How CIEM Works

Cloud Identity and Entitlement Management (CIEM), also known as Identity and Access Management (IAM), is used to control who can access which resources in your cloud environment and what actions they can perform on them. Wiz analyzes your IAM policies, both explicit and effective, to help you prevent over-privileging and lateral movement.

See the License Comparison page to understand which license is required for CIEM essential and CIEM advanced and learn what is included in CIEM advanced.

For step-by-step walkthroughs of key scenarios related to cloud entitlements, see the cloud entitlements scenarios. Learn also about the Explorer > Cloud Entitlementes page.

> You can watch our webinar on cloud entitlements.

How it works

Visualize your cloud identities

Wiz utilizes cloud provider APIs to provide you with full visibility into your cloud identities, their permissions, and effective access across your cloud providers. All the different identity components—users, service accounts, roles, groups, policies, etc.—are collected and standardized across the different cloud providers so that you can see everything related to IAM in Wiz using normalized terminology.

Supported Platforms

Wiz supports the major cloud providers:

- AWS
- Azure
- GCP

Wiz also supports the following SaaS platforms[1]:

- Snowflake
- GitHub

> 1️⃣ Coverage includes only Graph-based modeling, effective permission analysis, and MFA (Snowflake only).

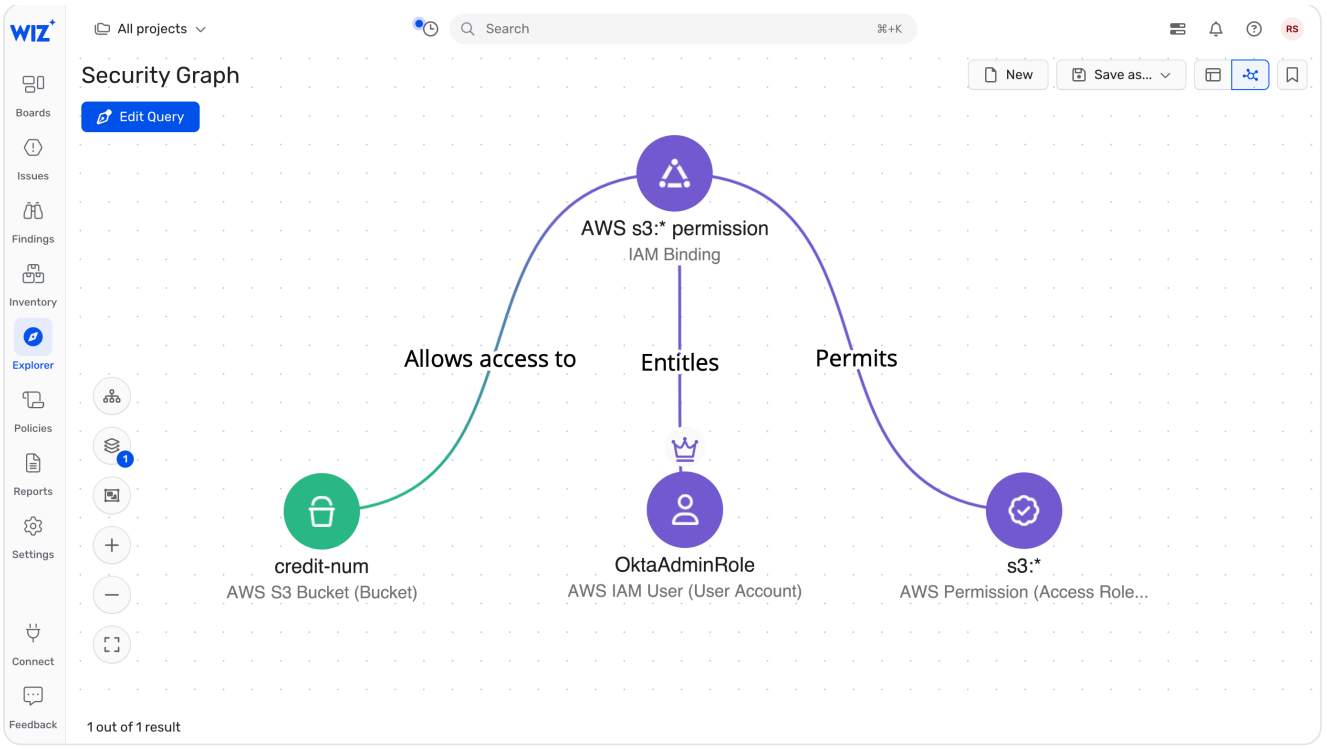
Additionally, Wiz supports all Kubernetes flavors, managed, self-hosted, or on-prem instances:

