Refine — IAM Permissions Reference

Purpose: AWS Marketplace reviewers and customer security teams need to know exactly what IAM permissions the Refine CloudFormation stack creates in the customer's account. This document is the authoritative reference.

Refine is **read-only by design.** Across both the host stack (Marketplace-deployed) and the per-account onboarding stack (added once per AWS account being monitored), Refine never requests, accepts, or uses write permissions on customer cloud resources.

Stack A — Marketplace Host Stack (`refine-marketplace-cfn.yaml`)

This is the stack the customer launches from the AWS Marketplace listing. It provisions the Refine EC2 host in the customer's account.

IAM resources created

Resource	Type	Purpose
`RefineInstanceRole`	`AWS::IAM::Role`	Attached to the Refine EC2 instance. Grants only the minimum needed to poll the delivery bucket and run SSM Session Manager.
`RefineInstanceProfile`	`AWS::IAM::InstanceProfile`	Wraps the role for EC2 attachment.

The customer may bring their own existing instance profile via the `ExistingInstanceProfileArn` parameter, in which case the CFN creates **no** IAM resources.

Inline policy attached to `RefineInstanceRole`

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ReadDeliveryBucket",
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::<delivery-bucket>",
        "arn:aws:s3:::<delivery-bucket>/*"
      ]
    },
    {
      "Sid": "SSMSessionManager",
      "Effect": "Allow",
      "Action": [
        "ssm:UpdateInstanceInformation",
        "ssmmessages:CreateControlChannel",
        "ssmmessages:CreateDataChannel",
        "ssmmessages:OpenControlChannel",
        "ssmmessages:OpenDataChannel",
        "ec2messages:GetMessages",
        "ec2messages:AcknowledgeMessage",
        "ec2messages:DeleteMessage",
        "ec2messages:FailMessage",
        "ec2messages:GetEndpoint",
        "ec2messages:SendReply"
      ],
      "Resource": "*"
    }
  ]
}
```

Plus AWS-managed policies:

- `AmazonSSMManagedInstanceCore` — for SSM Session Manager browser-based SSH.

Plus, for the **Public-listing (Contract) fulfillment path only** — verifying the AWS Marketplace entitlement via AWS License Manager (no effect in the default BYOL mode):

- `license-manager:CheckoutLicense`, `GetLicense`, `CheckInLicense`, `ExtendLicenseConsumption`, `ListReceivedLicenses` (Resource: `*` — these don't support resource-level scoping; access is gated by the product + the buyer's contract).

That's it. The Refine container does not assume any IAM role to query customer cloud resources. Cloud-account onboarding is a separate, customer-initiated stack per account (Stack B below).

Non-IAM resources the stack creates (for completeness)

- 1× EC2 instance (default `t3.medium`, customer-sizable up to `m6i.4xlarge`).
- 1× EBS root volume (default 30 GB gp3).
- 1× S3 bucket for the delivery ZIP (customer-named or auto-named).

- 1× Security Group (inbound: SSH from `AllowedSshCidr`, app port 8000 from `AllowedAppCidr`).
- Optional: 1× Elastic IP.
- Optional: 1× Application Load Balancer + ACM listener + target group + ALB Security Group (when `EnableHttps=true`).

No Lambda, no IAM users, no IAM access keys, no Secrets Manager secrets.

Stack B — Per-Account Onboarding Stack (`customer-onboarding-v2.yaml`)

This stack is launched once **per AWS account the customer wants Refine to monitor.** It's separate from the Marketplace stack and gets launched from inside the running Refine UI (Cloud Accounts → Add Account → click "Launch CloudFormation"). Marketplace doesn't ingest this template directly, but reviewers commonly ask about it.

IAM resources created

Resource	Type	Purpose
`RefineCollectorRole`	`AWS::IAM::Role`	Assumed by the Refine host's instance role (cross-account, ExternalID-protected) to read inventory + cost data.
`RefineCollectorPolicy`	`AWS::IAM::ManagedPolicy`	Attached to the role. Read-only across the AWS services Refine analyzes.
`RefineCollectorLambdaRole`	`AWS::IAM::Role`	Execution role for the per-account collector Lambda (writes summaries to the customer's own S3 bucket).

Read-only permissions granted to `RefineCollectorRole`

The role grants `Read*`, `List*`, `Get*`, `Describe*` only — **never** `Create*`, `Update*`, `Modify*`, `Delete*`, `Put*` (except `s3:PutObject` on the customer-owned summary bucket, scoped to that one bucket).

Services covered (one read-only block per service):

Category	Service	Actions
Cost	Cost Explorer, Cost & Usage Reports, Budgets	`ce:Get*`, `ce:Describe*`, `cur:DescribeReportDefinitions`, `budgets:Describe*`, `budgets:View*`
Compute	EC2, EBS	`ec2:Describe*`, `ec2:GetEbsEncryptionByDefault`
Database	RDS	`rds:Describe*`, `rds:ListTagsForResource`
Storage	S3	`s3:ListAllMyBuckets`, `s3:GetBucketLocation`, `s3:GetBucketTagging`, `s3:GetBucketPolicyStatus`, `s3:GetBucketLifecycleConfiguration`, `s3:GetBucketIntelligentTieringConfiguration`
Network	ELB, NAT, VPC	`elasticloadbalancing:Describe*`, `ec2:Describe*` (NAT/EIP/VPC)
Serverless	Lambda	`lambda:List*`, `lambda:Get*`
Containers	ECS, Fargate	`ecs:Describe*`, `ecs:List*`
Cache	ElastiCache	`elasticache:Describe*`
Misc	Batch, Lightsail	`batch:Describe*`, `lightsail:Get*`
Monitoring	CloudWatch	`cloudwatch:GetMetricStatistics`, `cloudwatch:GetMetricData`, `cloudwatch:ListMetrics`, `logs:Describe*`
Identity (for boundary detection)	STS	`sts:GetCallerIdentity`
Savings Plans / RI	Savings Plans, EC2	`savingsplans:Describe*`, `savingsplans:List*`, `ec2:Describe*ReservedInstances*`
Write — single exception	S3 (own bucket only)	`s3:PutObject`, `s3:DeleteObject` scoped to `arn:aws:s3:::<summary-bucket>/*`

Trust policy

The role can only be assumed by:

- The Refine host's instance role ARN (provided as a CFN parameter at launch).
- With a matching `sts:ExternalId` value (random per customer; provided in the Refine UI).

This is the standard third-party cross-account access pattern documented at https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-user_externalid.html.

Per-account collector Lambda

The Lambda runs in the customer's AWS account on a nightly EventBridge schedule. Its execution role grants:

- The same read-only service permissions listed above.
- `s3:PutObject` to the customer-owned summary bucket.
- `logs:CreateLogGroup`, `logs:CreateLogStream`, `logs:PutLogEvents` (for its own CloudWatch logs).
- Nothing else.

No outbound network calls outside AWS. No third-party SDKs.

Stack C — GovCloud Onboarding Stack (`customer-onboarding-govcloud.yaml`)

Identical to Stack B with two differences:

- Partition switched to `aws-us-gov` throughout the IAM resource ARNs.
- The Cost Explorer / Cost & Usage Reports permissions are **dropped** (those services don't exist in AWS GovCloud).

Verification commands (for the customer's security team)

After Stack A is deployed, the customer can verify the host's permissions:

