Network Architecture

This is the canonical reference for **how Refine sits in your network** — where the server lives, how end users reach it, and how it pulls data from monitored AWS accounts. All other customer docs link here for topology questions.

Overview — what Refine is, network-wise

What: Refine is a single Linux EC2 (`t3.medium` is the default size) running a Docker container, optionally fronted by an Application Load Balancer with HTTPS, deployed entirely inside your AWS account. It pulls inventory + cost data from AWS accounts you tell it to monitor.

Why: All of your data — login credentials, cost numbers, AWS account IDs — stays inside your AWS organization. Blacktip never sees any of it. The Refine codebase has zero phone-home: no telemetry, no version-check API calls, no Sentry/Datadog/Auth0 SDKs. License verification is purely local Ed25519 signature checking against a public key baked into the Docker image.

How: You pick one of the topologies below based on your security posture and where your end users are.

Decision tree — pick a topology

```
1. Where are your users?
   ├─ All on the public internet (or a few office IPs)
   │  → Topology A: Internet-Facing ALB + Public Subnet
   │
   ├─ All on corporate VPN / Direct Connect / inside the VPC
   │  → Topology B (with NAT) or Topology C (with VPC endpoints)
   │
   └─ Mixed (some public, some VPN)
      → Recommended: route everyone through the VPN — Topology B or C.
        Public users connect to your VPN, then hit the same internal URL
        as everyone else. One DNS, one auth flow, one network posture.

2. Does your AWS account block public internet egress (gov / SCP-restricted)?
   ├─ No (typical)
   │  → Topology A or B (both rely on internet egress for Docker pull,
   │    AWS CLI, GitHub, etc.)
   │
   └─ Yes
      → Topology D: Air-Gapped GovCloud (custom AMI + offline image bundle +
        full set of VPC endpoints; only same-partition monitoring possible)

3. Do you already have a corporate ALB you must put traffic through?
   ├─ No
   │  → Topology A, B, C, or D as above
   │
   └─ Yes
      → Topology E: Reusing an Existing Corporate ALB (in planning — not yet available; use Topology F too

4. Do you want to bring your own EC2?
   → Topology F: Manual EC2 + Helper Stack (use aws-server-support.yaml)
```

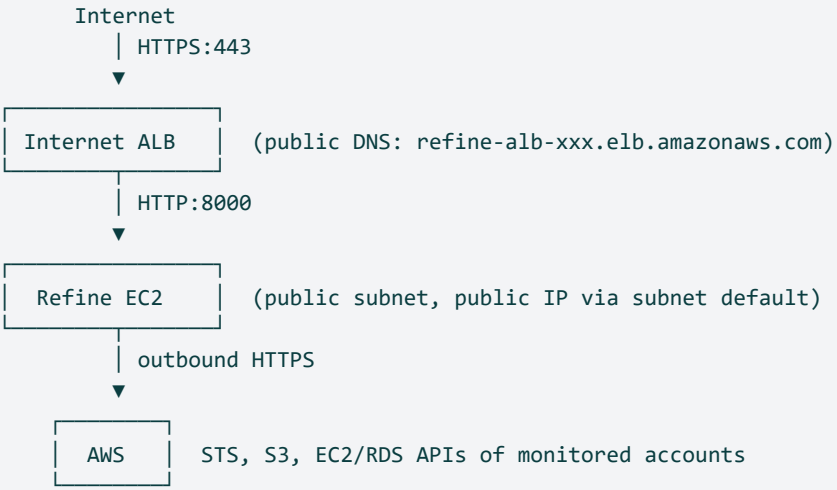
Topology A — Internet-Facing ALB + Public Subnet

What: EC2 in a public subnet, ALB with a public DNS name, HTTPS via ACM.

Why: Default for commercial customers. End users from anywhere on the internet can reach it (subject to `AllowedAppCidr`). Auth (LDAP / Azure AD / local password) gates access.

How: CloudFormation parameters:

Parameter	Value
`LoadBalancerScheme`	`internet-facing` (default)
`EnableHttps`	`true`
`AcmCertificateArn`	your ACM cert ARN (must be in this region)
`LoadBalancerSubnetIds`	two **public** subnets in different AZs
`SubnetId`	a **public** subnet (must have route to internet gateway)
`AutoAssignPublicIp`	`subnet-default`
`AllowedAppCidr`	`0.0.0.0/0` is OK (HTTPS protects in transit), or restrict



Limitations: the EC2 has a public IP. If your security policy forbids public-IP EC2s, use Topology B or C.

