

Okta Technical Consultant Boot Camp 2

Student Guide

Copyright 2017 Okta, Inc. All Rights Reserved.

Window captures and dialog box sample views are the copyright of their respective owners.

Use of this user documentation is subject to the terms and conditions of the applicable End-User License Agreement.

Printed April 2017

Contact Okta, Inc @ training@okta.com

Table of Contents

Introduction	
About the Course	2
Course Flow	3
Session Schedule	4
Module 1: Configure Application Masters	
Configure Application Masters Objectives	8
Scope the Business Requirements	11
Design the User Schema	13
Configure an Application Master	15
Lab 1-1: Configure an Application as a Master	26
Best Practices	27
Troubleshooting	28
Lab 1-2: Configure an Application as a Master: Troubleshooting	29
Module Summary	30
Module 2: Scope Multiple Forest and Multiple Domain Environments	
Scope Multiple Forest and Multiple Domain Environments Objectives	32
Describe Business Requirements for Different Domain Environments	34
Describe Active Directory and DSSO Agents in Complex Domains	36
Describe High Availability in Large Deployments	39
Best Practices	41
Troubleshooting	42
Module Summary	43
Module 3: Configure Inbound Federation	
Configure Inbound Federation Objectives	46
Scope the Business Requirements	47
Describe the Okta Solutions	48
Configure Inbound Federation	56
Lab 3-1: Configure Inbound Federation Using SAML	57
Lab 3-2: Configure Inbound Federation Using SAML and Org2Org	60
Best Practices	61

Troubleshooting	62
Module Summary	63
Module 4: Deploy Advanced SSO Solutions	
Deploy Advanced SSO Solutions Objectives	66
Describe Reverse Proxy Flows for Header-based Authentication	68
Configure Advanced SAML Configurations	85
Lab 4-1: Encrypt a SAML Assertion	88
Configure RADIUS Authentication	92
Lab 4-2: Deploy the Okta RADIUS Agent	103
Best Practices	104
Troubleshooting	105
Module Summary	106
Module 5: Develop Applications with APIs	
Develop Applications with APIs Objectives	108
Scope the Business Requirements	110
Scenario: Populate a CADD	121
Implement Okta API and SDK Solutions	122
Best Practices	137
Module Summary	140
Module 6: Configure API as Masters	
Configure API as Masters Objectives	142
Scope the Business Requirements	144
Describe the Okta Solutions	154
Configure API Masters	155
Lab 6-1: Configure the Marketing Structure for Contractors	159
Lab 6-2: Configure API as a Master	160
Lab 6-3: Onboard Users with API as a Master	161
Lab 6-4: Update Users with API as a Master	162
Lab 6-5: Deactivate and Delete Users with API as a Master	163
Best Practices	164
Troubleshooting	165
Module Summary	166

Module 7: Create a Custom Sign In with the Okta Sign-in Widget

Create a Custom Sign In with the Okta Sign-in Widget Objectives	168
Describe the Okta Sign-in Widget Features	170
Develop Authentication Workflows	172
Configure the Okta Sign-In Widget	173
Lab 7-1: Configure the Okta Sign-in Widget	177
Best Practices	178
Troubleshooting	179
Module Summary	180

Module 8: Configure Authentication with Multiple Providers

Configure Authentication with Multiple Providers Objectives	182
Scope the Business Requirements	185
Describe the Okta Solutions	186
Configure Multiple Provider Authentication	191
Demonstration: Perform IdP Discovery	192
Lab 8-1: Configure IdP Discovery	193
Lab 8-2: Test the IdP Discovery	194
Best Practices	195
Troubleshooting	196
Module Summary	197

Module 9: Deploy Lifecycle Management for Custom Applications

Deploy Lifecycle Management for Custom Applications Objectives	200
Describe Lifecycle Management	202
Group Discussion	211
Deploy Native SCIM	212
Lab 9-1: Launch and Test the SCIM Server	242
Lab 9-2: Define a Native SCIM Application in Okta	243
Lab 9-3: Extend Native SCIM with Custom Attributes	244
Deploy On Premise Provisioning	241
Lab 9-4: Deploy an On Premise Provisioning Connector	258
Lab 9-5: Install and Configure the Okta Provisioning Agent	259
Lab 9-6: Integrate the Custom Application Provisioning	260
Best Practices	261

Troubleshooting	263
Module Summary	264
Module 10: Configure OpenID Connect	
Configure OIDC Objectives	266
Describe OAuth 2	269
Describe Claims and Scopes	284
Describe OAuth Flows	285
Describe OIDC	291
Describe Social Authentication	295
Deploy SSO with OIDC and the Auth SDK	300
Lab 10-1: Implement Social Authentication with Facebook	305
Lab 10-2: Create an OIDC Application Using the AIW	318
Lab 10-3: Test the OIDC SSO	319
Lab 10-4: Configure API AM	320
Lab 10-5: Test API AM Requests	321
Lab 10-6: Enable the Development Team	322
Module Summary	323
Summary	
Okta Learning Paths	326
Okta Certification	327
Course Summary	328

okta

Okta Technical Consultant Boot Camp 2





About the Course

Title: [Okta Technical Consultant Boot Camp 2](#)

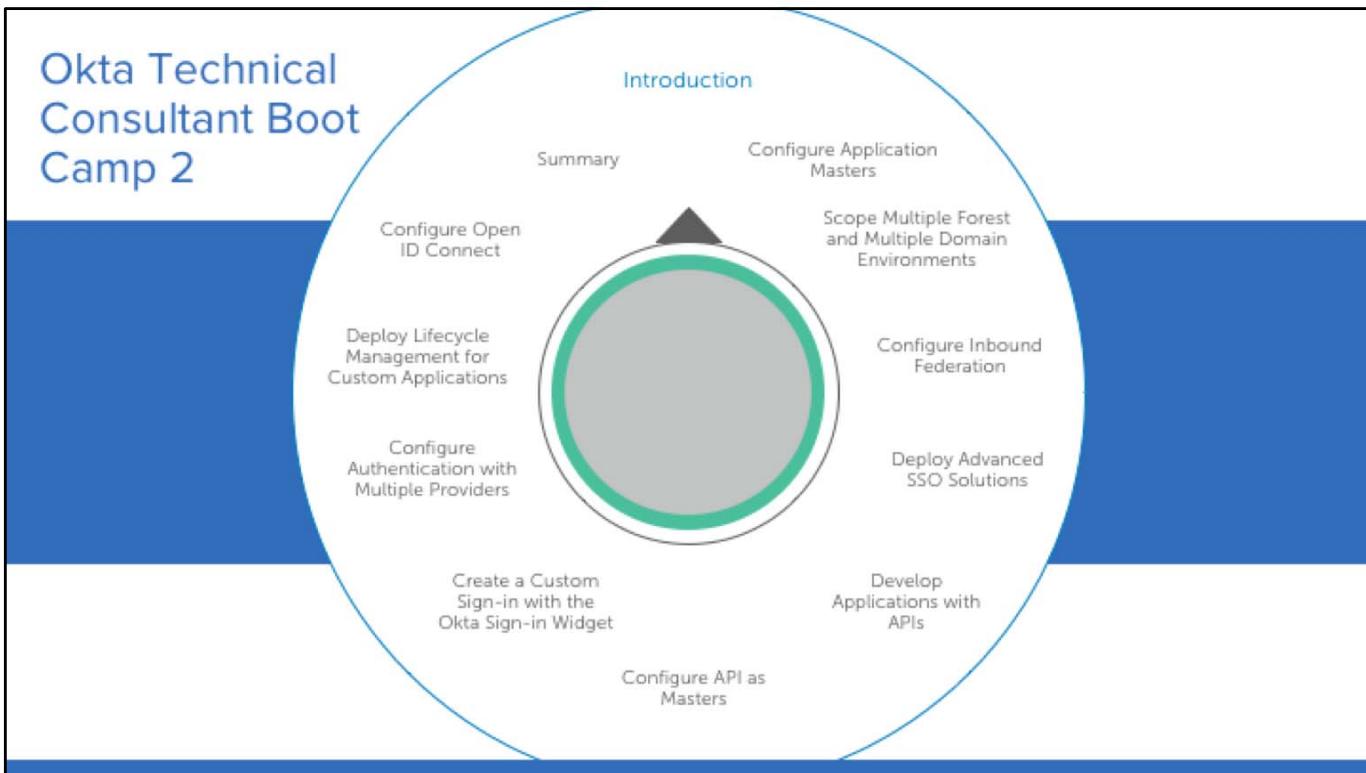
Duration: 5 days

Prerequisites:

- Partner Sales Foundation (recommended)
- Okta Technical Consultant Boot Camp I (recommended)
- Familiarity with Active Directory concepts
- Basic understanding of Identity and Access Management (IAM)

Hands on labs:

- Lab guide
- Unique Virtual Machines
- Unique credentials



Session Schedule

Day One	Time (in minutes)
Introductions	30
Configure Application Masters	150
Meal Break	60
Configure Application Masters - Continued	90
Scope Multiple Forest and Multiple Domain Environments	90

Day Two	Time (in minutes)
Configure Inbound Federation	180
Meal Break	60
Configure Inbound Federation - Continued	70
Deploy Advanced SSO Solutions	135

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Session Schedule Continued

Day Three	Time (in minutes)
Develop Applications with APIs	90
Configure API as Masters	90
Meal Break	60
Configure API as Masters - Continued	30
Create a Custom Sign-in with the Okta Sign-in Widget	120
Configure Authentication with Multiple Providers	100

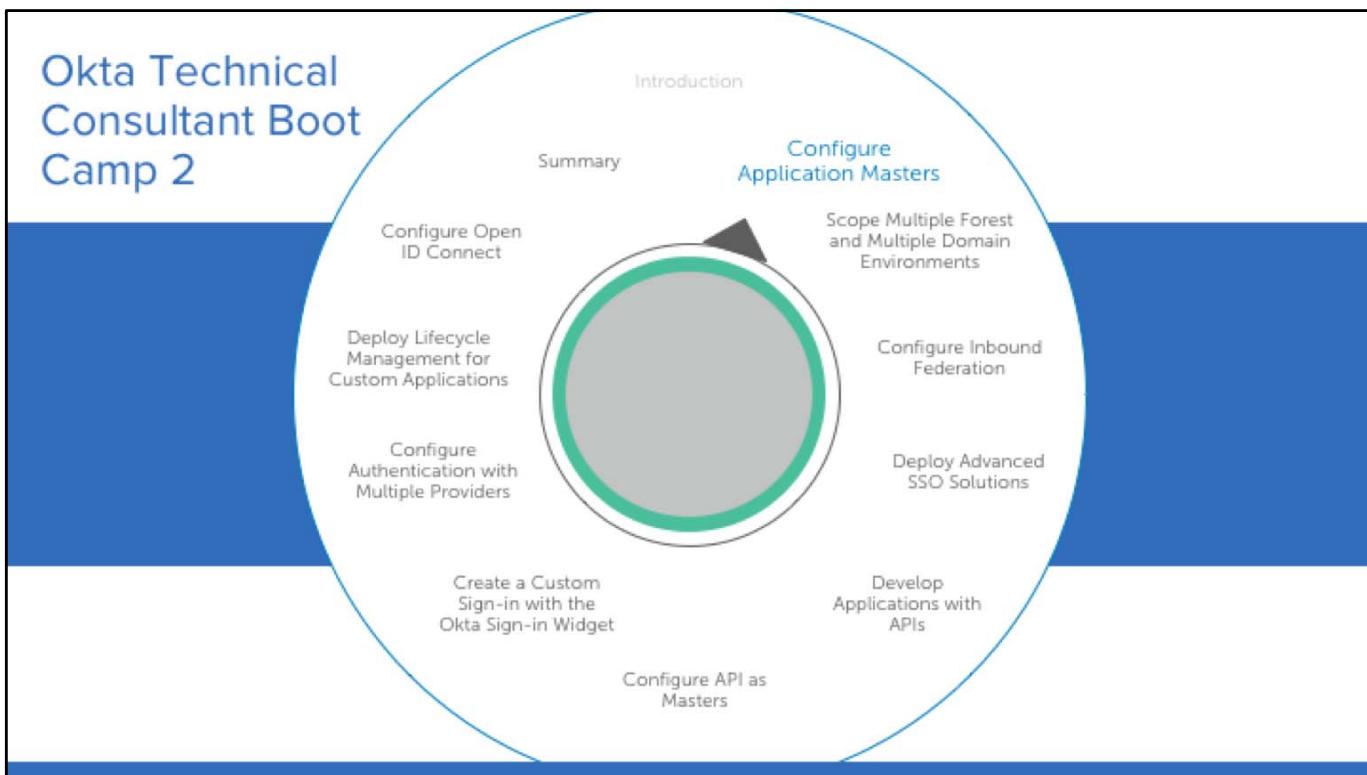
Day Four	Time (in minutes)
Deploy Lifecycle Management for Custom Applications	180
Meal Break	60
Deploy Lifecycle Management for Custom Applications - Continued	180

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Session Schedule Continued

Day Five	Time
Configure Open ID Connect	180
Meal Break	60
Configure Open ID Connect - Continued	180

© Okta and/or its affiliates. All rights reserved. Okta Confidential.



Configure Application Masters

Okta recognizes that all user attributes exist in every user store - there are applications with specific attributes for a complete user profile. Using Universal Directory in Okta, you can establish a custom user profile through more granular attribute-level mastering.



Configure Application Masters

Scope the Business Requirements
Design the User Schema
Configure an Application Master

Configure Application Masters Overview

In this module, you will work through the phases of the as a master process from overviewing the application to getting the application live for the customer.

This module consists of a lab and review questions.

Application Masters Overview

Enables you to...

- Help Okta customers plan for, design, and properly configure an application as a master based on business requirements

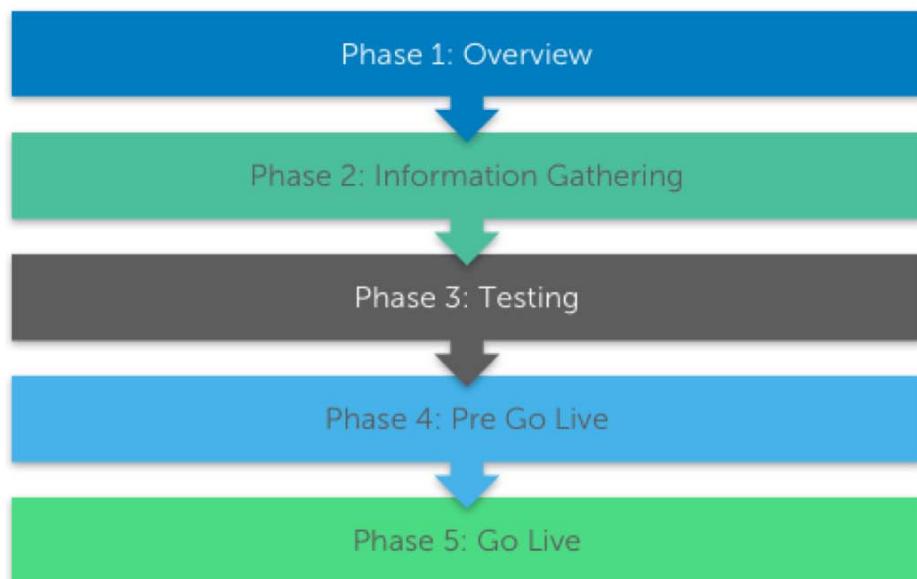
Is important because...