- EKS
- AKS
- GKE
- Self-hosted Kubernetes

Normalized data model

Each cloud provider has implemented its own IAM system using different terms, objects, and behaviors. Wiz converts these disparate systems into a single IAM model that standardizes the different cloud providers. The Wiz identity model has four different object types:

1. Principals—Users, service accounts, and groups, which are objects that have permissions to act in the environment.
2. Resources—Buckets, databases, keys, etc., which are objects that can be acted upon. Some of these objects have additional permissions restriction mechanisms, e.g. S3 buckets in AWS have resource policies.
3. Permissions—The actual API or actions available to the principal. They are mapped to one or more Access Types, depending on their functionalities.
4. IAM Bindings—A Wiz abstraction representing the effective permissions that connect a principal to a target resource. Represented on the Security Graph as a three-legged node connecting principals to resources with specific permissions (sample query).



Calculating cloud entitlements

Wiz has identity and entitlement algorithms that use a set of engines to calculate the following:

- IAM bindings
- Privileges
- Lateral movement

IAM bindings (aka effective permissions) calculation

All the identity components of your cloud environment, which the Cloud Scanner fetched, are processed and normalized. Then, your cloud environment is divided hierarchically into scopes in order to calculate the IAM bindings in two ways:

- Internally–Creating IAM bindings between principals and resources within the scope
- Externally–Creating IAM bindings between principals and the scope level itself

This calculation not only takes into account the granted permissions or principal-level restrictions such as AWS boundary policies, AWS service control policies, organizational restrictions, and guardrails, but also object-level policies (i.e. resource policies). This allows Wiz to show effective access to sensitive data, resources, or pipelines. For example, all principals with read access to a specific sensitive bucket.

For convenience, IAM bindings are only visible on the Security Graph; the Explorer > Cloud Entitlements page directly shows the link between principals, resources, and permissions.

Refer to the AWS condition keys breakdown in Wiz.

Privileges calculation

To calculate the privileges of principals in your environment, Wiz aggregates and accounts for all the IAM bindings previously calculated. Principals are flagged with admin or high-privilege permissions only if their IAM bindings refer to the scope level itself, not a specific resource within the scope.

- Admin permissions
- High-privilege permissions
- Data permissions
- Admin Kubernetes permissions
- High-Privilege Kubernetes permissions

Refer to the Impersonate and Admin and High-Privilege breakdowns in Wiz.

Admin permissions

Admin permissions are defined as permissions that can allow an attacker persistence in the environment. These can be IAM permissions to provision, create, delete, or update identities or wild card permissions on the Subscription (sample query).

Therefore, not every user or service account with admin permissions will be marked as one; for example, if a user is granted a set of admin permissions, but there is an organizational or account level block (such as an SCP in AWS), Wiz does not mark the user as an admin. Additionally, if a user is only granted permissions on a specific resource, Wiz does not indicate that user as an admin, since there is little to no risk of account takeover or wide disruption of workflows.

High-Privilege permissions

High-privilege permissions are defined as any permission or combination of permissions that allow(s) the creation, modification, deletion, or assignment of resources, identities, or other sensitive configurations that might disrupt the integrity of cloud workflows.

