Yes, access-review platforms can use cloud APIs to collect access data—but whether **Access Auditor** can pull data directly from a particular cloud service depends on the product's available connectors, supported API endpoints, and configuration. "Cloud API support" is not a single feature that works identically across every application. Before choosing a tool or planning a rollout, confirm exactly which systems it can connect to, what data it retrieves, and how that data is refreshed and retained.

That distinction matters for organizations moving from spreadsheet-based reviews to a repeatable identity governance process. A platform may support direct connections for some systems, scheduled file imports for others, and manual uploads for applications that lack a practical integration path. The goal is not to check an "API" box. It is to collect reliable, reviewable access data and route it to the right people with evidence that can be produced later.

What "pulling access data from a cloud API" means

A cloud API is a defined interface an application exposes so another system can request information or perform actions. For access reviews, an integration might use an API to retrieve some combination of:

- Accounts, including account identifiers and status
- Users or other identities associated with those accounts
- Groups, roles, permissions, or entitlements
- Application ownership or organizational attributes
- Changes made since a prior collection, if the source supports them

Those categories are not interchangeable. A directory API may provide group membership but not the business meaning of each permission. A software-as-a-service application may expose user accounts but omit a complete entitlement catalog. Some systems distinguish human accounts from service accounts, integrations, and other non-human identities. A useful integration has to preserve these distinctions rather than flattening everything into a generic "user" record.

There is also a difference between collecting access data and managing access. An API connection might be read-only and used to prepare review campaigns. It might also support remediation actions, such as disabling an account, but that is a separate capability with separate risk and approval considerations. Ask what the integration actually does—not just whether it is called an API connector.

Can Access Auditor connect directly to cloud applications?

The answer is specific to the Access Auditor configuration and the applications in scope. A buyer should request a current connector list and validate each needed integration against its actual requirements. Confirm whether the connection is direct, uses a third-party integration layer, or depends on an export that is imported later. Also ask which API endpoints and data fields are supported, how authentication works, and how often the data can be refreshed.

If a direct connector is available, it may help reduce manual file preparation and make recurring collection more consistent. But "connected" does not automatically mean "complete." The connector may retrieve only certain account types or permissions, rely on attributes that are not populated in the source, or need additional mapping to produce review-ready records. A proof of concept should test real data, including edge cases such as disabled accounts, duplicate names, nested groups, and service accounts.

For organizations with a mix of cloud and on-premises systems, the practical question is broader: Can the platform support a reliable collection method for every application that belongs in the review? A hybrid environment may include cloud directories, core banking platforms, file shares, databases, and locally managed applications. Direct APIs can be valuable, but they are only one part of the integration strategy.

Flexible data imports matter as much as APIs

Not every system offers a suitable API, and not every API is accessible to an organization's review platform. Some applications expose limited endpoints, require custom development, or restrict access for security or licensing reasons. Others may produce the necessary information through scheduled reports or structured exports. That is why **flexible data imports** are an important selection criterion.

A capable import approach should let the organization bring in data from multiple sources while preserving its meaning. Depending on the application and platform, that could involve a standard connector, a structured CSV file, an automated file transfer, or a more tailored integration. The important test is whether the import can be repeated consistently and whether the resulting records can be reconciled to the source.



Ask how the platform handles:

- Required and optional fields, including identifiers used to match records
- Data transformation, such as mapping source roles to review categories
- New, changed, and removed records between review cycles
- Duplicate accounts and identities with similar names
- Accounts without a clear owner or manager
- Errors, rejected rows, and incomplete imports
- Retention of the source file, import results, and collection timestamp

Flexible imports should not mean accepting opaque spreadsheets without control. Define who can submit files, where they are stored, how they are validated, and what happens when a file fails. Where possible, automate delivery from the source system and retain the original export alongside processing results. That gives reviewers and auditors a path from the campaign record back to the collected evidence.

Moving from spreadsheets to platform-based reviews

Spreadsheet-based access reviews often begin as a practical solution: an administrator exports a list, distributes it, and collects responses. Over time, the process can become difficult to control. Files are copied, columns are changed, reminders are sent manually, and approval decisions may be scattered across email threads. The result can be hard to reproduce, particularly when someone asks who reviewed a specific account and what information was available at the time.

Automating the process with an access-review platform can centralize the campaign, assignments, decisions, and follow-up. It does not eliminate the need to validate source data or design a sound review. It can, however, make recurring tasks more consistent when the workflow is configured well.

A typical process may look like this:

1. **Collect:** Retrieve access data through a supported API or prepare a controlled import from the source system.
2. **Validate:** Check record counts, required fields, duplicates, and exceptions before launching a campaign.
3. **Map:** Associate records with the correct reviewers using reliable ownership, manager, or application data.
4. **Assign:** Route decisions to designated reviewers, with escalation rules for missing or overdue responses.
5. **Review:** Present enough context for the reviewer to make a decision without obscuring the underlying account or entitlement.
6. **Remediate:** Track approved changes to completion, whether performed through an integration or in the source system.
7. **Retain evidence:** Preserve campaign details, decisions, timestamps, exceptions, and remediation status.