- You work with different Okta customers with different user stores and requirements.
- You need help customers configure Okta based on company structures and policies.

Additional Information

Within Okta, you can establish an application as the master, or source of truth, for some or specific user attribute information. Before configuring attribute information, you must fully scope the application and customer business requirements.

Phases of As-a-Master



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

While every customer, every application, and every as-a-master implementation is different, there is a standard flow and recommended practice to follow.

Phase 1: Overview

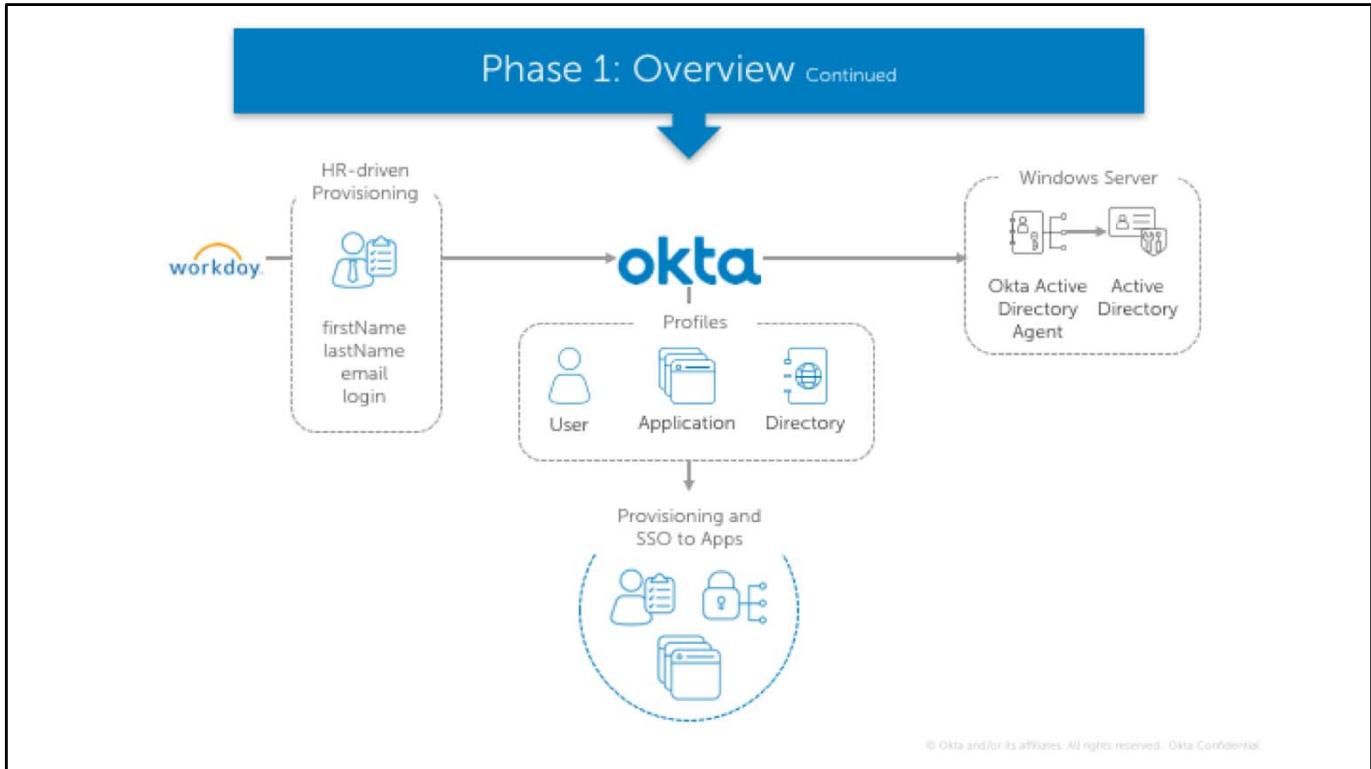
- Review the solution with the customer:
 - Explain limitations
 - Answer questions
 - Set expectations
 - Define high-level requirements
 - Show the Okta and as-a-master benefits
 - Conduct meetings with the proper people

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In the overview phase of as-a-master, you will spend a lot of time with the customer. In this phase, review the * as a master solution with the customer to:

- Explain what you can and cannot do; level set expectations.
- Answer any questions from the audience
- Set expectations on timelines based on resource availability on the customers side. This is a critical step because HR needs to see exactly how resources are impacted.
- Define any high level requirements which will become more detailed in the next phase.
- Show off the benefits of Okta and * as a Master.
- Make sure all the right parties are involved in overview meetings because excluding a group such as HR and only including IT can set you up for later complications.



Additional Information

Using information from <https://www.okta.com/products/lifecycle-management/> and the slide image, explain how an as a master solution provides:

- **New user and account update flows:** as users are created and updated in an HR system, Okta imports and creates or updates the users in Universal Directory and other provisioning-configured downstream applications.
- **Account terminations:** as users are deactivated in an HR system, the account is deactivated in Okta and disabled in provisioning-configured downstream applications, including external directory services such as Active Directory.

To create new users in Okta, the firstName, lastName, email, and username attributes must contain values. Depending on the timing of the business process integration, some customers might not yet have valid emails configured which can cause issues with provisioning to other applications. If possible, use Universal Directory to automatically create email addresses during the HR to Okta import or creation.

While the slide shows Active Directory, some customers do not have an external directory service making Okta the authentication provider for all users.

- Different HR systems use different group types. Workday uses provisioning groups enabling HR to manage users in groups directly in Workday.
- Bamboo/Ulipro create some groups as part of provisioning, while other applications such as SuccessFactors rely on Okta groups.

Phase 2: Information Gathering

Existing Infrastructure and Process

- How many Active Directory domains must be integrated with Okta?
- How many total Active Directory users?
- Which permissions are currently enabled/which permissions are required for the Okta Active Directory agents?
- What are the existing new hire processes?
- When is the email generated for the new hire and who generates the email?
- Do you have a separate OU structure for full time employees and contractors in Active Directory?
- How many different locations (Active Directory OUs) do you use for user management?
- Do contractors get accounts in Active Directory?
- What are the formats of the UPN, sAMAccountName, and email attributes?
- Do you have an Okta Preview site?
- Do you have an Active Directory sandbox?

Additional Information

To understand the customers requirements, environment and processes many questions need to be asked. At a high-level, you must get a picture of the current state, which helps the project direction of the future state. Username in HR system = Username in Okta = Username in Active Directory (If applicable)

- Import times required to meet business needs
- Creation of users before start date (If HR application provisioning supports it)
- Base and custom attributes:
 - All * as a Master solutions in Okta pull in a base set of attributes.
 - Additional attributes are configured in applications individually. The application deployment guide will highlight the steps to include custom attributes.
- Clean Data: Important to explain that as a master, the application becomes the system of record after mastering is enabled. Data in the source application must be current. The customer is responsible in making sure the correct data is in the source, but you might need to remind the customer several times before go live.
- Syncing or write back:
 - Real-time synchronization is a feature in some as a master solutions.
 - Write back of email and phone number is supported by some as a master solutions.