Sample query.

Data permissions

Data permissions are privileges granted to principals to read, write, or manipulate the data stored within a cloud service. They define what actions are permitted on the data and usually can be applied down to the "most granular level" of a resource. For example, in the case of AWS S3 service, data is stored within bucket objects, which are the smallest possible scope in that service. Therefore, permissions such as `s3:GetObject` and `s3:DeleteObject` would be classified as data permissions.

Sample query.

Admin Kubernetes permissions

Admin Kubernetes permissions are wildcard (*) permissions on the cluster and not only on specific namespaces.

Sample query.

High Privilege Kubernetes permissions

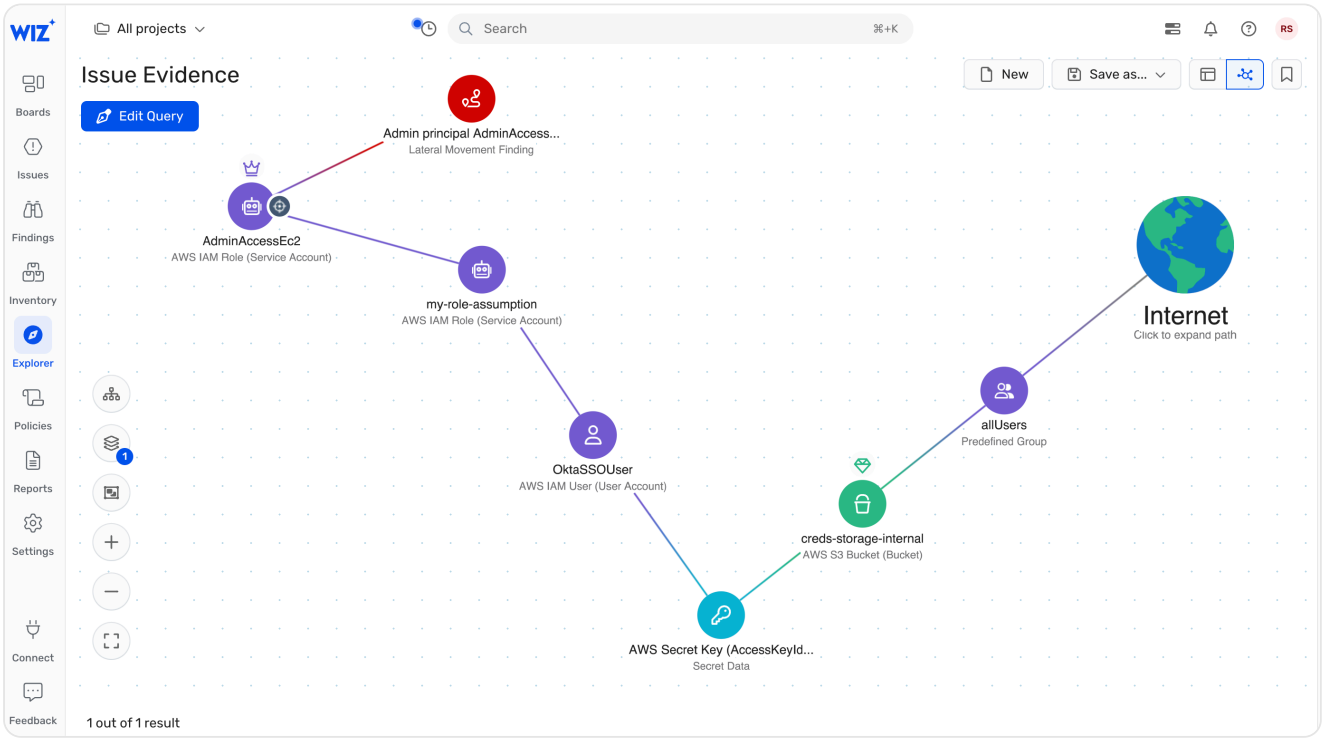
High-Privilege Kubernetes permissions are derived from three guidelines:

- Code execution—Any permission that allows the creation of a pod, job, or deployment can lead to code execution, whether the permission is on a cluster level or namespace level.
- Reading secrets—Any permission that allows reading secrets and thus might allow lateral movement or privilege escalation.
- Workflow disruption—Any permission that allows the deletion or update of critical resources such as IAM bindings, network components, etc. Such permission can be misused to break the cluster and the workflow.

