Bridging the Gap Between AD and the Cloud

okta

Okta Inc.
100 1st St 6th floor,
San Francisco, CA 94105

info@okta.com
1-888-722-7871

Moving AD to the cloud with Okta

Active Directory (AD) has been around since 1999 when Microsoft first released it with Windows 2000 Server. Over the years it has become a key on-premises service for managing identity-based related activities. But with the rising demand for cloud services and apps, organizations have begun to realize that AD wasn't built for a cloud-centric world and today's use cases. Still, they want a way to leverage their investment in AD and also take advantage of the benefits of modern single sign-on (SSO) to get access to all their cloud and on-premises applications and resources through a single interface while reducing administrative overhead.

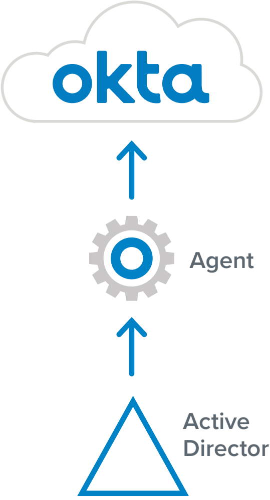
That's why forward-thinking IT organizations look to integrate their legacy AD environment with modern SSO from Okta and the Okta Identity Cloud. Integration with Okta provides a gateway to easily and securely connect your people to any service and resource they want to use. To help you plan for and implement that integration, this document gives you technical insights into functional and operational aspects of the AD to Okta integration.

Okta Active Directory Agent

At the heart of the AD to Okta integration is the Okta AD agent. It provides the interface between Okta and your local AD instance. The design of Okta and the agent is scalable in nature, making it easy to add more users and deploy more agents as needed. The agent employs secure outbound communication, provides load balancing and job management via long polling, and uses long-lived authorization refresh tokens.

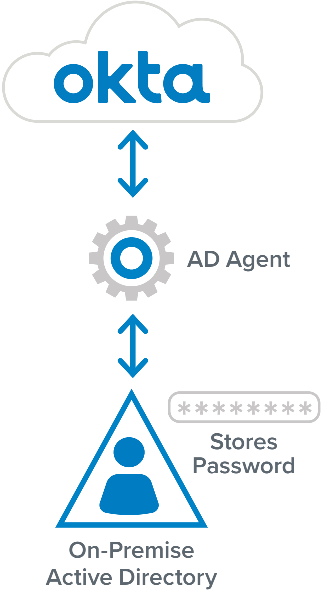
You install the agent on a dedicated local member server within an AD domain. In no case, should the agent be installed on a domain controller, since if you ever need to reboot or reinstall the agent it would impact the domain controller

.



AD Agent Integration Functionality

The AD agent provides two primary functions. The first of these is delegated authentication, which allows your on-premises AD instance to continue as your authentication source. With delegated authentication all passwords remain in AD and AD decides whether or not a user can gain access into Okta. Every time a user attempts to access a resource through Okta, Okta routes those authentication attempts to the on-premises AD directory. Once authenticated, AD allows Okta to grant the user access to Okta and the assigned resource.

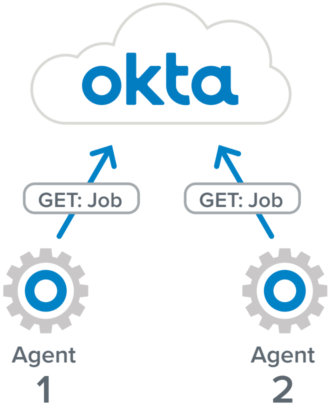


The agent's second primary function is to import users and groups with their associated attributes from AD into Okta. The import process creates a new user object in Okta for every user object that exists in the AD organizational units (OUs) that you specify. Okta then links together the respective AD and Okta users. This allows Okta to manipulate and enhance as needed the user attributes of the objects stored in its Universal Directory without affecting the user objects stored in AD. Bringing your users into Okta is also what allows you to take advantage of Okta's feature set, such as providing centralized SSO for any resources your users need, whether those resources reside on-premises or in the cloud.

