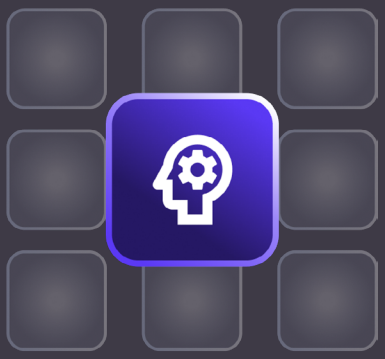
ConductorOne

13 Identity and Access Management (IAM) Best Practices



Identity and Access Management (IAM) has become a critical, yet increasingly complex, pillar of modern cybersecurity. In the past, managing user access was a relatively straightforward process confined within the four walls of the organization. Today, that perimeter has dissolved. Your team is now responsible for securing a sprawling ecosystem of cloud applications, on-prem systems, infrastructure, and a rapidly growing workforce of non-human identities.

This explosion in identity count is staggering. According to a 2024 report from ESG, organizations now have, on average, 20 times more non-human identities than human ones, and over half expect that number to grow by another 20% in the next year alone. [*] This sprawl, coupled with the rise of hybrid work, creates significant security and operational challenges. Manually provisioning access, conducting user access reviews, and ensuring compliance across dozens of disconnected systems is no longer sustainable. It's a constant drain on IT and security resources, and the risk of error—ike orphaned accounts for unauthorized users or excessive permissions—grows with each new user, service account, and application.

To effectively manage this new reality, organizations must move beyond legacy approaches and adopt modern IAM best practices. Doing so is not just about strengthening security; it's about enabling the business to operate with speed and confidence.

This guide provides a clear, actionable guide to the essential IAM best practices that will help you secure your environment, streamline operations, and ensure you are prepared for the future of identity.

01 Adopt a Zero Trust architecture

Adopting a **Zero Trust architecture** is critical because the traditional network perimeter has dissolved. With resources in the cloud, a remote workforce, and a growing number of non-human identities, you can no longer assume that requests originating from “inside” the network are safe.

This modern approach directly confronts this reality by operating on a simple but powerful principle: never trust, always verify. It treats identity as the new security perimeter, requiring strict validation for every access request, regardless of its origin.. The impact is significant; according to IBM, organizations that have fully deployed a Zero Trust model save an average of $1 million in data breach costs, making it a cornerstone of both a secure and financially sound security strategy. [*]

How to implement it:

- → **Identify your attack surface**: Begin by discovering and classifying your most critical data, applications, and assets. You cannot protect what you do not know you have.
- → **Map the transaction flows**: Analyze how users and systems interact with the critical assets you’ve identified. Understanding these flows is essential for designing effective security policies without disrupting business operations.
- → **Architect your Zero Trust network**: Instead of one large perimeter, design micro-perimeters around your critical assets. A software-defined perimeter can create these individualized and isolated network segments.
- → **Create and enforce Zero Trust policies**: Apply the principle of least privilege to create granular access policies that protect sensitive information. Enforce these policies with strong identity controls like multi-factor authentication (MFA), just-in-time (**JIT**) access, and continuous authorization.
- → **Monitor and maintain**: Continuously monitor all network traffic and access requests for suspicious activity. Use this data to adapt and refine your policies.

02 Establish an IAM governance committee

An effective Identity and Access Management program is a core business function that requires strategic oversight. Without a formal governance structure, IAM initiatives often become siloed, leading to inconsistent policies, conflicting priorities between departments, and a failure to deliver their full value.

Establishing an IAM governance committee elevates identity management from a purely operational task to a strategic enterprise program. This central body ensures that IAM policies align with business objectives, regulatory requirements like GDPR and HIPAA, and security goals.

According to Gartner, IAM projects frequently suffer from excessive time and cost unless organizations adopt a program-centric approach. [*] A governance committee is the engine that drives that approach, providing the authority and cross-functional collaboration needed for success.

How to implement it:

- → **Define the charter and mission**: Clearly document the committee's purpose, scope, and authority. This charter should outline its responsibility for setting IAM policy, resolving disputes, and measuring the program's success.
- → **Assemble a cross-functional team**: The committee should not be limited to IT and Security. Include leaders and stakeholders from HR, Legal, Compliance, and key business units to ensure policies are practical and supported across the organization.
- → **Establish user roles and responsibilities**: Designate a committee chair and define the roles of each member. The committee's focus should be on strategic direction and policy, not on the day-to-day technical management of IAM systems.
- → **Set a regular meeting cadence**: Schedule recurring meetings to review risks, approve new policies, assess the program's performance against key metrics, and adapt to new business needs or threats.