Phase 2: Information Gathering

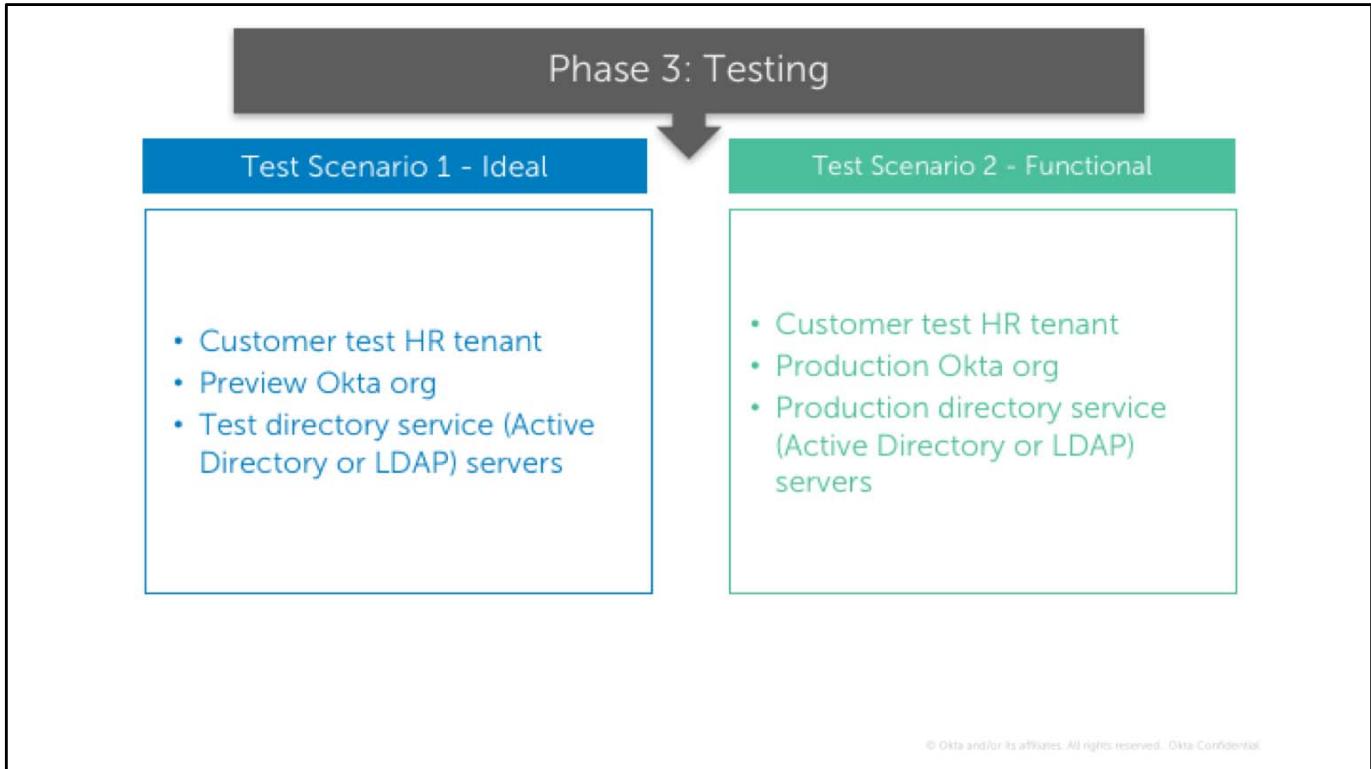
Existing Workday Configuration

- How many active users in Workday?
- Is the user data synchronized between Workday and Active Directory?
 - If not, which attributes are not synchronized?
- Is the user data in Workday accurate?
- What are the existing new hire and termination processes?
- Is the email address populated in Workday when the New Hire is created?
 - If not, what is the current HR process around this?
- What is the current username format in Workday?
- Are you using Contingent workers?
- Do you have a workday sandbox?

Additional Information

To understand the customers requirements, environment and processes many questions need to be asked. At a high-level, you must get a picture of the current state, which helps the project direction of the future state. Username in HR system = Username in Okta = Username in Active Directory (If applicable)

- Import times required to meet business needs
- Creation of users before start date (If HR application provisioning supports it)
- Base and custom attributes:
 - All * as a Master solutions in Okta pull in a base set of attributes.
 - Additional attributes are configured in applications individually. The application deployment guide will highlight the steps to include custom attributes.
- Clean Data: Important to explain that as a master, the application becomes the system of record after mastering is enabled. Data in the source application must be current. The customer is responsible in making sure the correct data is in the source, but you might need to remind the customer several times before go live.
- Syncing or write back:
 - Real-time synchronization is a feature in some as a master solutions.
 - Write back of email and phone number is supported by some as a master solutions.



Additional Information

When working through the test phase, be ready with the ideal scenario, but also provide a functional one. In the ideal scenario, you have accounts for every item in place that must be tested. In the functional scenario, you are limited on testing because you cannot use production accounts in Okta or the directory service - you must use test users only.

To perform the tests, setup the following:

1. Okta - might already be done
2. Active Directory - might already be done or not a factor
3. Provisioning tab in the as a master app; configure all options and refer to the application deployment guide
4. Create users section under Active Directory Settings tab - this is required to push users to Active Directory
5. Group methodology - map user creates to Active Directory OUs
6. Attributes
7. (Optional) Real-time sync
8. (Optional) Write back to * as a master application

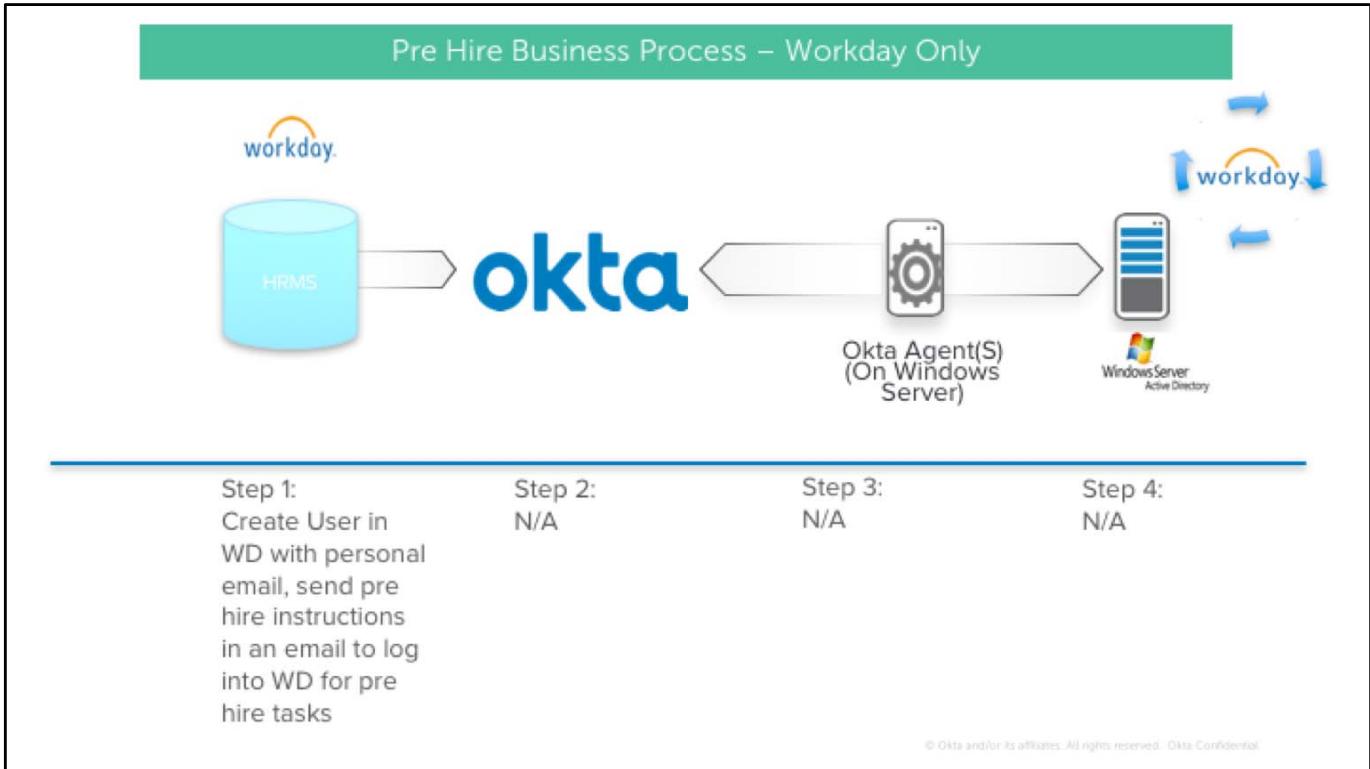
Phase 3: Testing

Use Case Testing

Use Case Testing

- With the environment configured, run thru use cases thoroughly testing the following:
 - Pre Hire
 - Hire user
 - Update user attributes
 - Terminate user
 - Attribute mappings
- Fix any issues found