Active/Active and Long Polling

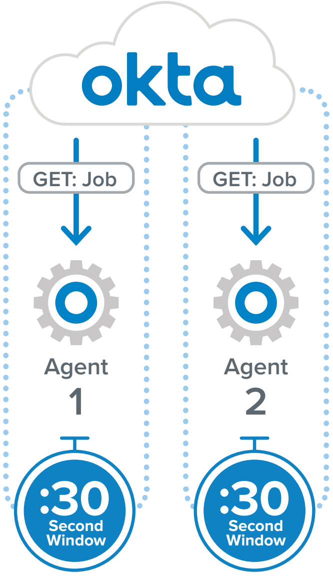
Okta deploys AD agents in an active/active configuration with no primary or secondary agents. As a result, there's no need to proactively load balance agents. All deployed agents are always active and use long polling to constantly look for new jobs as their capacity to handle those jobs allows.

So, instead of Okta waiting for requests from the agent as would happen in a traditional client server relationship, with our long polling model, Okta continually pushes jobs into a job pool and agents grab them as soon as they can. Since jobs get grabbed almost immediately, wait time is eliminated and jobs can be processed faster than would happen with a traditional model.



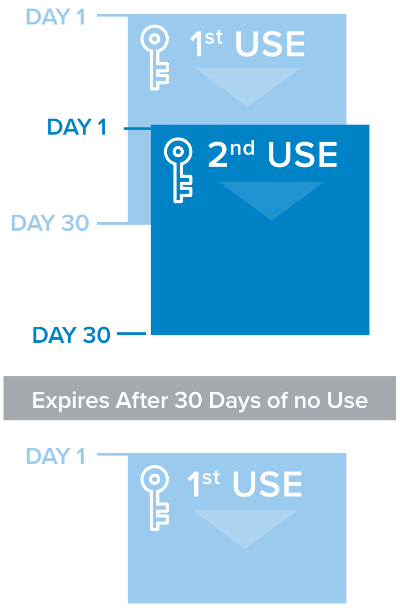
AD Agent Connection

The Okta AD agent creates an outbound HTTPS connection to the Okta Identity Cloud using certificate pinning with TLS server authentication. Each connection lasts no more than 30 seconds. During that connection time the AD agent will listen for events from the Okta service that it can process, such as AD authentication events. If such an event appears, the AD agent will grab it for processing and close the connection. The AD agent will then immediately open a new connection and listen again for new jobs. If no events appear during the connection's 30-second window, the agent closes the connection and initiates a new 30-second connection to listen for another job.



AD Agent Authorization

Communication at the authorization level occurs with the AD Agent using a bearer token, also known as an API token, to authorize against the Okta service's API. The agent passes that token within the HTTP authorization header with each request to Okta. The agent obtains the token through an OAuth flow during agent deployment. As part of the deployment process, an administrator will need to approve the token's registration. The bearer token has a 30-day life, but since it uses a sliding scale expiration the agent automatically renews the token with each request continuing its life as long as requests continue to occur.



AD Agent Job Management

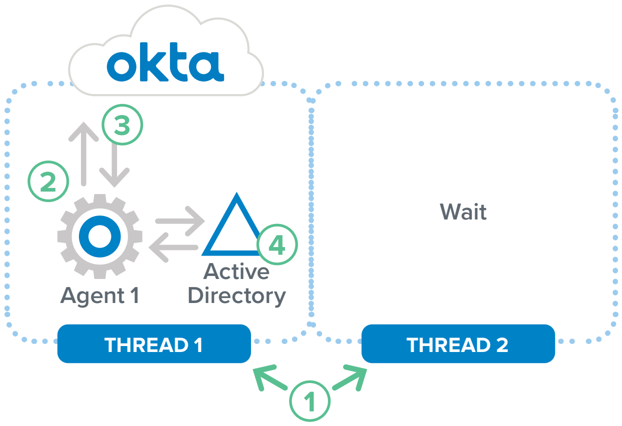
Since authentication jobs are frequent small jobs that need to be processed quickly, and import jobs are significantly larger jobs that take more time, Okta uses different workload distribution methods for each job type. This allows Okta to better accommodate the job types' inherent differences. Any available agent can process an authentication job. So, Okta randomly chooses an agent from a pool of available agents for each individual authentication job.

Import jobs use a preferred agent. While, the initial agent assignment is random, Okta will continue to use that same agent for subsequent tasks until that agent becomes overwhelmed or unavailable. At that point, Okta will assign a different agent to handle import tasks.