03 Establish a single source of truth for all identities

In most organizations, identity data is scattered across dozens of disconnected systems—from HR platforms and on-prem Active Directory to cloud directories and individual SaaS applications. This identity sprawl creates massive security and operational challenges. It leads to inconsistent data, duplicate accounts, and dangerous security gaps, like orphaned accounts that retain access after an employee has left.

Establishing a single source of truth (SSoT) for identity data is the solution to this chaos. By centralizing and consolidating digital identities into one authoritative repository, you create a complete and reliable view of every user and system in your environment. This is no longer optional; research from the Identity Defined Security Alliance (IDSA) shows that 71% of organizations believe their number of identities is growing to a problematic level. [*] An SSoT provides the foundation needed to manage this scale, enabling consistent policy enforcement and effective automation.

How to implement it:

- → **Discover and inventory all identity silos**: Conduct a comprehensive discovery process to locate every place identity data is stored, including cloud and on-prem directories, applications, and databases.
- → **Select your authoritative source(s)**: Choose a primary identity store to act as the central repository. You may have different authoritative sources for different attributes (e.g., HR for employment status, IT for group memberships).
- → **Integrate and synchronize data**: Use connectors and APIs to integrate your various systems with the central store. Implement automated synchronization processes to ensure that when identity data is updated in the source, the changes are propagated everywhere.
- → **Incorporate all identity types**: Ensure your SSoT includes not just human employees and contractors, but also the rapidly growing number of non-human identities, such as service accounts, API keys, and AI agents.
- → **Define data governance rules**: Establish clear processes for who can create, modify, and approve changes to identity data to maintain its accuracy and integrity over time.

04 Automate the full identity lifecycle

Manually managing user access at every stage—from onboarding to role changes and eventual offboarding—is inefficient and a significant source of security risk. IT teams get bogged down in repetitive ticketing work, and delays are common. More importantly, manual processes inevitably lead to errors, such as privilege creep, where users accumulate unnecessary access over time. This leads to risky standing privileges, which will be explained in detail later.

Automating the full identity lifecycle addresses these challenges directly by turning manual, error-prone tasks into a streamlined, policy-driven workflow. By integrating your authoritative identity source (like an HR system) with your IT applications, you ensure access is granted, modified, and revoked in real-time based on a user's status. This not only frees up your technical teams for more strategic work but also dramatically strengthens your security posture by closing access gaps the moment they appear.

How to implement it:

- → **Map your "Joiner, Mover, Leaver" (JML) processes**: Document the exact access an individual needs on day one (Joiner), how that access should change with a promotion or transfer (Mover), and what must be done immediately upon their departure (Leaver).
- → **Implement birthright provisioning: Implement birthright provisioning**: For new employees, create automated workflows for granting access based on their role, department, or other attributes defined in your HR system. This ensures new hires are productive from day one.
- → **Automate access modifications**: When an employee moves to a new role, trigger automated workflows that revoke unnecessary permissions from their old role and grant the new ones. This systematically prevents privilege creep, where users accumulate access rights over time.
- → **Ensure immediate de-provisioning**: This is the most critical security step. Configure your system to automatically disable all accounts and revoke access across all connected applications the moment an employee's status changes to "terminated" in the authoritative source.

05 Enforce the principle of least privilege (PoLP)

Over-privileged accounts are one of the most significant (and common) security risks. When a user or service account has more access than it needs, it creates an unnecessarily large attack surface. If that account is compromised, an attacker instantly gains all of its excessive permissions, allowing them to move laterally, escalate privileges, and access sensitive data.

Enforcing the **principle of least privilege** (PoLP) means granting only the bare minimum level of access required for a user or system to perform its specific function. This is a core tenet of a Zero Trust strategy. According to the most recent Verizon Data Breach Investigations Report, the use of stolen credentials remains a top cause of data breaches. [*] By limiting access permissions, PoLP dramatically reduces the potential damage (the blast radius) of a compromised credential.

How to implement it:

- → **Audit all existing permissions**: You cannot enforce what you cannot see. Start by conducting a comprehensive audit of all current access rights across your applications and infrastructure to establish a baseline.
- → **Implement a default deny policy**: Shift your security posture so that users start with zero access by default. Permissions should then be explicitly and individually granted based on a legitimate business need.
- → **Define granular entitlements**: Move away from broad, generic access levels like "user" or "admin." Break down application permissions into granular entitlements based on specific job functions and tasks.
- → **Conduct regular access reviews**: PoLP is not a "set it and forget it" policy. Use automated access reviews and certifications to regularly verify that permissions are still required and remove any that are no longer necessary.