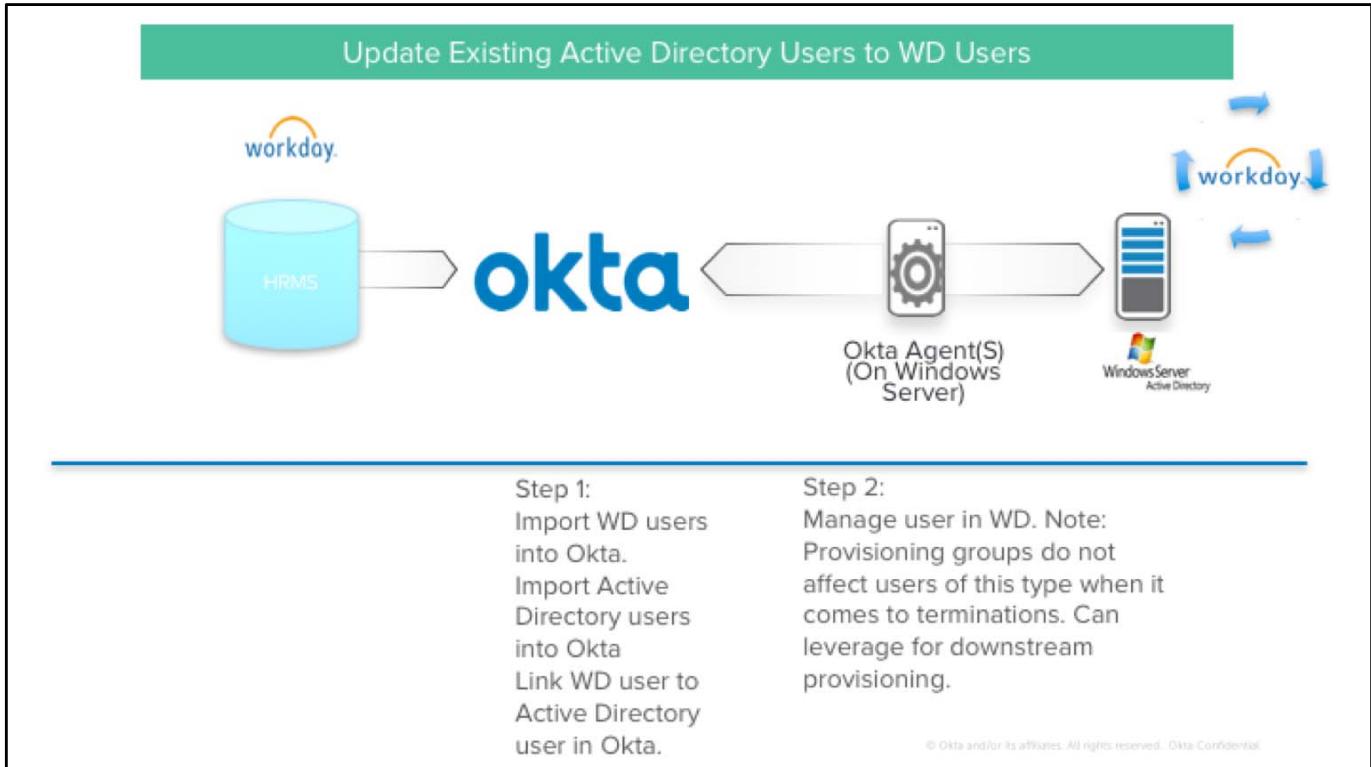
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

