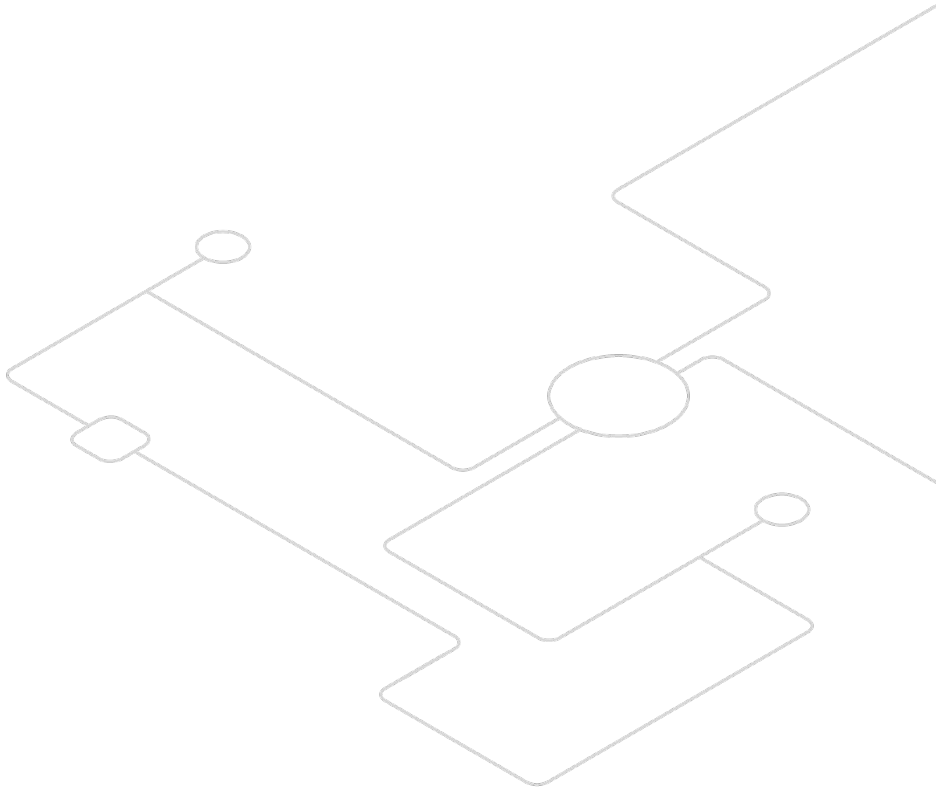
Secure Identity-Led Access to AWS, GCP, and Azure Instances Using Okta

okta

Index



Removing IAM barriers to cloud strategies

As more enterprises continue to move to the cloud, their investments in Infrastructure as a Service (IaaS) across Amazon Web Services (AWS), Google Cloud Platform (GCP), and Microsoft Azure rise as well. Whether it's to reduce costs or increase deployment speed and agility at scale, leveraging the cloud to power server applications and workloads can play a major role in helping organizations achieve their strategic objectives. But some of the biggest obstacles to fully embracing the cloud revolve around identity and access management (IAM).

Legacy IAM tools and software don't translate well to the cloud and it can be a significant challenge to figure out a cost-effective way to rearchitect IAM for new environments. Enterprises might also worry about the impacts on their existing environments if they move from their traditional IAM methods and procedures to more modern ones.

Of course, all major cloud providers offer their own IAM services, but under their shared responsibility models enterprises still own the heavy burden of trying to figure out how to use those services to keep everything in their cloud environment secure. The difficulty of that effort is substantially increased by the complexities inherent to the IAM services offered by the providers. But perhaps a bigger concern regarding those IAM services is that their proprietary nature can prevent enterprises from executing multi-cloud strategies.

> *A recent Forrester study indicated that 86% of enterprises have adopted multi-cloud strategies.*

Designed for elastic cloud infrastructures, Okta Advanced Server Access removes these barriers to cloud migration. It gives you a cloud-native approach to IAM that enables you to preserve the aspects of your IAM environment that you want to keep, while modernizing those aspects that you want to modernize. It allows you to enjoy the benefits of multifactor authentication, seamless single sign-on for all your workflows, as well as automated provisioning and deprovisioning of accounts. It simplifies and automates your ability to inject identity and access controls to your servers and cloud instances, therefore eliminating manual operations. Okta Advanced Server Access gives you a central control plane for access and a single source of truth for all your users and groups, unifying your identity across any cloud environment enabling you to seamlessly execute multi-cloud strategies.