Sample query.

Lateral movement calculation

Lateral movement is the ability to move from one resource to another. This can be done through roles or permissions that allow access to different areas of your cloud environment. Attackers often attempt to move laterally from one resource and/or environment to another to gain Admin permissions that allow them to take over your environment.



Based on the calculated IAM Bindings and privileges, Wiz calculates the potential paths from any resource in your environment to highly-privileged principals to determine if there is potential for account takeover within ten hops. The lateral movement hops are calculated by hopping from:

- Secret keys
- User accounts
- Service accounts
- Virtual machines
- Kubernetes cluster (long-term cloud keys, pod escape, etc.)

Try it in your environment by searching for VMs with Lateral Movement Findings.

Lateral movement between Kubernetes and cloud

Attackers leverage several techniques and functionalities to conduct lateral movement attacks from managed Kubernetes clusters to the cloud. These include the Instance Metadata Service, IAM/Microsoft Entra ID (AAD) identities, long-term cloud keys, and pod escape.

To reduce your clusters' attack surfaces, Wiz generates Issues to help you implement strict Kubernetes least-privilege approaches and curb network access. With regard to lateral movement from cloud to Kubernetes, Wiz maps IAM service accounts to Kubernetes service accounts, giving you deeper attack path analysis within your cloud environment.

Sample query.

Excessive access privileges

By incorporating Excessive Access Findings into its existing cloud entitlement capabilities, Wiz shows you inactive user accounts and service accounts, unused permissions, and publicly-exposed AWS resources. The excessive access analysis period can be configured on the Settings > Scanners > Cloud Entitlements page.

> ⚠ Excessive Access Findings...
>
> - ... for Azure require cloud events to be enabled in Wiz, with a learning period of 90 days.
> - ... for GCP require either access to GCP Policy Intelligence features or cloud events to be enabled in Wiz, with a learning period of 90 days.

Publicly-exposed AWS resources

AWS resources that have a resource-based policy that declares a wildcard for the principal that can access them are flagged with an Excessive Access Finding. See

sample query.

Inactive user accounts and service accounts

By default, user accounts and service accounts that were inactive for the last 90 days are flagged with an Excessive Access Finding. See sample query.