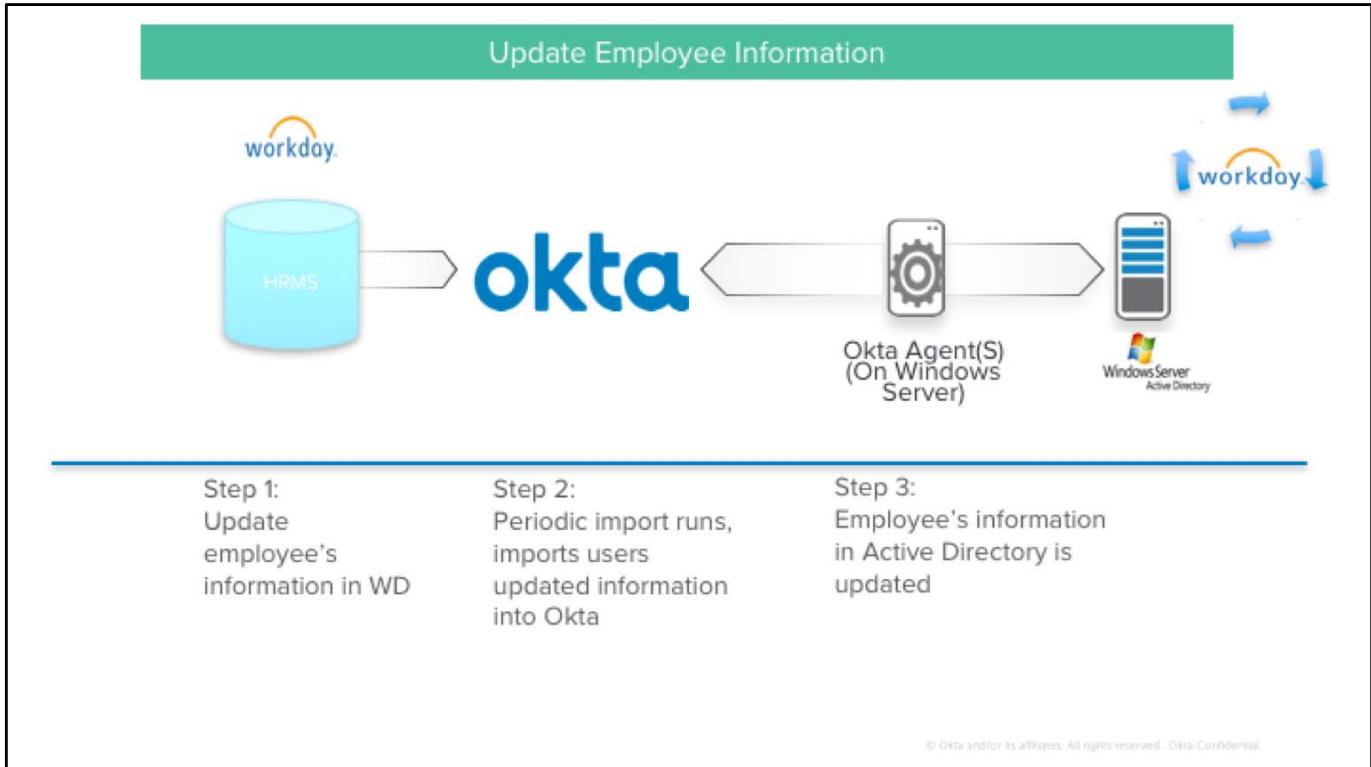


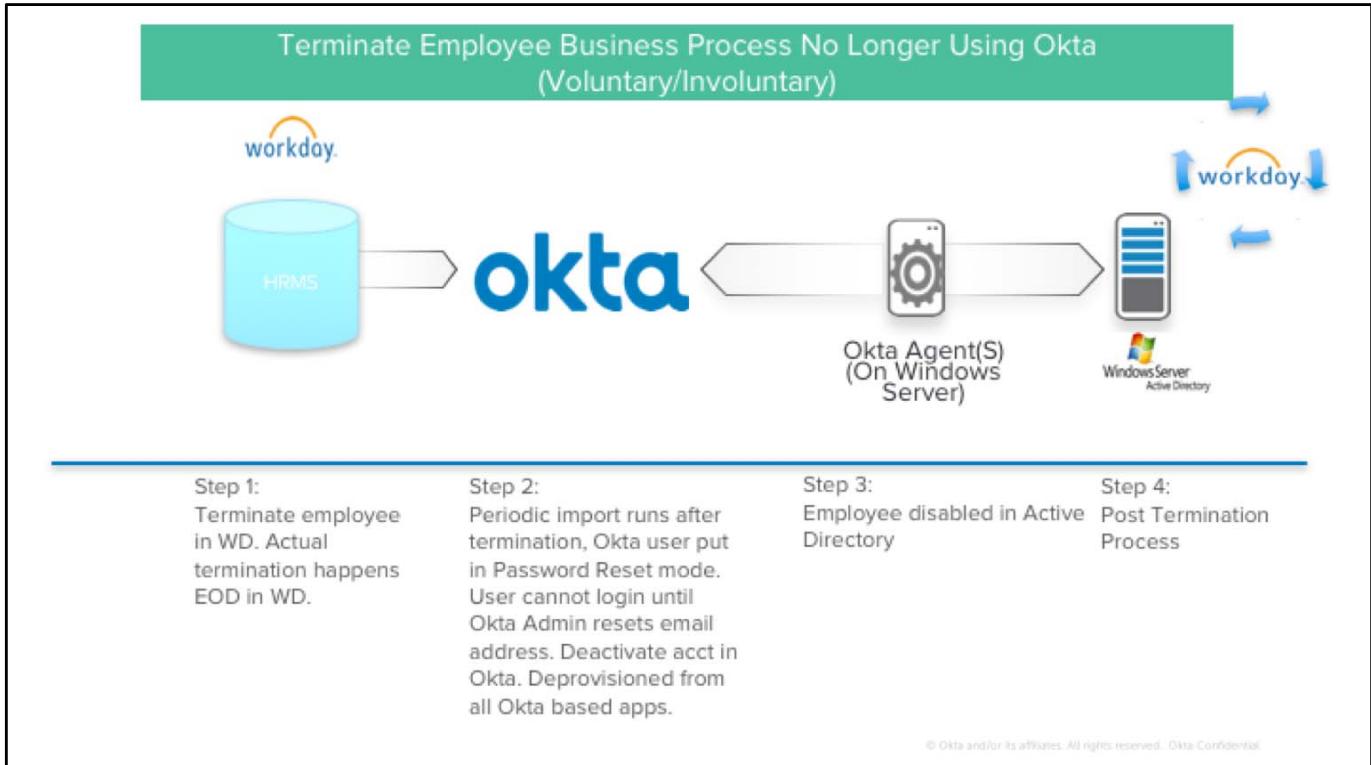
Additional Information

Creating pre hire username consistent with Active Directory naming standards at this point makes hire transition easier and less confusing for employee.

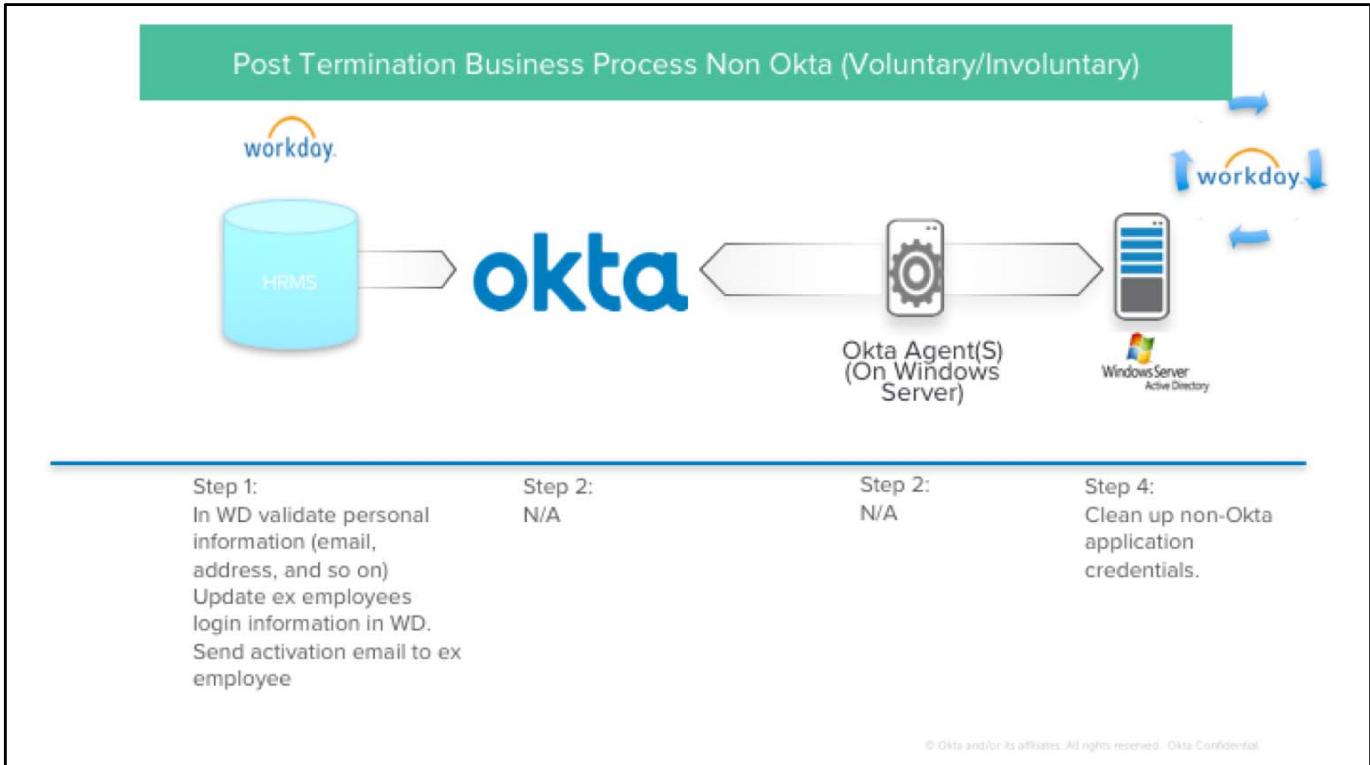












Phase 4: Pre Go Live

- As soon as testing is accepted with the application provisioning test cases:
 1. Import and activate existing directory (Active Directory or LDAP) service accounts.
 2. Configure provisioning on the administrator account and set permissions in the production tenant.
 3. Create groups.
 4. Configure custom attributes.
 5. Test API user, password, endpoints, or custom reports (Workday only).
 6. Run user import.
 7. Configure application provisioning in Okta.
 8. Verify account imports.
 9. Match application provisioning settings.
 10. Match Universal Directory mappings between Preview and Production.
 11. Map provisioning groups to the directory service.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

1. Import and active external users: All Active Directory users that exist in the as a master application must be activated in Okta. Make sure all OUs are being synchronized with Okta because end users are not notified. The import is done manually in Okta on the Active Directory instance.
2. In the as a master application, you must setup of provisioning the administrative user and associate permissions in the production tenant before configuring provisioning in Okta production.
3. For groups creation, make sure the required provisioning groups are ready (Workday) and configure any required groups in Okta.
4. If using Workday. make sure existing employees are in the correct provisioning groups if you want to use provisioning groups as part of application assignments.
5. Configure custom attributes as necessary.
6. Test API user, password, endpoints, and (Workday only) Custom Reports.
7. Run an import verify all users are visible in Okta
8. Configure application provisioning in the Okta production org.
9. Pre-check all users on the import tab for accuracy; partial matches and so on.
10. Match all settings on the application Provisioning tab.
11. Match all Universal Directory mappings between preview and production.
12. Map Workday provisioning groups to Active Directory; disable Workday provisioning and then re add users the application before go live.