06 Leverage role-based and attribute-based access control (RBAC & ABAC)

As an organization grows, managing permissions on a purely individual, user-by-user basis becomes unscalable and chaotic. This approach makes it nearly impossible to enforce policies consistently, creating security gaps and a massive administrative burden for IT teams.

Role-based access control (RBAC) and attribute-based access control (ABAC) provide scalable, policy-driven frameworks to solve this problem. RBAC simplifies administration by grouping users with similar job functions into roles and assigning permissions to those roles. ABAC enhances this model by adding real-time, contextual data, such as user location, device security posture, or resource sensitivity, to make more dynamic and granular access decisions. Using them together gives you both a stable structure and the flexibility required for a modern, secure environment.

How to implement it:

- → **Start with an RBAC foundation**: Begin by analyzing your organization and defining a clear set of roles based on job titles and responsibilities (e.g., "Accounts Payable Clerk," "Network Engineer").
- → **Map entitlements to roles**: Following the principle of least privilege, assign the specific permissions required for each role to perform its duties.
- → **Layer on attributes for dynamic control**: Enhance your RBAC model by incorporating ABAC. Create context-aware policies, such as allowing access to a financial application only during business hours and from a corporate-managed device.
- → **Automate policy enforcement**: Use an identity platform that can ingest roles and attributes from your various systems to automatically evaluate and enforce access decisions in real-time.
- → **Review and refine roles continuously**: Roles and access needs are not static. Regularly review your RBAC and ABAC models to ensure they align with organizational changes and evolving security requirements.

07 Implement just-in-time (JIT) access for privileged resources

Standing privilege, which grants administrators, developers, and other technical users 24/7 access to sensitive systems, poses a significant security risk. These “always-on” permissions are a prime target for attackers. If a privileged account is compromised, the attacker gains immediate, unfettered access to your most critical infrastructure, such as cloud consoles, databases, and production servers.

Just-in-time (JIT) access directly solves this problem by eliminating standing privileges. Instead of having access all the time, users are granted temporary, elevated permissions on-demand and only for as long as needed to complete a specific task. This approach, a core pillar of modern Privileged Access Management (PAM) and Zero Trust, dramatically shrinks the attack surface. By ensuring credentials are only valid for a short time, you neutralize the threat of a compromised privileged account being used against you later.

How to implement it:

- → **Identify and inventory privileged accounts and assets**: Begin by discovering all your privileged resources (e.g., servers, databases, SaaS admin consoles) and mapping the human and machine accounts that can access them.
- → **Define JIT access policies**: For each resource, establish rules for who can request access, the level of permission they can receive, and the maximum duration for which access can be granted.
- → **Establish a self-service request workflow**: Make it easy for users to request temporary access through a streamlined portal or a chat tool like Slack or Microsoft Teams.
- → **Automate approvals and provisioning**: Configure the system to automatically approve requests that fit pre-defined policies or route exceptions to a manager for approval. Upon approval, access should be provisioned automatically.
- → **Ensure automatic revocation and auditing**: Once the approved time expires, access must be revoked automatically without any manual intervention. All JIT sessions—including the request, approval, and user activity—should be fully logged for security reviews and compliance audits.

08 Deploy single sign-on (SSO) to unify authentication

The explosion of cloud and SaaS applications has led to severe password fatigue. A recent NordPass survey revealed that the average employee now manages nearly 90 different work-related accounts. [*] This forces users into insecure habits, like reusing the same password across multiple services. For IT teams, this sprawl means there is no central point of control, making it nearly impossible to enforce security policies or quickly deprovision a user’s access to everything.

Single sign-on (SSO) solves both the user experience and security problem. It allows users to authenticate once with a primary identity provider and gain secure access to all their approved applications without needing to enter different credentials for each one. This centralization gives security teams a single place to enforce authentication policies, like MFA, and provides complete visibility over who is accessing what. It also allows IT to grant or revoke access to all connected applications with a single click.