By default, Okta configures each agent with two threads. However, you can configure an agent with up to 10 threads if needed to address agent demand. Typically, one thread is always left available in an agent to handle authentication tasks. For example, when an agent gets issued an import job, Okta won't send any additional import jobs to that agent until that job completes, but the second thread of that agent will remain available to accept authentication jobs

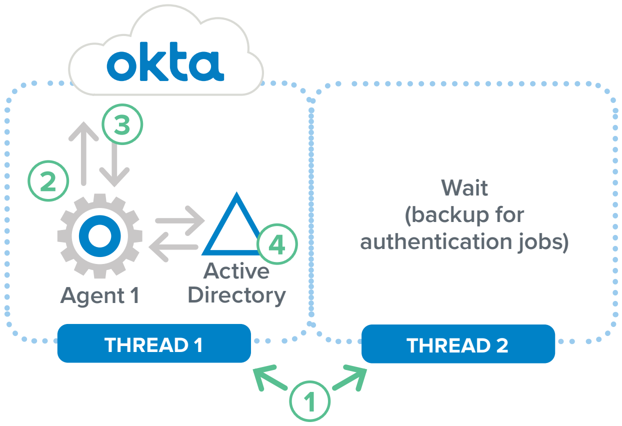
Example of an authentication job in action

1) Job thread remains open for 30 seconds
2) Agent performs HTTP GET to get a job from Okta
3) Agent receives an authentication job from Okta
4) Agent performs authentication job



Example of an import job in action

1) Job thread remains open for 30 seconds
2) Agent performs HTTP GET to get a job from Okta
3) Agent receives an import job from Okta
4) Agent performs import job
5) Agent becomes a preferred agent for import jobs



Planning your AD Integration with Okta

When planning and building out your integration it's important to understand the differences between how user authentications function versus user imports. For example, the frequency of these job types and their user bases have different impacts on the operability of your AD integration with Okta. Also, understanding when a user base will execute each job type is just as important, if not more important, than knowing the size of the user base. To help you with your integration planning and building efforts, the following provide some additional details on each job type, including differences on how each job type operates.

User Authentication Jobs

When users enter their credentials into the Okta User interface to authenticate, Okta determines if they are an AD mastered user. If they are an AD mastered user, Okta adds their authentication request to a job list associated with the user's directory instance. Once the authentication job is added to the job list, Okta will randomly assign the job to any available AD agent polling for jobs and then the assigned agent will open the job on server where the agent runs.

An AD mastered user is a user whose original attributes are owned by an AD instance. For more information on profile mastering, visit help.okta.com/en/prod/Content/Topics/Directory/eu-profile-masters.htm

The AD agent next performs a look up of the user using the username format specified in the AD integration settings, such as User Principal Name (UPN). When the agent finds the user, it uses the credentials entered by the user to perform a BIND to the AD instance. The following is an example of an UPN query that the agent may use to find a user:

**(&(sAMAccountType=805306368)
(userPrincipalName=user@domain.com))**

Compared to user imports, user authentication jobs execute quickly. The moment you finish setting up your integration, an agent can start performing a large number of authentications. As needed, you can deploy multiple additional agents as part of your integration to allocate more resources for authentications.

One reason for deploying additional agents for authentication might be to better handle the needs of seasonal trends. For example, retail stores often experience large upward fluctuations during the holiday season. The need for high availability is another reason organizations often deploy additional agents for authentication.

Additionally, global organizations often deploy additional AD agents in close proximity to each of their different geographic locations. Deploying an AD agent on a member server that has close proximity to a domain controller or a domain controller pool can reduce latency during authentication. As stated previously, it's important to make sure you only deploy AD agents on dedicated member servers and not on domain controllers.

User Import Jobs

User imports can be initiated by either a scheduled import or a manual import. Both methods create a job within Okta and assign that job to the preferred agent. As with authentication jobs, the AD agent reaches out to Okta, pulls down the job, opens it on the local server running the agent, and then executes the job. The agent first performs a topology read of the entire AD structure. The agent next updates users and groups, and in the case of an incremental import it performs the import based on the usnChanged AD attribute values it read for each object.* The scope of the objects it reads will be based on your AD agent's import settings that you configured in your Okta tenant's AD integration settings.