Topology B — Internal ALB + Private Subnet + NAT

What: Internal-only ALB (private DNS), EC2 in a private subnet, NAT Gateway provides outbound egress.

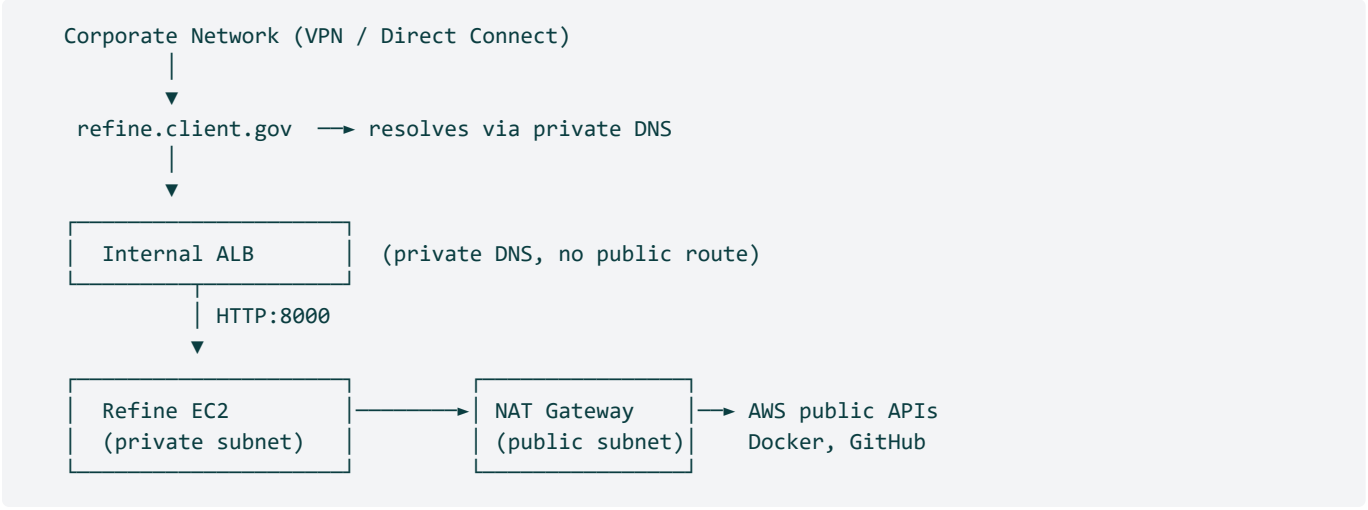
Why: Recommended for most regulated commercial / gov-leaning customers. Server is not reachable from the public internet at all. End users connect to your corporate VPN / Direct Connect, then hit a private DNS name (e.g., `refine.client.gov`) that resolves to the internal ALB.

How:

Parameter	Value
`LoadBalancerScheme`	`internal`
`EnableHttps`	`true`
`AcmCertificateArn`	your ACM cert ARN
`LoadBalancerSubnetIds`	two **private** subnets in different AZs
`SubnetId`	a **private** subnet (with NAT route to 0.0.0.0/0)
`AutoAssignPublicIp`	`false`
`AllowedAppCidr`	your VPN/VPC/peered-VPC CIDR (e.g., `10.0.0.0/16`)

VPC requirements you set up separately:

- A NAT Gateway in a public subnet (for the EC2's outbound internet)
- The private subnets route `0.0.0.0/0` → the NAT Gateway
- A Route 53 private hosted zone (or corporate DNS) maps `refine.client.gov` → ALB DNS



Limitations: NAT Gateway costs ~$32/mo + data-processing charges. Cross-partition monitoring works because NAT egresses to public AWS APIs of any partition.

Topology C — Internal ALB + Private Subnet + VPC Endpoints (No NAT)

What: Same as Topology B but no NAT Gateway. The EC2 reaches AWS APIs via VPC Interface/Gateway endpoints. Public-internet endpoints (Docker Hub, GitHub) are unreachable, so you must use a custom AMI with all tools pre-installed.

Why: Highest-security commercial / regulated customers whose security policy forbids any public-internet egress, but who don't need air-gapped GovCloud. Same-partition monitoring only.