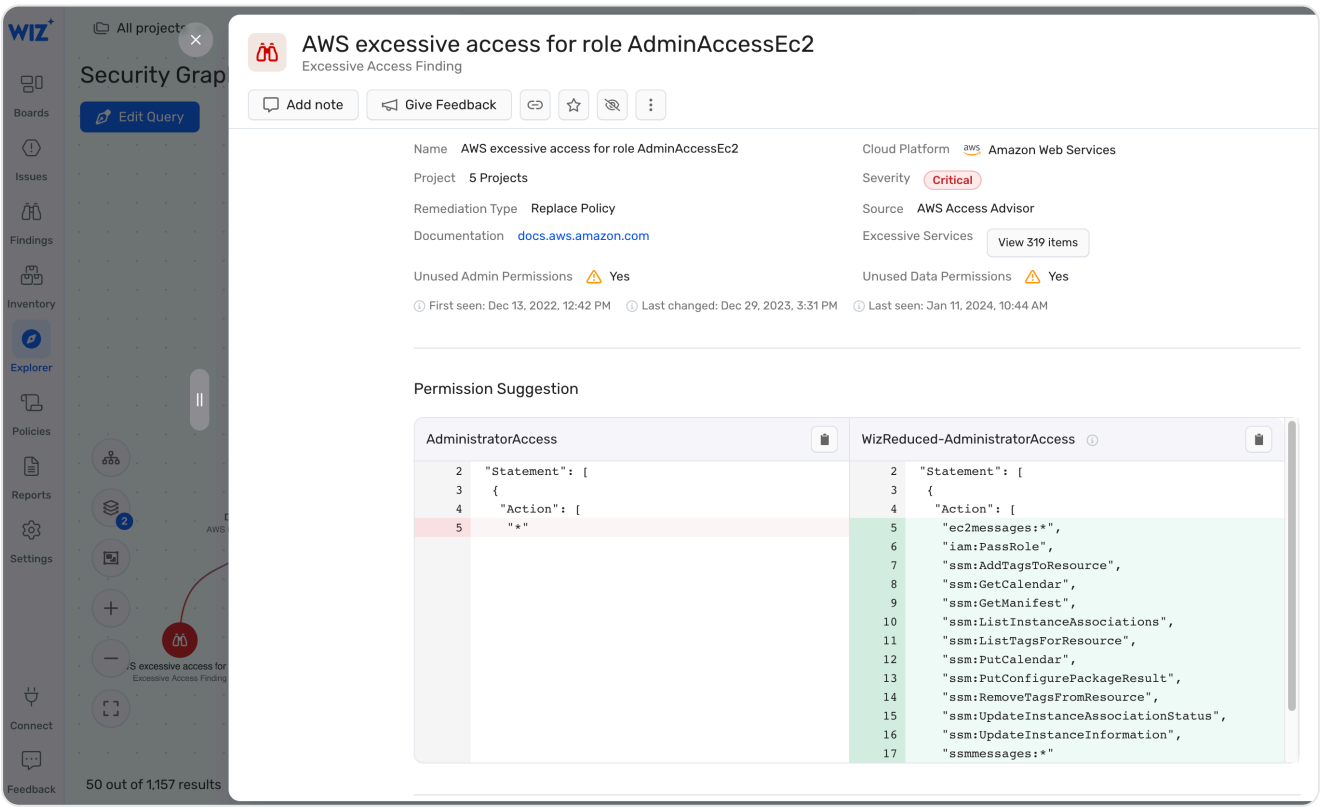
Unused permissions

By default, permissions that were not used for the last 90 days are flagged with an Excessive Access Finding. See sample query.

Excessive Access Findings on permissions are determined by comparing the permissions that each principal used against the total permissions the principal has. They are determined for each role assignment or attached policy and are derived as follows:

- AWS—Permissions that have not been used according to AWS Access Advisor.
- Azure—Permissions that have not been used according to Azure cloud events.
- GCP:
  - Permissions that have not been used according to GCP cloud events.
  - Permissions that have either not been used or determined as excessive access by GCP IAM Recommender.

To prevent unintended privilege escalation, Wiz's recommendations include only scoping down permissions, not increasing them.



The `Inactive For The last 90 days` property

The details drawer of principals includes the property `Inactive For The last 90 days`, that is based on metadata directly fetched from your cloud environment using your cloud provider's APIs. If your cloud provider does not support extracting this information (for example, in the case of GCP Service Agent or a user with Microsoft Entra ID (AAD) pricing plan less than P1), this property does not appear in the details drawer.

> AWS users configured with either active SSH public keys or a service-specific credentials (e.g. HTTPS Git credentials) are not marked as inactive by Wiz. This decision was made because API calls to AWS do not include "last usage" information, necessitating continuous vigilance to promptly address any potential threats to your environment.

Even if your cloud provider supports extracting this information, the following prerequisites must be completed:

- **Microsoft Entra ID (AAD)**

For user accounts—Your Azure Connector must have sufficient permissions to allow Wiz to extract principals' metadata. See the Microsoft Entra ID (AAD) required permissions.