Simplifying IAM for secure cloud infrastructure access

No matter the cloud provider, IAM can be a major challenge. Because of that difficulty, admins too often use a shared account or pass around shared credentials when spinning up infrastructure resources. But security-minded IT teams want better control and understanding of who's deploying what and who has the right levels of access. Traditional practices of pushing accounts and credentials through configuration management pose significant risk. A class of privileged access management products attempt to solve this risk, but they are heavy to operate and difficult to scale.

Trying to keep all your user accounts and credentials synchronized across all your servers with traditional methods requires a monumental, ongoing effort. Every new server spin-up requires a re-examination of what users and credentials need to be provisioned. When someone leaves the organization, identifying all the servers where their access needs to be deprovisioned can be an extensive, time-consuming, manual effort.

Okta Advanced Server Access provides a modern, highly secure and simpler approach. It utilizes a lightweight client application and server agent to leverage the core Identity services provided by Okta – Universal Directory, Lifecycle Management, Single Sign-On, and Multifactor Authentication - for seamless and secure server access. The unobtrusive and elegant design of the product eliminates the need for extensive custom scripting and configuration, credential vaulting and rotation, and systems operations. Once installed on a server, the agent performs all necessary configurations to support a modern, Identity-led workflow. This includes automatically creating and managing local user and group accounts, writing local entitlements for command-level permissions, and capturing and logging audit events.

How Okta Advanced Server Access works

Once installed and enrolled with Okta, each agent polls the backend API for any changes in user status, group membership, or group entitlements. As changes occur, the agent determines which changes are applicable, then applies those changes automatically to the local server. Since the server agent directly sources machine accounts from Okta's Universal Directory and ASA end-to-end Lifecycle Management capabilities, it eliminates the need for you to operate a local directory (like LDAP) to synchronize identities and accounts.

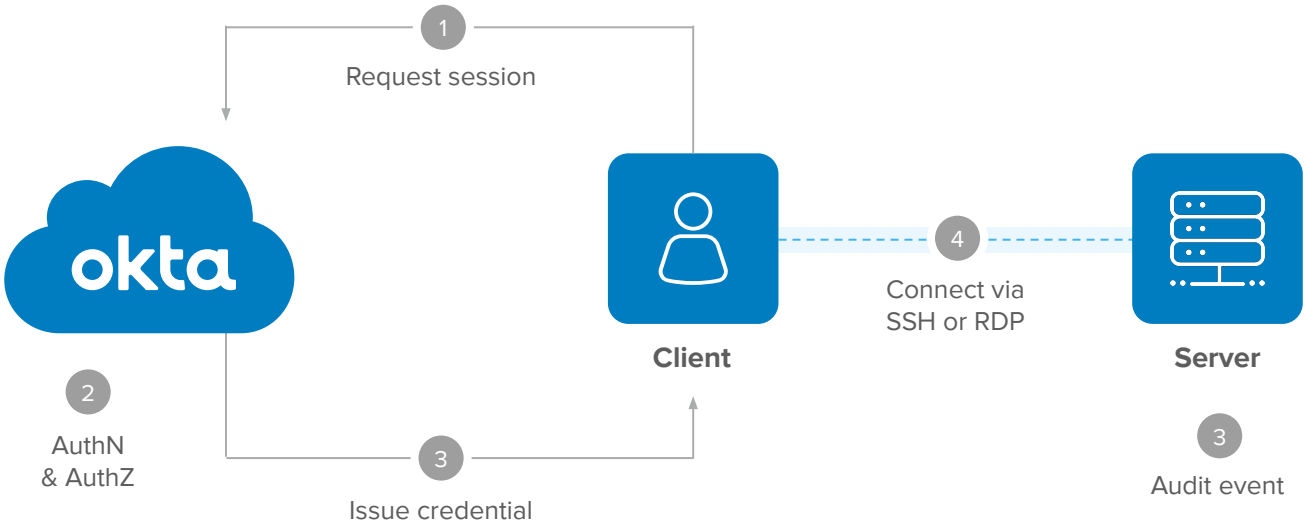
In addition, the server agent interacts with the following core components within Okta Advanced Server Access:

- **Advanced Server Access client application** gives you a lightweight desktop application that interfaces with your local SSH and RDP tools. It also provides a command-line tool for performing various configuration, administration, and user tasks.