How: Same CF parameters as Topology B, except no NAT Gateway is needed. Plus build a custom AMI per `AIR_GAPPED_SETUP_GUIDE.md` and pass `OperatingSystem=Custom` + `CustomAmiId=ami-xxx`.

VPC endpoints required (Interface unless noted):

Service	Endpoint name
S3 (Gateway)	`com.amazonaws.<region>.s3`
STS	`com.amazonaws.<region>.sts`
Lambda	`com.amazonaws.<region>.lambda`
SSM	`com.amazonaws.<region>.ssm`
SSM Messages	`com.amazonaws.<region>.ssmmessages`
EC2 Messages	`com.amazonaws.<region>.ec2messages`
CloudWatch Logs	`com.amazonaws.<region>.logs` (optional)

Cost: ~$7/mo per Interface endpoint × 6 endpoints = ~$42/mo (less than NAT for low-data deployments).

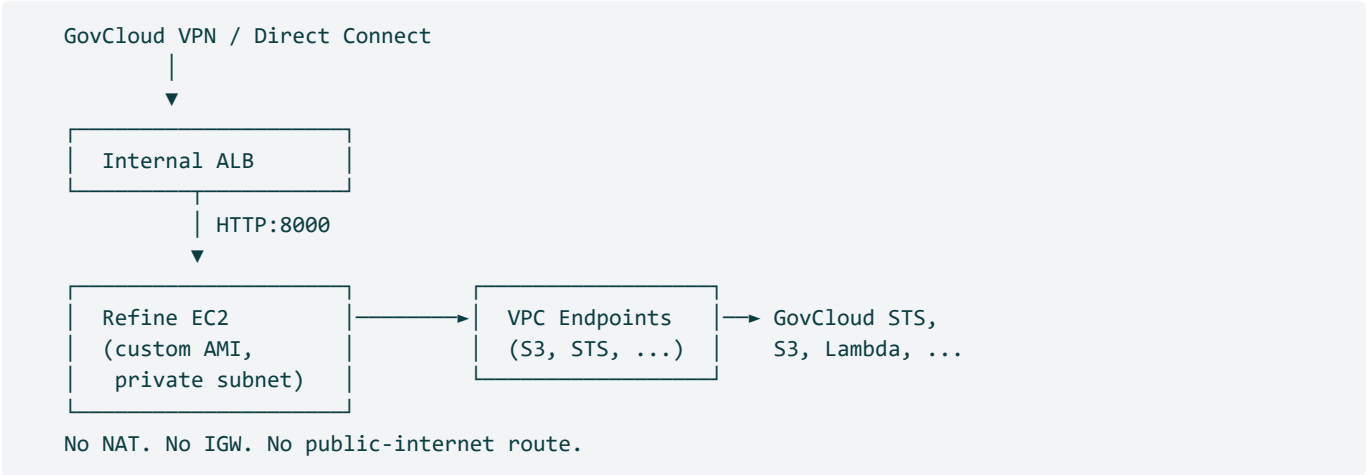
Limitations: same partition only — VPC endpoints can't span commercial ↔ GovCloud. If you have monitored accounts in the other partition, use Topology B instead.

Topology D — Air-Gapped GovCloud

What: Refine in GovCloud, no NAT, no IGW, only VPC endpoints. Custom AMI with Docker / Compose / AWS CLI / Python pre-baked. Offline Docker image bundle (`refine-image.tar`) loaded via `docker load` .

Why: The hardest gov posture: SCP blocks all public-internet egress. Refine's codebase already has zero phone-home, so this works — but every step needs preparation.

How: See `AIR_GAPPED_SETUP_GUIDE.md` for the complete walkthrough.



Limitations:

- Same-partition only (cannot reach commercial AWS APIs)
- Cost Explorer is unavailable in GovCloud — Refine skips cost data for `aws-us-gov` partition accounts (gracefully)
- Azure AD does not work (requires `login.microsoftonline.com` / `.us` over public internet) — use LDAP

Topology E — Reusing an Existing Corporate ALB

Status: In planning — not yet available. The BYO-ALB parameters described in earlier design notes are not yet implemented in `refine-server-auto.yaml`. If your security policy requires routing through your existing corporate ALB (with WAF, custom listeners, Web ACL), use **Topology F** today — let your team manage both the ALB and the EC2, with `aws-server-support.yaml` creating only the helper resources (S3 bucket, IAM role, security group).