This workflow is more useful than a feature list because it exposes where the process can break. A technically successful import can still produce a poor review if accounts are assigned to the wrong owner, reviewers cannot distinguish similar entitlements, or decisions are captured outside the system of record.

Workflow assignment and reviewer experience

Reviewer experience is not a cosmetic concern. If reviewers receive large, unclear assignments, they may rush through decisions or ask administrators to explain the data outside the platform. Effective workflow design presents the right records to the right people, provides enough context to make a decision, and records that decision in a consistent place.

During a product evaluation, test how Access Auditor—or any alternative—supports assignment rules. Can campaigns be divided by application, department, manager, or account owner? What happens when an assigned reviewer leaves the organization, has no available manager, or is responsible for an unusually large number of records? Can a delegate be assigned with appropriate controls? Are overdue items visible, and can reminders or escalations be managed within the workflow rather than through informal email replies?

Reviewers should also be able to Symitar access review automation understand what they are certifying. For example, an account identifier alone may not explain whether an account is active, what permissions it holds, or why it exists. Include context that supports a decision, but avoid overwhelming reviewers with irrelevant attributes. Make sure the interface distinguishes direct and inherited access where that distinction is available in the source data.

Finally, test exceptions. A process that works for ordinary employee accounts may fail for shared mailboxes, administrative clones, service accounts, emergency accounts, and vendor identities. These are common sources of review confusion. Define who owns each type, what decision is expected, and whether it should follow a different review path.

Identity governance selection criteria

When assessing an identity governance tool, start with the organization's systems and control objectives—not with a claim that a vendor "supports cloud." Build an inventory of applications, identity stores, data owners, review populations, and collection methods. Then compare platforms against that inventory.

Selection area Questions to ask Evidence to request Integration coverage Which systems connect directly, and which require imports or custom work? Connector documentation and a demonstration using in-scope data Data completeness Which account types, roles, groups, and permissions are collected? Field mapping, sample output, and documented exclusions Collection operations How are credentials, refresh schedules, failures, and changes handled? Run history, error handling, and access-control details Review workflow How are reviewers assigned, reminded, delegated, and escalated? A tested campaign with realistic exceptions Audit evidence Where does the evidence live, and can decisions be tied to source data? Exportable logs, timestamps, campaign history, and retention settings Remediation Can actions be tracked or executed, and who approves them? A documented workflow showing authorization and completion status

For financial institutions and credit unions, implementation context matters. An organization working with a core-processing provider such as Member Driven Technologies (MDT) may need to coordinate application access data across its own environment and provider-supported systems. A security or compliance partner such as Security Compliance Corp (SCC) may also be involved in assessment or control work. In either case, clarify who owns the source data, who can authorize access, and which party maintains evidence. A vendor relationship does not remove the need to define those responsibilities.

Security, evidence, and operational details

Before enabling an API connection, review the security model. Ask whether credentials are stored securely, whether the connection can use a least-privilege service account, and how secrets are rotated. Confirm whether the integration is read-only or can modify accounts and permissions. If the platform supports remediation actions, determine whether those actions require approval and whether they are logged with the initiating identity and outcome.

Then ask the question that often gets skipped: **Where does the evidence live?** A useful record should show what was collected, when it was collected, how the campaign was assigned, who made each decision, and what happened afterward. Logs and retained system records are generally more dependable than screenshots, which can omit context and are difficult to reconcile with later changes. Screenshots may supplement evidence, but they should not replace available logs and structured records.

Also establish how long source exports, campaign records, and remediation evidence are retained. Make sure the organization can retrieve them for an audit or investigation without relying on one administrator's mailbox or local drive.

How to validate Access Auditor's cloud API capabilities

A focused proof of concept is usually more informative than a general product demonstration. Select one cloud application with a supported API, one system that will need flexible data imports, and a review population that includes ordinary and exception accounts. Then test the whole process—not just whether records appear on a screen.



- Compare collected records with a trusted source export or administrative report.
- Confirm whether the integration captures the required users, accounts, groups, and entitlements.
- Check how disabled, orphaned, duplicate, service, and privileged accounts are represented.
- Run a sample import and inspect validation messages and rejected records.
- Assign reviewers and test delegation, reminders, escalations, and reassignment.
- Record a decision, track remediation, and retrieve the evidence afterward.

Document limitations as well as successes. If a field is unavailable from the API, decide whether another source can supply it or whether the review should be redesigned. If an import requires manual cleanup, estimate the effort and identify who owns that work. This makes the rollout plan realistic and helps prevent a polished demonstration from being mistaken for a complete operating process.

The bottom line

Access Auditor may be able to pull cloud API access data for supported applications, but the answer depends on the specific connector, data scope, and configuration. For systems without a suitable direct integration, flexible data imports can help bring them into the same review program—provided those imports are controlled, repeatable, and traceable.

Choose an identity governance platform based on the end-to-end workflow: reliable collection, clear reviewer assignment, understandable decisions, tracked remediation, and evidence that can be retrieved later. The strongest solution is not necessarily the one with the longest connector list. It is the one that can represent your real accounts and entitlements accurately, handle exceptions without relying on email as the system of record, and leave a defensible trail from source data to final decision.