- **Advanced Server Access programmable** certificate authority (CA) provides a mechanism for minting and issuing client certificates used for server authentication. For every authenticated and authorized request, the CA creates either an OpenSSH (Linux) or X.509 (Windows) client certificate on-demand. The ephemeral nature of each certificate eliminates the need for risky static credentials and chain of trust worries. Each minted certificate is temporary in nature and scoped to the servers within their associated project.

- **Advanced Server Access team** is the backing SaaS tenant which is mapped directly to your Okta Org for users and group assignments. Within a team is any number of projects, which represent the authorization scope of a group of servers and their role-based access controls. To support automation, the Okta API exposes all of the administrative functions available within the Advanced Server Access dashboard. Additionally, audit events are exposed through an API endpoint, as well through the ASA dashboard as structured objects or within Splunk via an Okta-supported plugin that directly synchronizes those events from your ASA tenant to your Splunk instance.

With the Client Application installed, the end user authentication workflow is a seamless Okta experience built natively into the SSH and RDP protocols.



1. Users login to a server using native SSH or RDP command-line or GUI tools, which are integrated with the Client Application to initiate the authentication process.

2. The user is authenticated via Okta Single Sign-On, and optionally Multifactor Authentication, based on customizable access policies. The authenticated user session is then bound to the user's device to establish a strongly authenticated user + device session. The login request is then independently authorized based on the role-based access controls of the Advanced Server Access Project that the specific target server belongs to.

3. Once fully authenticated and authorized, the backing Project CA mints a short-lived client certificate that is scoped to the authenticated user and device. Each certificate expires in 3 minutes by default, self revocating once used for the login request.

4. The client certificate is delivered to the Client Application. As part of an in-memory process on the user's workstation, the Client Application uses the certificate to authenticate to the target server. This authentication succeeds because the target server had been previously configured to support an additional authentication mechanism, specifically, a mechanism that trusts certificates signed by a CA hosted within Advanced Server Access.

5. The Server Agent generates an event containing user identity, time-of-day, authentication method, IP address, etc. This event is stored both on the local server and is delivered to the Advanced Server Access tenant. These events are consumable via API, or can be directly streamed to Splunk using an Okta-supported Splunk plugin.

How to enroll cloud instances with Advanced Server Access

In Advanced Server Access a server belongs to a project, which is done through an automated enrollment process. Similar to a domain in Active Directory or a realm in Kerberos, the Advanced Server Access project provides the authorization scope, associating a collection of resources with a set of configurations, including role-based access controls, entitlements, and access policies. To enroll a server with Okta, you can choose to use either the token enrollment method or the auto enrollment method.

Token enrollment method

Token enrollment supports the widest range of cloud and on-prem environments. It also gives you greater flexibility and customization than auto enrollment. The token itself is an arbitrary base64 encoded string that is stored in a local file on the server. The token does not contain any properties or metadata that require additional security controls.

You can generate a token for a server agent on-demand using the API or by using the Advanced Server Access dashboard. The token can be written manually to a local file on the server or it can be injected into machine images or automation scripts as part of a configuration management service such as Chef, Puppet, Ansible, or Terraform. Depending on whether it's a Linux or Windows server, the local file and path for the token will look like one of the following:

```
/var/lib/sftd/enrollment.token
```

```
C:\windows\system32\config\systemprofile\AppData\Local\ScaleFT
\enrollment.token
```

When a server agent starts up, the Okta backend services will simply recognize that agent's token as being associated with a project and the project's defined authorization scope.

Auto enrollment method

Auto enrollment is the easiest way to enroll server agents, since as the name suggests, it automates the enrollment process and bypasses the need to generate a token. To do this, it uses your cloud service account ID. For example, if you're using Amazon Web Services (AWS) you would enter your AWS account ID in the Advanced Server Access dashboard to associate it with a project. Likewise, if you're using Google Cloud Platform (GCP), you would enter your GCP account ID in the dashboard. So, when the server agent first starts on an instance within one those specific AWS or GCP accounts, the agent will automatically enroll with the respective project and its defined authorization scope. Auto enrollment uses the AWS and GCP public metadata API as the mechanism to associate the account ID and server agent with the defined project scope.

Even though auto enrollment is easier to use, it's more rigid in nature. First, it only supports AWS and GCP instances. Second, you can only assign one project to one account ID. So, if you only have a single AWS account ID, all of your AWS instances will be assigned to the same project. The same would be true for your GCP instances if you only have one GCP account ID. If that's not a problem, auto enrollment can be a great option. It can also work great if you have a multi-account strategy with either AWS or GCP, or both.