Topology F — Manual EC2 + Helper Stack

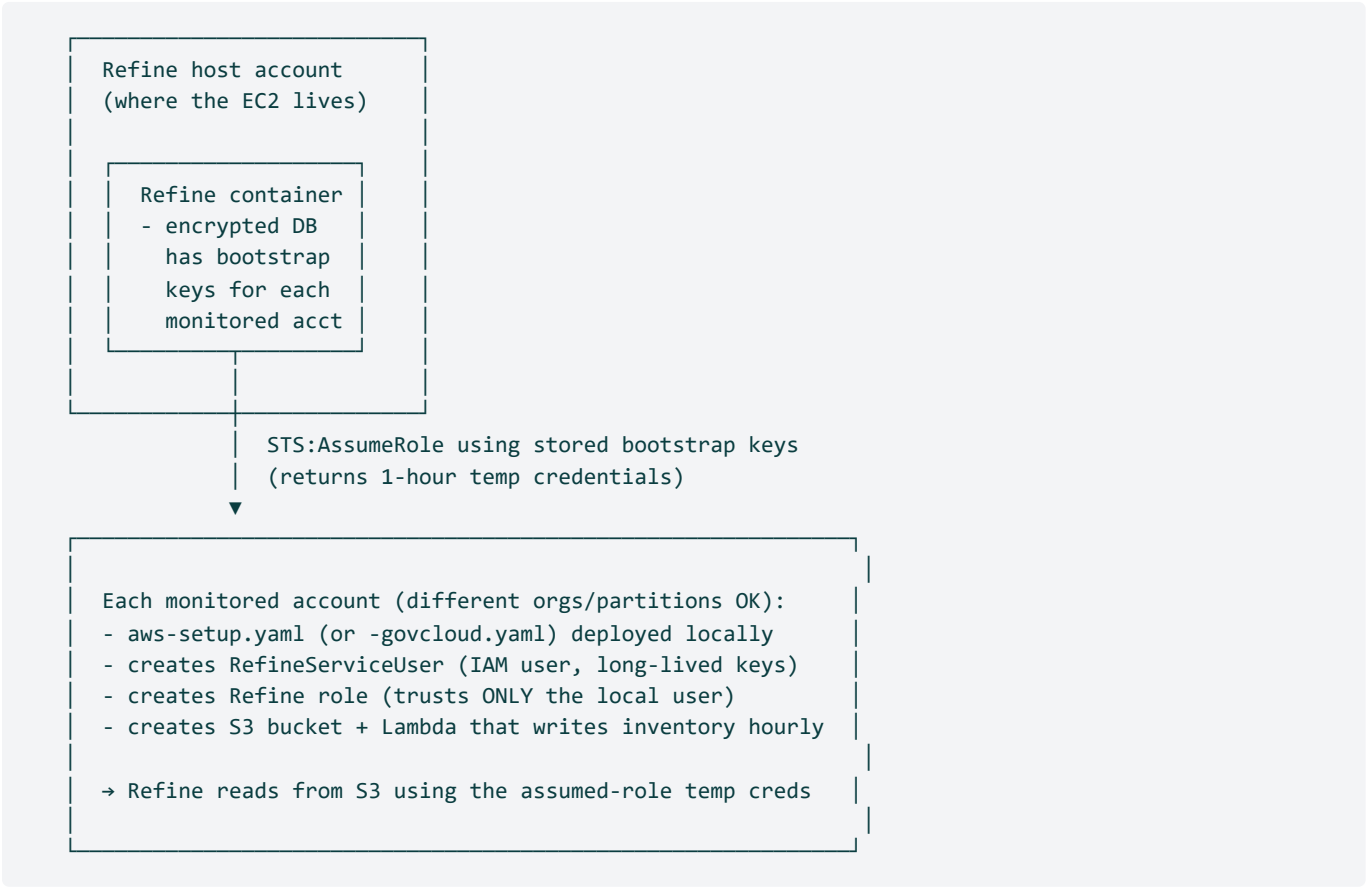
What: You launch and manage your own EC2 (custom AMI, hardened image, golden image). Refine's helper template (`aws-server-support.yaml`) creates only the supporting AWS resources: S3 delivery bucket, IAM role, security group.

Why: Highest customization. You handle the EC2 lifecycle, networking, and Refine install yourself.