The following is an example of an import job query:

Topology Read:

```
(|(objectCategory=organizationalUnit)
(&(objectCategory=container)
(!showInAdvancedViewOnly=TRUE)))
```

User Account Read:

```
(&(sAMAccountType=805306368)
(uSNChanged>=889511)(uSNChanged<=889547))
```

It's important to remember that once a server runs an import, it becomes the preferred import server. That means subsequent import jobs will be sent to that same server. As a result, if you deploy AD agents on multiple servers, imports jobs will not be spread across all those servers. Instead, they will execute only on the current preferred server.

Additionally, it's important to understand that import speeds depend on the size of your directory, import scope, and frequency. For example, an import scope that contains 10,000 OUs with millions of users who have all been updated since their last import will take longer than an import of 5,000 OUs and users who have not been updated.

** Note: For more information on Microsoft AD's usnChanged attribute, visit* **docs.microsoft.com/en-us/windows/win32/adschema/a-usnchanged**

Real Time Sync Jobs

In addition to user authentications and user imports, the Okta AD agent also performs real-time sync jobs if you enable just-in-time (JIT) updates. Real-time sync jobs update user attributes, group memberships, and create new groups in Okta at the time of login. To do this, the agent performs a BIND to a randomly chosen AD server. Upon a successful BIND, the server receives a security token that contains the list of group security identifiers (SIDs) that the current user belongs to. The agent pulls the SIDs from this token, uses the SIDs to find the groups in its local SID cache, and sends the memberships to Okta to enable Okta to add the users to the groups. If the SIDs are not present in the local cache the agent will search for them in AD.

Installing and Troubleshooting Your AD Agent

Detailed requirements, procedures, and tasks for installing your AD agent for an AD to Okta integration can be found at **help.okta.com/en/prod/Content/Topics/Directory/ad-agent-install.htm**. To assist you with troubleshooting efforts, Okta records in the AD agent's logs and the Windows Event viewer actions that the AD agent performs. The AD agent logs are stored in C:\Program Files (x86)\Okta\Okta AD Agent\log and in the Windows event viewer. If additional logging information is desired, verbose logs can be enabled. For more information, please visit **support.okta.com/help/s/question/0D50Z00008G7UppSAF/how-can-i-enabled-verbose-logging-in-my-ad-agent**.

Since the AD agent communicates with Okta through HTTPS, you can capture this traffic using web monitoring tools such as Fiddler. For more information on how to setup Fiddler, please visit **support.okta.com/help/s/article/Capturing-A-Fiddler-Trace-For-Okta-Customer-Support**.

Integration Harmony

Integrating AD and Okta with the Okta AD agent enables you to take full advantage of modern applications, including mobile apps, SaaS apps, and on-premises apps. By adding these apps to Okta, you automatically give your AD users access to them. Okta also makes it easy to secure access to these apps through a central set of policies, adaptive multi factor authentication (AMFA), and even the option to go passwordless. By taking advantage of our Okta RADIUS agent and Okta LDAP interface, you can even extend some of Okta's security capabilities to local network equipment, such as Wi-Fi routers, VPNs, and other network appliances.

With centralized identity access management (IAM) and a single login interface for your users to access all the apps and services they need, you significantly reduce administrative overhead, speed up app rollouts, provide users more seamless and consistent access, and strengthen your overall security posture with more modern authentication capabilities.

AD has been an integral part of your environment in the past and can continue to provide value in the future. But as you take advantage of the many benefits of modern IAM with Okta through this integration, the way you view AD's role may change. Okta opens a gateway to easy and secure access to the growing array of modern apps and services. With our vendor neutral nature and integration network, you can connect any technology through Okta, giving you the flexibility you need to drive your business innovation and success.

To learn more about how you can benefit from an AD integration with Okta, visit www.okta.com/rethinkad.

About Okta

Okta is the leading provider of identity for the enterprise. The Okta Identity Cloud connects and protects employees of many of the world's largest enterprises. It also securely connects enterprises to their partners, suppliers and customers. With deep integrations to over 6,000 apps, the Okta Identity Cloud enables simple and secure access from any device. Thousands of customers, including Experian, 20th Century Fox, LinkedIn, Flex, News Corp, Dish Networks and Adobe trust Okta to work faster, boost revenue and stay secure. Okta helps customers fulfill their missions faster by making it safe and easy to use the technologies they need to do their most significant work. For more information, visit us at www.okta.com or follow us on www.okta.com/blog.