How to implement it:

- → **Select a primary identity provider (IdP)**: Choose a modern IdP to serve as the central authentication authority for your organization.
- → **Integrate your most critical applications first**: Start by connecting your most widely used or highest-risk applications (e.g., your cloud provider, CRM, and primary communication tools) to the SSO solution to demonstrate immediate value.
- → **Use modern and secure federation standards**: Prioritize using open standards like SAML 2.0 and OpenID Connect (OIDC) to ensure secure and interoperable integrations with your applications.
- → **Combine SSO with multi-factor authentication methods (MFA)**: Because SSO centralizes authentication, protecting the primary login is critical. Always enforce MFA on the SSO login itself to add a crucial layer of security.
- → **Communicate and drive user adoption**: Clearly communicate the benefits of SSO to your employees and provide straightforward instructions to ensure a smooth transition away from legacy, direct-to-app logins.

09 Enforce strong password policies

Even with modern security controls, passwords remain a fundamental part of authentication and are often the weakest link. The latest Verizon Data Breach Investigations Report shows that the use of stolen credentials continues to be a top vector in successful breaches. [*] Attackers relentlessly use brute-force and credential-stuffing attacks, which are highly effective when users choose simple, guessable passwords or reuse them across multiple services.

Enforcing a strong password policy is a foundational security control that establishes a critical baseline of defense. It makes it significantly harder for attackers to guess or crack user credentials, thereby reducing the risk of an initial compromise. While not a silver bullet, a robust policy is an essential layer in a defense-in-depth strategy, forcing attackers to work much harder to gain a foothold in your environment.

How to implement it:

- → **Go beyond basic complexity**: In addition to mandating length and character types (uppercase, lowercase, numbers, symbols), use a service to proactively block users from choosing common and previously breached passwords like “Password123” or “Summer2025.”
- → **Establish a reasonable rotation schedule**: Require users to change passwords periodically (e.g., every 90 days) to limit the window of opportunity for an attacker if a credential is stolen.
- → **Implement account lockout policies**: Automatically lock a user account after a set number of consecutive failed login attempts to thwart automated brute-force attacks.
- → **Educate users on password hygiene**: Teach employees why these policies exist. Encourage the use of reputable password managers, which allow them to create and store unique, complex passwords for every service without needing to memorize them.

10 Require multi-factor authentication (MFA) universally

Multi-factor authentication (MFA) is arguably the single most effective security control you can deploy to prevent unauthorized access. By requiring an extra layer of validation beyond the password, such as a code from an authenticator app (something you have) or biometric authentication (something you are), you neutralize the threat of stolen credentials. According to Microsoft's research, implementing MFA blocks over 99.9% of account compromise attacks. [*] It is an essential, non-negotiable layer of security for every modern organization.

How to implement it:

- → **Apply MFA to all users without exception**: Enforce MFA for every single user, including all employees, contractors, partners, and especially privileged administrators and executive accounts.
- → **Secure every point of access**: Require MFA not just for application logins via SSO, but for all access pathways, including VPN connections, direct cloud console logins, and remote access to servers (SSH/RDP).
- → **Prioritize phishing-resistant factors**: While SMS codes are better than nothing, encourage the adoption of more secure methods like authenticator apps (TOTP), push notifications with number matching, or the gold standard: hardware security keys (FIDO2/WebAuthn).
- → **Leverage adaptive MFA for enhanced security**: Use your identity platform to create context-aware policies. This allows you to step up authentication challenges based on risk signals, such as a login from a new device, an unusual location, or an impossible travel scenario.

11 Automate regular access reviews and audits

Manual access reviews are notoriously painful. They require IT teams to spend dozens of hours pulling data from spreadsheets and chasing down business managers. This leads to review fatigue, where overwhelmed managers simply rubber-stamp access without proper scrutiny, defeating the entire purpose of the review. This practice allows privilege creep and standing access to go unchecked, leaving dangerous security gaps open for auditors (or attackers) to find.

Automating the user access review (UAR) process transforms it from a dreaded annual event into a streamlined and continuous security function. Automation can provide reviewers with the critical context they need to make intelligent decisions quickly. According to various industry reports, companies that automate their UARs can reduce the time spent on reviews by as much as 70% compared to manual methods, while simultaneously improving accuracy and providing a frictionless experience for everyone involved.

How to implement it:

- → **Trigger reviews based on risk and events**: Instead of a single, massive annual review, create smaller, more frequent campaigns. Trigger reviews automatically based on events like a user changing roles, or conduct them on a regular cadence for high-risk applications and privileged access.
- → **Provide reviewers with intelligent context**: Present access information in a simple, easy-to-understand format. Enrich the data with insights like how a user's access compares to their peers, when they last used a permission, and an AI-driven recommendation to approve or revoke.
- → **Automate the entire workflow**: Implement a system that automates notifications, sends reminders to reviewers for incomplete tasks, and escalates overdue reviews to a manager to ensure campaigns are completed on time.
- → **Integrate revocation into the process**: Once a reviewer decides to revoke access, the system should automatically trigger a workflow to deprovision the entitlement immediately, closing the security loop without manual intervention.
- → **Generate audit-ready reports instantly**: Use a platform that can generate a complete report of any review campaign with a single click, showing who reviewed what, when, and the final outcome, to easily satisfy auditor requests.