How:

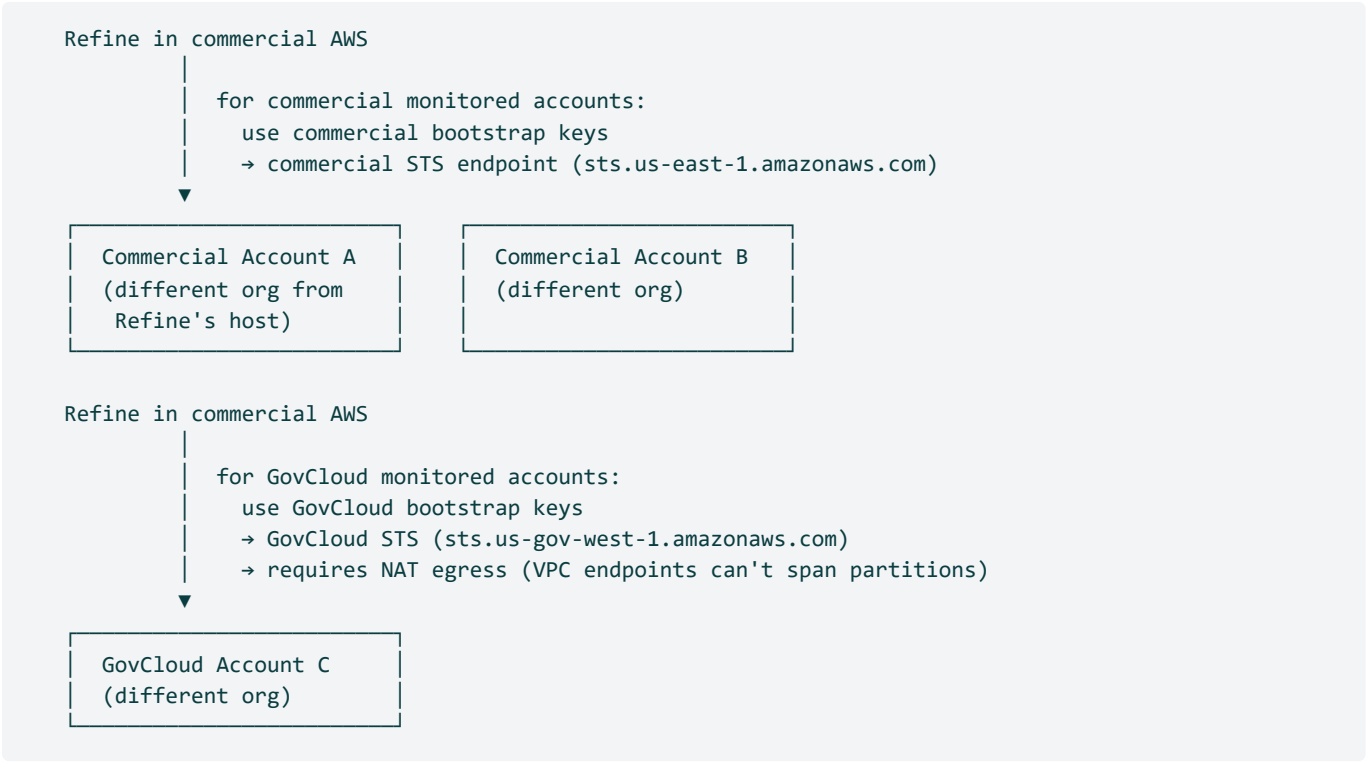
1. Deploy `aws-server-support.yaml` → outputs: bucket name, IAM instance profile ARN, security group ID
2. Launch your own EC2 with: that instance profile attached, that security group attached, internet egress (or VPC endpoints + custom AMI for air-gapped)
3. Upload the delivery ZIP to the bucket
4. SSH/SSM into your EC2, `aws s3 cp s3://<bucket>/refine-delivery.zip /opt/refine/` , unzip, run `./setup.sh`

Cross-account data flow — how Refine reads from monitored accounts



Key fact: the monitored account's role trusts an IAM user **inside its own account**, NOT the Refine host account. There is no cross-account trust to set up between host and monitored. This is why mixed orgs and mixed partitions both work with the same onboarding flow.