To learn more about server enrollment, visit the Documentation page:
https://help.okta.com/en/prod/Content/Topics/Adv_Server_Access/docs/setup/enrolling-a-server.htm

Automating server agent installation

Okta designed Advanced Server Access to support highly scalable, elastic infrastructure fleets through the use of automation. To make installation of the server agent as simple as possible, you can use the automation infrastructure deployment tools you're already using without requiring significant changes to your infrastructure code, including the following:

- **Configuration management** - If you're already using configuration management to spin up servers, automating the server agent installation can easily be done using Terraform, Chef, Puppet, Ansible, or SaltStack. Configuration management is the most flexible option since it allows you to make configuration changes on the fly, make API calls, and take advantage of more dynamic capabilities in the tools you use.

- **Machine images** – If you use cloned AWS, GCP, or Azure instances to deploy new infrastructure, these machine images can also be easily used as the mechanism for automating the server agent installation. The machine image method also aligns well with auto enrollment for your AWS or GCP environments. Since you are explicitly cloning a machine image, this method is more rigid in nature since it doesn't accommodate much additional configuration.

- **Multi-method installation** – You also have the option to use a combination of the configuration management approach and the machine image approach. In this scenario, the machine image would be used for standard pieces such as adding the package repository and configuring the server for client certificate authentication. Configuration management could use any provided inputs to perform dynamic configurations, such as project enrollment, bastion configuration, and more.

Whether you use configuration management, machine images, or a combination of both to install the server agent, each option is seamless in nature. No matter the method, you only need to inject a small install script into the process. Once you use one of these options to create a new instance, the server agent will be installed, the local configuration will be written to the machine, and the machine will be configured to trust client certificates signed by Okta as a viable authentication mechanism.

The most significant benefit of the automated installation of the Advanced Server Access server agent is that you eliminate the unsecure practice of pushing down the account and credential information needed for access management onto your machines since the server agent will enable you to use Okta as your backend identity store.

NOTE

Examples of the server agent install scripts used for the configuration management or machine image methods can be found on the following GitHub pages:

Linux - https://gist.github.com/fortyfivan/3b22cde2ad764363091f14e7648574ea

Windows - https://gist.github.com/fortyfivan/ac8ef6ea09a7ce691fee9dd307ab77ac

Deploying a bastion as a best practice

Not only does Advanced Server Access allow you to configure a server with a bastion architecture, but it's a recommended best practices and treats it as a first-class feature. When configured, the bastion establishes an encrypted and mutually authenticated SSH connection to the target server that the client application can hop through. The sshd process on the bastion opens a network connection to the target server, while the client application makes a logical connection.

Seamless and elegant way to extend IAM to cloud infrastructure

As you expand your cloud infrastructure, identity and access management needs to play a vital role in your long-term strategy for securing that infrastructure. Traditional challenges with IAM no longer need to be a barrier to secure cloud adoption and expansion. Built for the modern cloud, Okta Advanced Server Access gives you a seamless and elegant way to extend core Okta IAM services to your Linux and Windows servers.

Advanced Server Access facilitates your execution of multi-cloud strategies by offering you a unified identity layer across all of your cloud environments. It abstracts the complexities of IAM at scale across any cloud, public or private. It gives you a central control plane for access to Linux and Windows cloud instances through SSH and RDP. Its lightweight server agent approach makes enrolling and installing IAM on your cloud instances as simple as baking a few lines of bash or powershell into your automation. It automates your account lifecycle management, provisioning and deprovisioning local server accounts with Okta as your identity and access management source of truth. The Zero Trust access management infrastructure that Advanced Server Access delivers lets you quickly spin up workloads across AWS, GCP, or Azure with the assurance they're secure because you always know who has access to what and what they can do on your server instances.

To learn more about how Advanced Server Access makes it easy to extend Okta identity and access management to your cloud infrastructure, visit www.okta.com/products/advanced-server-access/.

About Okta

Okta is the leading independent provider of identity for the enterprise. The Okta Identity Cloud enables organizations to securely connect the right people to the right technologies at the right time. With over 6,000 pre-built integrations to applications and infrastructure providers, Okta customers can easily and securely use the best technologies for their business. Over 6,550 organizations, including 20th Century Fox, JetBlue, Nordstrom, Slack, Teach for America and Twilio, trust Okta to help protect the identities of their workforces and customers. For more information, visit us at www.okta.com or follow us on www.okta.com/blog.