Phase 5: Go Live

1. Verify no users appear on the Assignments tab of the application.
2. Re-enable application provisioning if it was disabled in previous step.
3. Verify import matching automation is still disabled before running the import.
4. Run the application import.
5. Verify all users/provisioning groups are imported and still mapped to Active Directory OUs.
6. Match some test users to verify attribute mappings is working correctly.
7. Troubleshoot any provisioning issues.
8. Match existing Active Directory and Okta users on the application Import tab.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

At go live time perform the following in order:

1. Make sure there are not any users are on the Assignments tab of the application; this reflects a manual user assignment.
2. If necessary, re-enable application provisioning.
3. Verify import matching automation disabled before running the import.
4. Run the application import.
5. Verify all users/provisioning groups are imported and still mapped to Active Directory OUs.
6. Match several test users to verify attribute mappings are working correctly. One or more test users might be a new user coming from Workday, otherwise create a new test user to emulate.
7. Troubleshoot any provisioning errors that occur.
8. On the application Import tab, match existing Active Directory users with existing Okta users in the following order:
 - a. Exact Matches
 - b. partial matches
 - c. No matches



Configure an Application as a Master

- Create and Configure an Okta Admin Account
- Install the Okta AD Agent and Import Users
- Create Groups and Configure Group Rules
- Add SFDC as a Master
- Verify Configurations



- Understand the customers existing technical environment
- Understand the business process flow especially if implementing an HR as a Master solution
- Data mappings and transformation are critical to a successful implementation



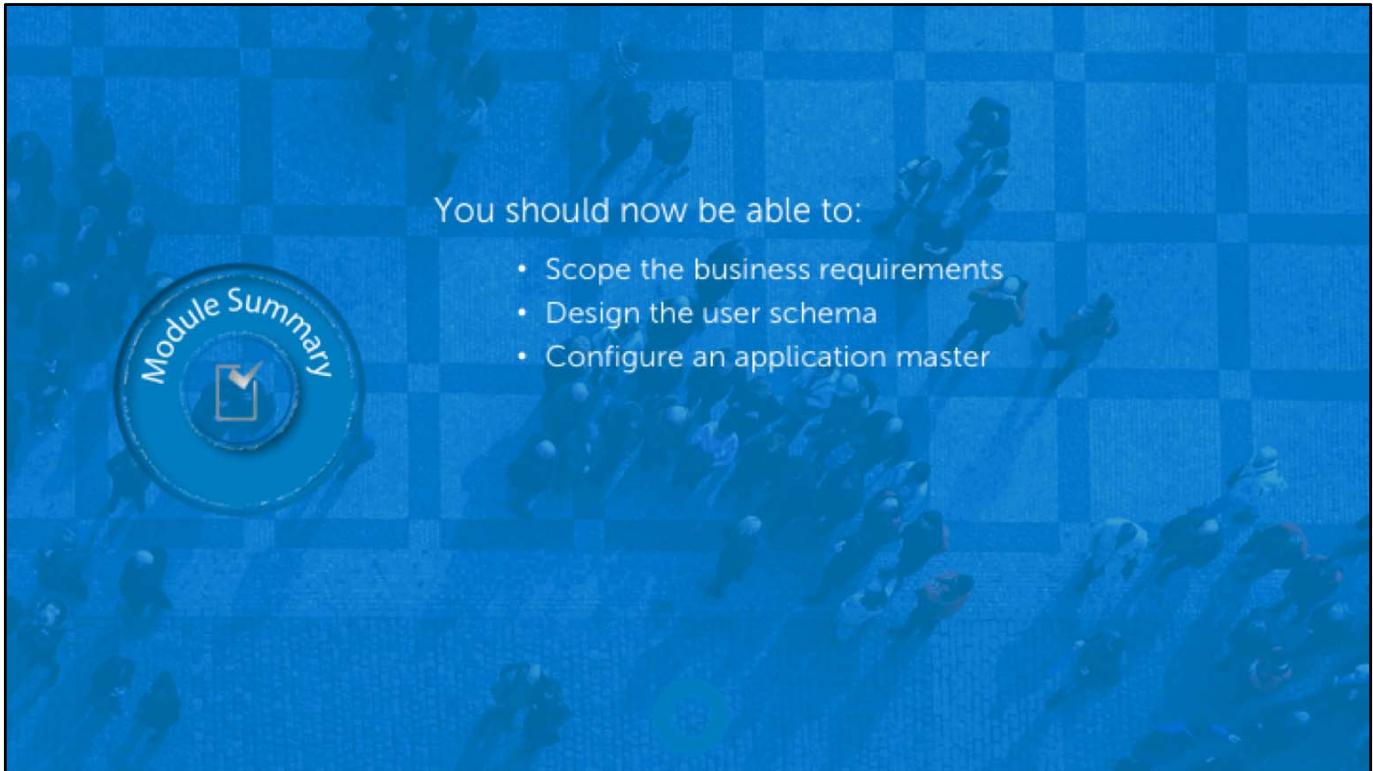
When troubleshooting application master issues:

- Verify the data mappings and transformations.
- Verify the profile master priority.
- Check your task for provisioning errors



Configure an Application as a Master - Troubleshooting

- Discuss the Scenario and the Steps to Fix the Issue



You should now be able to:

- Scope the business requirements
- Design the user schema
- Configure an application master

End of module review questions:

1. At which phase will you fully scope existing configurations?

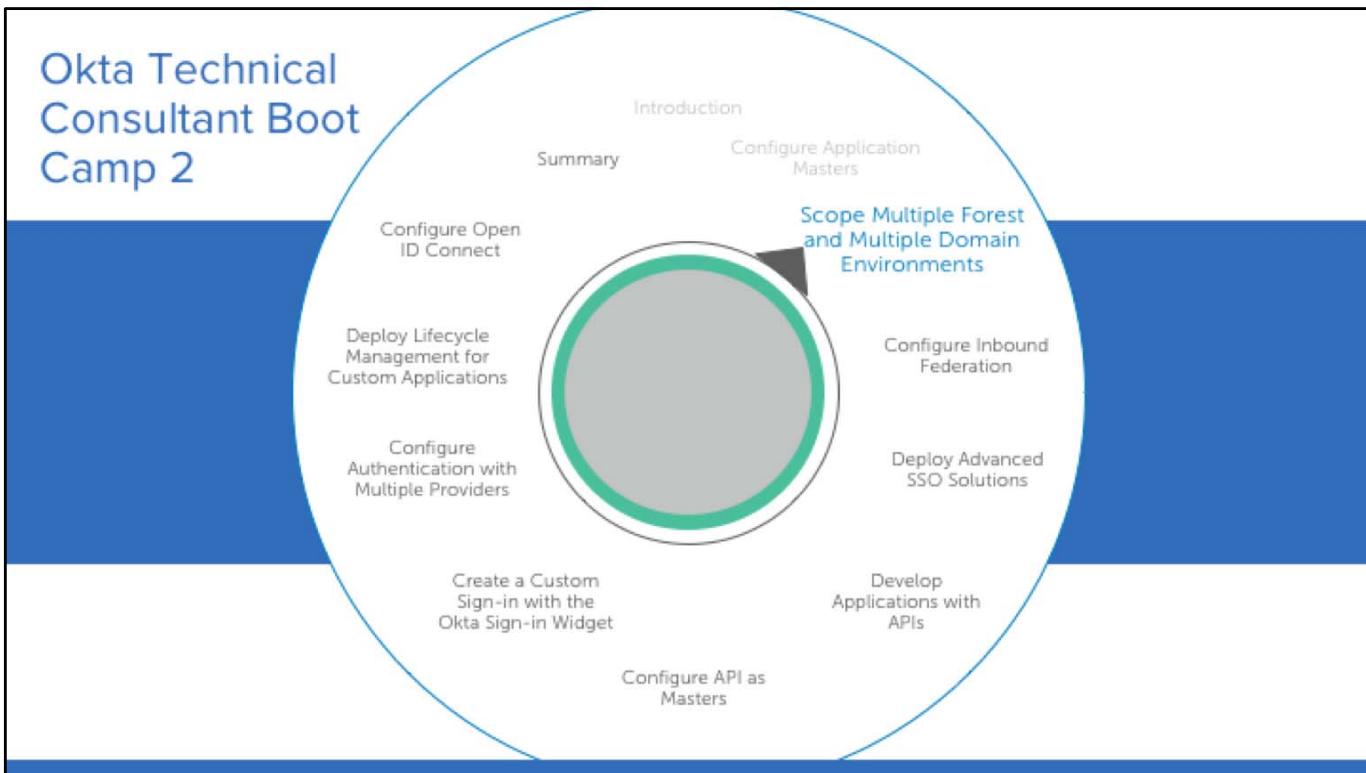
- a. Testing
- b. Overview
- c. Pre Go Live
- d. Information Gathering