Cross-partition data flow — commercial + GovCloud accounts



Setup is identical for every account regardless of partition:

1. Account owner clicks "Add Account" in Refine
2. Refine generates an External ID and a CloudFormation template URL specific to their partition (commercial or GovCloud)
3. Owner deploys the stack in their account → outputs `RefineServiceAccessKeyId` + `RefineServiceSecretAccessKey`
4. Owner pastes those keys into Refine's "Add Account" form
5. Refine encrypts them at rest, uses them to call STS in the right partition, gets 1-hour temp creds, reads inventory from the local S3 bucket

Air-gapped + cross-partition is NOT possible. No VPC endpoint can reach the other partition's APIs. If you need both partitions, you need NAT egress (Topology B), not air-gapped (Topology D).

Directory integration support per topology

Topology	LDAP / on-prem AD	Azure AD (commercial)	Azure AD (Government)
A: Internet-Facing	✓ if AD reachable	✓	✓ (set Cloud=government in Refine)
B: Internal + NAT	✓ if AD reachable	✓ (egresses via NAT)	✓
C: Internal + VPC Endpoints	✓ if AD reachable	✗ (no internet)	✗ (no internet)
D: Air-Gapped GovCloud	✓ if on-prem AD reachable via VPN/peering	✗ (requires internet)	✗ (requires internet)
E: Existing Corporate ALB	depends on EC2 subnet	depends on EC2 subnet	depends on EC2 subnet
F: Manual EC2	depends on EC2 setup	depends on EC2 setup	depends on EC2 setup

For details, see `LDAP_SETUP_GUIDE.md` and `AZURE_AD_SETUP_GUIDE.md`.

Day-2 operations

Adding a new monitored account

Same flow regardless of partition:

1. In Refine: AWS Accounts → Add Account → enter the AWS account ID and select the region/partition
2. Refine displays a "Launch Stack" link to the partition-appropriate CloudFormation template
3. The owner of the target account deploys the stack
4. Stack outputs: `RefineServiceAccessKeyId`, `RefineServiceSecretAccessKey`, role ARN, bucket name
5. Owner pastes those values into the Refine modal and clicks "Save & Validate"
6. Refine confirms it can assume the role and read from the bucket

Rotating bootstrap keys (recommended every 60-90 days)

Long-lived IAM access keys are flagged by AWS Security Hub / IAM Access Analyzer. Rotate per account:

1. In the monitored account's IAM console: Users → `RefineServiceUser-<stack>` → Security credentials → **Create access key** (you'll temporarily have 2 active keys — AWS limit)
2. Copy the new Access Key ID + Secret
3. In Refine: AWS Accounts → click the account → **Update Credentials** → paste new keys → Save
4. Click **Test Connection** — confirm the new keys work
5. Back in IAM: deactivate the OLD access key (don't delete yet)
6. After 24 hours of confidence: delete the old key

Day-2 hardening immediately after first validation: rotate once. The CF-emitted secret in stack outputs is invalidated, even if someone has cached the value.

Replacing a failed EC2

The EC2 is stateless — all per-account credentials are stored in the Refine SQLite DB, which is in `/opt/refine/data/`. Persist that directory:

- **In-place upgrade:** `docker compose pull && docker compose down && docker compose up -d` — preserves the volume
- **EC2 replacement:** snapshot the EBS root volume, restore on the new instance, mount at `/opt/refine`, restart

For a clean rebuild (e.g., compromised host), tear down the stack and rebuild — customers will need to re-paste their per-account keys.

Migrating from Topology A to Topology B (going from public to private)

1. Create the NAT Gateway + private subnets if not already present
2. Update the CF stack with new parameters: `LoadBalancerScheme=internal`, `AutoAssignPublicIp=false`, new `SubnetId`, new `LoadBalancerSubnetIds`
3. **CloudFormation will replace the EC2** because `NetworkInterfaces` is now set differently. Plan for ~5 minutes of downtime
4. Set up Route 53 private hosted zone or corporate DNS to point `refine.client.gov` at the new internal ALB
5. Update your VPN access lists if needed

VPC endpoints reference

Required when using Topology C or D. Create these in your VPC before deploying the CF stack.

Service	Type	Endpoint name (example)	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, for forwarding container logs
ECR API	Interface	`com.amazonaws.us-gov-west-1.ecr.api`	Only if using a private ECR for the Refine image
ECR DKR	Interface	`com.amazonaws.us-gov-west-1.ecr.dkr`	Only if using a private ECR for the Refine image

Replace `us-gov-west-1` with your region.