12 Centralize and monitor activity logs

In a typical organization, activity logs are scattered across hundreds of different applications, servers, and identity providers. Without a centralized view, it's nearly impossible to correlate events, detect a sophisticated attack pattern, or conduct an effective forensic investigation after an incident occurs.

Centralizing your identity and access logs into a SIEM (Security Information and Event Management) system provides a unified view of all activity, enabling you to spot threats in near real-time. This visibility is critical for reducing attacker dwell time —the period between initial compromise and detection. According to IBM, the average data breach takes 204 days to even identify [*]. Effective monitoring of centralized logs is one of the most powerful tools for reducing the window of vulnerability and minimizing the potential damage of a cyberattack.

How to implement it:

- → **Collect logs from all critical identity sources**: Ingest logs from all relevant systems, including your identity provider (IdP), cloud platforms (AWS, Azure, GCP), critical SaaS apps, servers, and network devices.
- → **Normalize log data into a consistent format**: Ensure that logs from different vendors and systems are parsed and converted into a standardized format so they can be correlated and analyzed together.
- → **Establish a baseline of normal activity**: Before you can spot anomalies, you must understand what is normal. Use analytics to profile typical user and system behavior.
- → **Create real-time alerts for suspicious events**: Configure automated alerts for high-risk activities, such as impossible travel (e.g., a login from New York followed by one from Tokyo five minutes later), multiple failed login attempts followed by a success, or privilege escalation outside of an approved JIT workflow.
- → **Regularly review and test your detection rules**: The threat landscape is constantly changing. Periodically test and refine your monitoring rules to ensure you can detect the latest attacker techniques.

13 Continuously improve and adapt your IAM program

Treating IAM as a continuous, iterative program is the only way to ensure it remains effective over the long term. This approach focuses on moving up the IAM maturity model—progressing from basic, manual management processes toward a fully automated, intelligent, and adaptive state. It requires a commitment to regularly assessing your program's performance, identifying new risks, and adapting your security measures and policies to meet the future of identity head-on.

How to implement it:

- → **Measure your program with key metrics**: Track KPIs to measure the health and effectiveness of your IAM program. Key metrics include time-to-grant/revoke access, percentage of MFA adoption, number of standing privileged accounts, and the average time to complete access reviews.
- → **Conduct regular IAM risk assessments**: At least annually, perform a formal risk assessment of your identity program to identify new threats, evaluate the effectiveness of existing controls, and prioritize areas for improvement.
- → **Solicit feedback from business stakeholders**: Regularly engage with department heads and end-users to ensure IAM processes are enabling, not hindering, their work. Use this feedback to find and reduce friction.
- → **Stay informed about emerging threats and technologies**: Keep up with new attack vectors targeting identities (like AI-driven phishing) and new technologies that can improve your posture (like passwordless authentication methods and agentic identity governance).
- → **Maintain a multi-year strategic roadmap**: Develop and maintain a forward-looking roadmap for your IAM program that aligns with business goals and plans for future technology adoption, policy enhancements, and process improvements.

Improve your IAM with ConductorOne

Reading about IAM best practices is one thing; implementing them across a complex environment with legacy tools is another. Manual processes for access reviews, lifecycle management, and privileged access don't scale, leaving your team overwhelmed and your organization exposed.

ConductorOne is the agentic **identity governance platform** built to automate the complexity of modern identity security. We help you move from theory to reality by:

- → **Automating** time-consuming access reviews and providing reviewers with the AI-powered insights they need to make smart decisions.
- → **Eliminating** standing privilege with self-service, just-in-time (JIT) access that is easy for users and secure by default.
- → **Streamlining** the entire user lifecycle to ensure access is granted, modified, and—most critically—revoked in real-time.
- → **Governing** every identity in your environment, from human employees and contractors to the growing number of non-human service accounts, API keys, and AI agents.

See how ConductorOne can transform your IAM program and help you enforce a true Zero Trust strategy at scale.

Frequently Asked Questions (FAQs)

What is the most critical first step to modernizing our IAM program?

