run setup, log in as Root
2. **Day 2:** Onboard 1-3 AWS accounts your team owns directly. Verify cost data appears within ~1-2 hours
3. **Days 3-5:** Have your team use Refine for their own AWS accounts. Capture feedback on the UI and recommendations
4. **End of Week 1:** Decide whether to broaden the pilot

Week 2 — Broader pilot

1. **Day 6:** Configure directory integration (LDAP / Azure AD)
2. **Day 7:** Create Account Groups matching your team structure. Map directory groups to Account Groups for auto-provisioning
3. **Days 8-10:** Have a second team / department use Refine with their own AWS accounts
4. **Day 11-12:** Review with team leads — confirm permissions are scoped correctly and recommendations are actionable

Week 3 — Full rollout

1. **Day 13:** Onboard all production AWS accounts
2. **Day 14:** Schedule monthly review cadence with FinOps / Cloud Engineering
3. **Day 15:** First savings review meeting — Refine surfaces idle EC2, oversized RDS, unused EBS, etc.

Success criteria

You're done piloting when you can answer "yes" to all of these:

- [ ] All target AWS accounts are onboarded
- [ ] Cost data is current (refreshed within the last 24 hours)
- [ ] At least one team member has identified concrete savings from Refine's recommendations
- [ ] Your security team has reviewed the deployment and signed off on the data-handling posture
- [ ] You have a documented rotation schedule for the bootstrap IAM keys

---

Appendix A — CloudFormation Parameters Reference

Full list of `aws-server-auto.yaml` parameters with defaults.

Server Setup

- `InstanceType` — `t3.medium` (default). Allowed: `t3.micro` / `t3.small` (smoke test only), `t3.medium` / `t3.large` / `m6i.large` / `m6i.xlarge` / `m6i.2xlarge`

- `KeyPairName` — empty (default; uses SSM)
- `RootVolumeSize` — `50` GB (default), min 30, max 500
- `EnableElasticIp` — `true` (default)
- `AutoAssignPublicIp` — `subnet-default` (default), `true`, `false`

Operating System

- `OperatingSystem` — `Ubuntu2204` (default), `AmazonLinux2023`, `Custom`
- `CustomAmiId` — required when `OperatingSystem=Custom`

Network

- `VpcId` — required (your VPC ID)
- `SubnetId` — required (EC2 subnet)
- `AllowedSshCidr` — your office/VPN CIDR (only if `KeyPairName` is set)
- `AllowedAppCidr` — your office/VPN CIDR (or `0.0.0.0/0` for HTTPS production)

HTTPS / Load Balancer

- `EnableHttps` — `true` (default)
- `AcmCertificateArn` — required when `EnableHttps=true`
- `LoadBalancerSubnetIds` — required when `EnableHttps=true` (≥2 subnets in different AZs)
- `LoadBalancerScheme` — `internet-facing` (default), `internal`

Bring Your Own (Advanced — leave defaults for typical deployments)

- `CreateInstanceProfile` — `true` (default), `false` to provide existing
- `ExistingInstanceProfileArn` — required when `CreateInstanceProfile=false`
- `CreateEc2SecurityGroup` — `true` (default), `false` to provide existing
- `ExistingEc2SecurityGroupId` — required when `CreateEc2SecurityGroup=false`
- `CreateAlbSecurityGroup` — `true` (default), `false` to provide existing (only valid with `CreateLoadBalancer=false`)
- `ExistingAlbSecurityGroupId` — required when `CreateAlbSecurityGroup=false`
- `CreateLoadBalancer` — `true` (default), `false` to use existing corporate ALB
- `ExistingLoadBalancerArn` — required when `CreateLoadBalancer=false`
- `ExistingLoadBalancerListenerArn` — required when `CreateLoadBalancer=false` (your HTTPS:443 listener)
- `ExistingLoadBalancerHostHeader` — required when `CreateLoadBalancer=false` (e.g., `refine.yourcompany.com`)
- `ExistingLoadBalancerListenerRulePriority` — `100` (default; any 1-50000 not in use)

Delivery Package

- `UseExistingBucket` — `false` (default)
- `ExistingBucketName` — used when `UseExistingBucket=true`
- `ExistingObjectKey` — `refine-delivery.zip` (default)

Tagging

- `EnvironmentTag` — `production` (default), `staging`, `dev`, `test`
- `CustomerName` — your internal team / customer label

Appendix B — FAQ

Q: Can we run Refine in a different AWS account from our monitored accounts? A: Yes. That's the standard pattern. The host account just needs network egress to AWS APIs. The monitored account trust policies don't reference the host account at all.

Q: Can we monitor AWS accounts from different AWS Organizations? A: Yes. Each account owner deploys `aws-setup.yaml` in their account, then sends the resulting access keys back to the Refine admin. No org-level trust required.

Q: Can we monitor commercial + GovCloud accounts from the same Refine instance? A: Yes, in non-air-gapped topologies (A or B with NAT). Refine routes STS calls to the right partition based on the role ARN. Each account uses its own bootstrap keys (commercial keys for commercial accounts, GovCloud keys for GovCloud accounts). Air-gapped (Topology D) cannot reach the other partition.

Q: How long until cost data appears after onboarding an account? A: ~1 hour for the first Lambda run, then hourly thereafter. The dashboard refreshes on page load.

Q: Does Refine require a Cost Explorer subscription? A: Yes for commercial AWS — Cost Explorer must be enabled in each monitored account (it's free to enable; daily rates may apply for API calls but are typically <$1/mo per account). GovCloud doesn't have Cost Explorer; Refine gracefully skips it for GovCloud accounts.

Q: Can we restrict which user roles see which dashboards? A: Yes. Account Groups + roles control this. Users see only the AWS accounts in groups they're assigned to. Admins manage groups; Root manages everything.

Q: What happens if our license expires? A: Refine logs a warning but continues running. New deployments require a valid license. Contact Blacktip Solutions before your expiration date to renew.

Q: Can we self-host the Refine Docker image (private registry)? A: Yes — see `AIR_GAPPED_SETUP_GUIDE.md` for the private-ECR pattern.

Q: Can we deploy multiple Refine instances? A: Yes — each is independent and has its own license. Common pattern: separate Refine for prod vs staging environments.

Q: How do we decommission Refine? A: Delete the CloudFormation stack. The S3 delivery bucket has `DeletionPolicy: Retain` to prevent accidental data loss — empty and delete it manually after the stack is gone. In each monitored account, delete the `aws-setup.yaml` stack to remove the IAM role + Lambda + S3.

Document version: aligned with Refine v2.20. For older deployments, see release-specific docs.