> The `Inactive For The last 90 days` property is currently not supported for service accounts in Microsoft Entra ID (AAD). We recommend using Excessive Access Findings (sample query).

Google Workspace

- For user accounts—Your GCP Connector must have sufficient permissions to allow Wiz to extract principals' metadata. See the Google Workspace Connection guide and its required permissions.
- For service accounts—You must enable the Policy Analyzer API in the relevant GCP projects.

Predefined Groups

A Predefined Group is a Wiz abstraction representing a set of users who, according to our IAM-based effective permissions calculation, can access your cloud resources. The Predefined Group object on the Graph is created and associated with your cloud resources based on Wiz's identity analysis; it does not take network exposure and boundaries into account.

Predefined Groups are mapped into four types:

Type	CSPs	Description
All Users	AWS, Azure, GCP	Represents all users who have a Subscription in the relevant cloud service provider. For GCP-related binding, this type also represents anonymous users without a Google Account.
All Authenticated Users	GCP	Represents all users who have a Google Account (similar to what All Users shows for Azure and AWS).
All Subscription Users	AWS	Represents all AWS principals (users, groups, and IAM roles) that reside within a Subscription. Usually associated with the delegation format `"AWS": "arn:aws:iam::<account>:root"`
Domain Entity	GCP	Represents the domain associated with your organization resource and its connection to the GCP projects and billing accounts (via the `Billing Account Creator` and `Project Creator` roles).

Cloud-based IAM services

Wiz maps the roles and permissions of users managed in AWS Identity Center (sample query) or Microsoft Entra ID (AAD) (sample query) on the Security Graph in order to provide visibility into what resources they can access and what actions they can perform.

Public access

By analyzing the IAM Bindings, Wiz detects resources and principals upon which actions can be executed over the public Internet. This analysis finds IAM Bindings to `allUsers` or `allAuthenticatedUsers` or an AWS trust/resource policy that has wild card permission on the principal field, potentially allowing unapproved access to your resources or roles.

Wiz maps these public access permissions to 8 public Access Types:

1. Admin: Permissions that can allow an attacker persistence in the environment. These permissions can allow provisioning, creating, deleting, or updating resources.
2. High Privilege: A permission or combination of permissions that allow/s the creation, modification, deletion, or assignment of resources that might disrupt the workflow.
3. Data: Any permission (Read, Write, List, etc.) that allows exposure or modification of data hosted on a cloud resource.

4. Impersonate: Permission or combination of permissions that allow/s an IAM identity (user/service account) to impersonate another IAM identity, either by a direct role assumption (AWS) or by having the ability to change and create authentication credentials.
5. Manage: Permissions that enable principals to configure, control, and monitor various aspects of cloud resources.
6. Write: Permissions that allow the creation, deletion, or modification of content in cloud resources.
7. Read: Permissions that allow reading access to the content hosted on cloud resources.
8. List: Permissions that allow listing the content in cloud resources without necessarily interacting with it or being able to read it.

Full Admin and High privilege permission breakdown

Sample query

External access

Wiz can also detect resources that are accessible from specific external subscriptions or third-party organizations. This analysis flags assumed roles or resources with `IAM access from external subscription`, providing security teams visibility into which principals have approved access, whether approved third-party vendors or unapproved access that was unintentionally created (sample query).

Sample queries

Use the following queries to gain visibility into the identity plane in your environment:

- Security Graph–Visibility into your cloud identities
  - View all cloud users
  - View all groups
  - View the service accounts of VMs
- Cloud Entitlements page–Visibility into effective access between principals or resources that assume roles and a resource, Subscription, Organization, or principal
  - View the permissions of a specific user
  - View the buckets accessible to a specific service account
  - View all the buckets a VM can write to

FAQ

Questions? Take a look at the FAQ.

Updated 9 days ago

← CIEM Tutorials

CIEM FAQ →

Did this page help you? Yes No