While every organization's journey is different, the most successful IAM modernizations begin with two parallel first steps: discovery and governance.

First, you must discover and audit what you currently have. You can't secure what you can't see, so a comprehensive review of all existing user accounts, permissions, and access policies is essential to understand your baseline and identify the highest-risk areas.

Second, you must establish a governance committee. A modern IAM program is a business-critical function, not just an IT project. Assembling a cross-functional team of leaders from IT, Security, HR, and key business units ensures you have the strategic oversight, buy-in, and authority needed to implement policies consistently across the entire organization.

What is the difference between IAM, IGA, and PAM?

These terms are closely related and often used together, but they refer to distinct disciplines within **identity security**.

→ **IAM (Identity and Access Management)**: This is the broad umbrella category. It encompasses the entire set of processes and technologies for ensuring the right entities (users and systems) have the right access to the right resources at the right time.

→ **IGA (Identity Governance and Administration)**: This is a subset of IAM that focuses on the "why." It answers the questions: Who has access to what, and should they? IGA is concerned with the policies, auditing, and review processes like access certifications, role management, and reporting to ensure access is compliant and aligned with business policy.

→ **PAM (Privileged Access Management)**: This is another specialized subset of IAM that focuses exclusively on securing access for the most powerful accounts in your environment—administrators, developers, and critical service accounts. PAM utilizes specific controls like session monitoring, credential vaulting, and just-in-time (JIT) access to protect these "keys to the kingdom."

How do these best practices apply to non-human identities (e.g., service accounts, API keys, AI agents)?

The same principles apply, but they must be adapted for machine-specific challenges. This ensures all user identities, whether human or not, are governed consistently.

→ **Lifecycle management**: Non-human identities also have a lifecycle. They must be provisioned securely, have their permissions reviewed regularly, and, most importantly, be de-provisioned the moment the application or service they support is retired.

→ **Least privilege (PoLP)**: This is even more critical for non-human identities, as they are a primary target for attackers. They should be granted only the specific permissions needed to perform their function, with no standing access to anything else.

→ **Ownership**: Every non-human identity must have a designated human owner who is responsible for it. This creates accountability and ensures that when an access review occurs, someone can confidently approve or deny its continued access.

How can we implement stronger security controls without creating too much friction for our end-users?

The goal of modern IAM is to make the secure path the easiest path. When implemented correctly, modern controls should **reduce friction**, not add to it.

→ **Single Sign-On (SSO)** eliminates password fatigue and the need for users to repeatedly log into different applications.

→ **Self-service workflows** for access requests and just-in-time (JIT) access are often much faster than traditional IT ticketing systems, allowing users to get the permissions they need in minutes, not days.

→ **Adaptive MFA** provides a seamless login experience for routine, low-risk access and only "steps in" with an MFA challenge when it detects a genuine risk, such as a login from a new device or an unusual location.

Our organization still relies on many legacy and on-prem applications. How can we integrate them into a modern IAM strategy?

This is a very common challenge. The key is to choose a modern identity platform that is built to bridge the gap between legacy and cloud environments. This is typically achieved through:

- → **Flexible connectors**: A robust identity platform should have a wide library of pre-built connectors and flexible APIs that can communicate with on-prem systems, whether they use standards like LDAP or proprietary methods.
- → **Identity gateways**: For applications that don't support modern standards like SAML or OIDC, an identity gateway can act as a bridge, translating authentication requests and enabling SSO.
- → **A phased approach**: Even if full SSO integration isn't immediately possible, you can still bring legacy apps under your governance umbrella by automating their user lifecycle (provisioning/de-provisioning) and including their entitlements in regular access reviews.

What are the key features to look for in a modern identity governance platform?

When evaluating a modern solution, look for a platform that moves beyond legacy approaches. Key features should include:

- → **Broad and deep connectivity**: The ability to connect to virtually any application or infrastructure, whether in the cloud or on-prem.
- → **Intelligent automation**: Deep, event-driven workflows that can automate the entire identity lifecycle, access reviews, and just-in-time provisioning.
- → **A user-centric experience**: An intuitive interface that makes it easy for end-users to request access and for managers to approve it, often integrated directly into tools they already use, like Slack or Microsoft Teams.
- → **Native Just-in-Time (JIT) access**: The ability to grant temporary, ephemeral access to critical resources as a core function of the platform.
- → **Comprehensive, audit-ready reporting**: The ability to generate reports for any compliance need with a single click, showing a complete audit trail of all identity activity.