```
# What the EC2 instance role can do — should be only the S3+SSM list above.
aws iam list-attached-role-policies --role-name RefineInstanceRole
aws iam list-role-policies --role-name RefineInstanceRole
aws iam get-role-policy --role-name RefineInstanceRole --policy-name <inline-policy-name>
```

After Stack B is deployed in a monitored account:

```
# Confirm the role is read-only — should return zero write actions outside the summary bucket.
aws iam get-role --role-name RefineCollectorRole
aws iam list-attached-role-policies --role-name RefineCollectorRole
aws iam get-policy-version --policy-arn <RefineCollectorPolicy ARN> --version-id v1
```

For an automated audit, IAM Access Analyzer should report **zero external access risk** on either role beyond the explicitly intended cross-account trust to the Refine host.

Why this matters for AWS Marketplace review

- Refine ships **zero write permissions** on customer resources outside of writing summary files to a bucket the customer owns. This is the foundational claim of the product.
- The Marketplace host stack creates **no IAM permissions for the customer's monitored cloud accounts**. Customers explicitly opt-in by launching Stack B in each account.
- Cross-account access uses **the AWS-recommended ExternalID pattern** with rotating IDs per customer.
- All customer cost/inventory data lives **in the customer's own S3 / Blob / GCS bucket** — Refine reads from there. Blacktip Solutions has no network path to customer data.
