Hoogly AWS Access and Permission Documentation

Customer reference for the IAM role, trust relationship, and AWS API permissions used by Hoogly during onboarding and security scans.

Document generated	2026-04-12 22:32:00 UTC
Role template source	static/hoogly-auditrole.yaml
Support contact	support@hoogly.cloud
Managed policy reference	https://docs.aws.amazon.com/aws-managed-policy/latest/reference/SecurityAudit.html

1. Overview

Hoogly connects to a customer AWS account by assuming a customer-created IAM role. The role uses a trust policy with an ExternalId and attaches read-oriented permissions required to collect security configuration metadata, validate findings, and export audit-ready evidence. Hoogly does not require long-lived access keys for this flow.

2. Access model and trust configuration

Control	Configured value	Why it exists
Access mechanism	sts:AssumeRole into a customer-provided IAM role	Uses temporary credentials instead of static keys.
Trusted principal	arn:aws:iam::432925062020:root	Limits who can assume the audit role.
ExternalId condition	Required in the trust policy	Reduces confused deputy risk during cross-account role assumption.
Role name default	HooglyAuditRole2 in the template; app-generated deployments use a per-account role name.	Prevents role-name collisions and keeps onboarding consistent.
Credential lifetime	Temporary STS session, defaulting to 3600 seconds in the app	Restricts how long the assumed credentials remain valid.

3. Permissions attached to the role

The onboarding template attaches one AWS-managed policy and one inline read-only policy. The managed policy covers broad security metadata across AWS services, while the inline policy keeps a focused set of additional operations available for IAM, AWS Organizations, account context, and Trusted Advisor coverage.

Item	Value
Managed policy ARN	arn:aws:iam::aws:policy/SecurityAudit
AWS reference state	SecurityAudit v85, edited March 02, 2026, 17:12 UTC

Item	Value
Managed policy purpose	AWS describes this policy as granting access to read security configuration metadata for software that audits the configuration of an AWS account.
Full current JSON	Review the official AWS managed policy reference for the latest complete JSON and service coverage: https://docs.aws.amazon.com/aws-managed-policy/latest/reference/SecurityAudit.html

SecurityAudit policy service families commonly relevant to Hoogly scans

- IAM, Access Analyzer, and AWS account metadata
- S3, CloudFront, Route 53, WAF, and related edge/network controls
- EC2, VPC, load balancers, Auto Scaling, EKS, ECS, Lambda, and other compute surfaces
- CloudTrail, CloudWatch, AWS Config, GuardDuty, Detective, and security monitoring services
- RDS, DynamoDB, KMS, ACM, Backup, and other data protection or encryption surfaces
- CloudFormation, Organizations, Trusted Advisor, and governance-oriented inventory metadata

Inline action	Why Hoogly keeps it	Impact classification

4. Direct AWS API operations used in the app

In addition to the broader managed policy coverage used by the scan engine, the current Hoogly codebase explicitly makes the following AWS API calls for session establishment and live validation workflows.

API action	Current use in Hoogly
sts:AssumeRole	Establishes a temporary session into the customer-provided IAM role using an ExternalId.
sts:GetCallerIdentity	Confirms the assumed identity and account context after the role is assumed.
cloudtrail:DescribeTrails	Validates whether CloudTrail is enabled and whether multi-region trails are present.
guardduty:ListDetectors	Checks whether GuardDuty detectors exist in the selected region.
config:DescribeConfigurationRecorders	Checks whether AWS Config recorders are configured.
config:DescribeConfigurationRecorderStatus	Checks whether AWS Config recorders are actively recording.
iam:GetAccountSummary	Reviews account-level IAM posture such as root MFA coverage.
iam:ListMFADevices	Checks whether specific IAM users have MFA devices associated.
ec2:DescribeSecurityGroups	Validates whether security groups expose ingress rules to the public internet.
s3:GetPublicAccessBlock	Checks bucket-level S3 Block Public Access settings.
s3:GetBucketPolicyStatus	Checks whether an S3 bucket policy is effectively public.

API action	Current use in Hoogly
s3:GetBucketEncryption	Checks whether default S3 bucket encryption is configured.
rds:DescribeDBInstances	Checks whether RDS instances are publicly accessible.

5. Data handling and operational boundaries

- The onboarding role is intended for read-oriented inspection of security configuration metadata and evidence generation.

- Hoogly uses temporary STS credentials. It does not ask customers to create long-lived IAM users or embed static access keys.

- The inline IAM actions that begin with Generate* trigger report generation jobs, but they do not alter IAM policy, users, or trust relationships.

- The template uses Resource '*' because the scan must read posture across many services and regions. The access pattern is still designed around inspection rather than mutation.

- Customers can revoke access immediately by deleting the role, detaching the policies, or tightening the trust policy conditions.

6. Customer review checklist

Review item	What to confirm before production use
Trusted principal	Confirm the trusted AWS account ID in the role template matches the Hoogly deployment you are approving.
ExternalId	Confirm the ExternalId is unique to your onboarding flow and required by the role trust policy.
Policy review	Review both the attached SecurityAudit managed policy and the inline HooglyExtraReadOnly policy.
Revocation path	Document who can disable or delete the role if access must be removed quickly.
Change control	Track that AWS may update the SecurityAudit managed policy independently of Hoogly because it is AWS-managed.

7. Source references

Template source: static/hoogly-auditrole.yaml
AWS managed policy reference: https://docs.aws.amazon.com/aws-managed-policy/latest/reference/SecurityAudit.html
Application logic reviewed: scanner.py and security_report/validator_registry.py