Source: Page 13

2. During the test phase, which information should you be testing?

- a. Create user and application assignments
- b. User imports, user management, and application assignments
- c. Hire and terminate user, update user attributes, and attribute mappings
- d. User management, application assignments, policy enforcement, and attribute mappings

Source: Page 16



Scope Multiple Forest and Multiple Domain Environments

Okta is very commonly integrated with Active Directory as a directory service. There are many customer variations around how Active Directory is implemented.

Scope Multiple Forest and Multiple Domain Environments



- Describe Business Requirements for Different Domain Environments
- Describe Active Directory and DSSO Agents in Complex Domains
- Describe High Availability in Large Deployments

Multiple Forest and Multiple Domain Overview

In this module, you will work through common multiple forest and domain scenarios.

This module consists of a scenario and review question.

Multiple Forest and Domain Overview

Enables you to...

- Help Okta customers properly configure and implement Okta in a multi-forest and domain environment

Is important because...

- You work with different Okta customers with different user stores and requirements.
- You need help customers configure Okta based on company structures and policies.

Additional Information

While it is not possible to establish a standard around integrating Active Directory with Okta, there is a common approach and Okta method you can use.

Multiple Forests and Domains

Business Problems	Complicating Factors
<ul style="list-style-type: none"> • Mergers and acquisitions • Subsidiaries • Vendors and partnerships • Administrative delegation or autonomy • Application isolation 	<ul style="list-style-type: none"> • Disaster Recovery (DR): Active Directory and DSSO agent placement for customers with multiple data centers. • DNS Capability: DSSO URL for customers with multiple DNS domains. • High Availability (HA) and Fault Tolerance (FT): DSSO requirements when primary server is unavailable. • Network Communication and Efficiency: Putting too much importance on reducing the agent footprint.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Complicated business relationships and requirements create complicated Active Directory infrastructures. Through mergers & acquisitions, subsidiaries, vendor, and partner relationships, user stores can come from many sources. In addition to multiple user stores, the administration of the mixed environment might also consist of various administrative delegation or autonomy for the different systems. Because not all users require access to all applications and some applications might have very restrictive access controls, you might also be dealing with application isolation scenarios.

Customers with multiple:

- Data centers must make sure that Active Directory and DSSO agents can continuously communicate.
- DNS domains must route user traffic to specific hosts that might share the same DNS name.

Customers who do not use the Okta automatic failover for DSSO solution, must use Global Redirect, and design servers accordingly.

The Okta Approach

Business Problems	Complicating Factors
<ul style="list-style-type: none"> • Are addressed by looking at the Active Directory trusts. <ul style="list-style-type: none"> • Start simple: Multiple agent servers for each Active Directory domain • If trusts allow, multiple Active Directory domains can be consolidated onto a single agent server. • Consider special trust situations and cross-forest limitations. 	<ul style="list-style-type: none"> • Are addressed by looking at other elements of the customer infrastructure. <ul style="list-style-type: none"> • DR Site: Plan for agents to always be available • DNS: Plan for DNS resolution to be unique for each domain • HA and FT: Plan to meet existing customer SLA. • Network Communications: Do not over-consolidate.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Okta addresses the:

- Business problems by looking at the Active Directory trusts, while also considering special trust situations and cross-forest limitations.
- Complicating factors by looking at other elements of the customer infrastructure, such as the DR site and DNS.

Agent Deployment Matrix

Scenario	Active Directory Agent Placement	DSSO Agent Placement
Default/No Trusts	Agent servers in each domain or forest	Agent servers in each domain or forest
1 way trusts	Agent servers in trusted domains	Agent servers in trusted domain
2 way trusts	Agent servers in any domain	Agent servers in any domain
Special Case: Resource Domains	Agent servers in trusted user domains	Agent servers in resource domains
Special Case: Cross Forest	Agent servers in each forest	Agent servers in each forest
Special Case: Domain on Slow WAN link	Only put Active Directory servers here if required	Agent servers in these domains

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

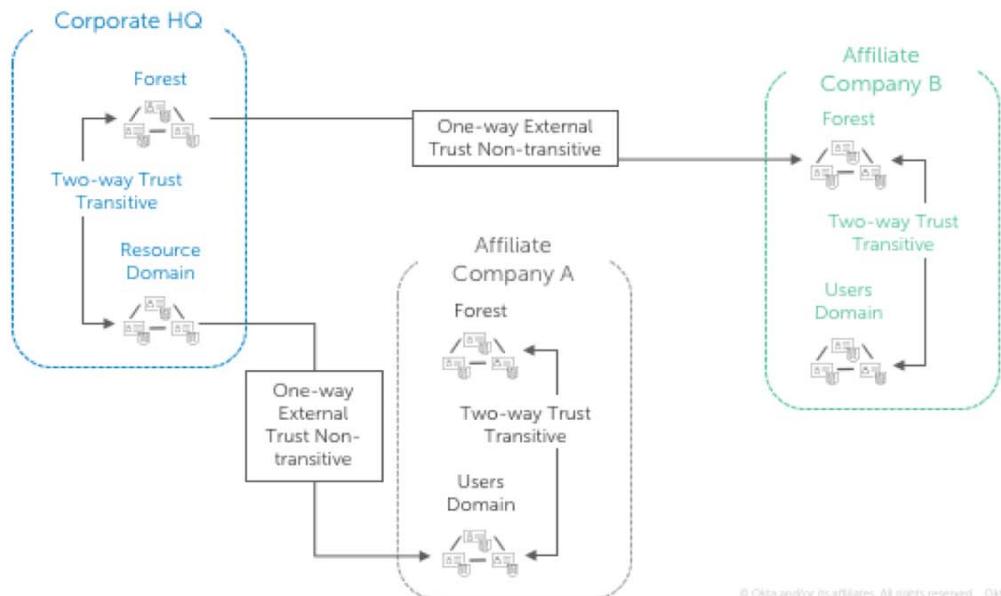
Additional Information

The table is a guide only and does not show every possible combination of trusts, domains, and forests, so there might be exceptions based on the Active Directory trusts to these guidelines.

Notes:

- At a minimum there must be one Active Directory agent for each forest, unless trusts require one for each domain. The default Active Directory trust setup should allow one agent for each forest.
- The Active Directory agent cannot communicate across forest boundary, even if supported by trust.
- A DSSO agent can be placed once for each forest.

Scenario 1: Shared Resource Domain



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

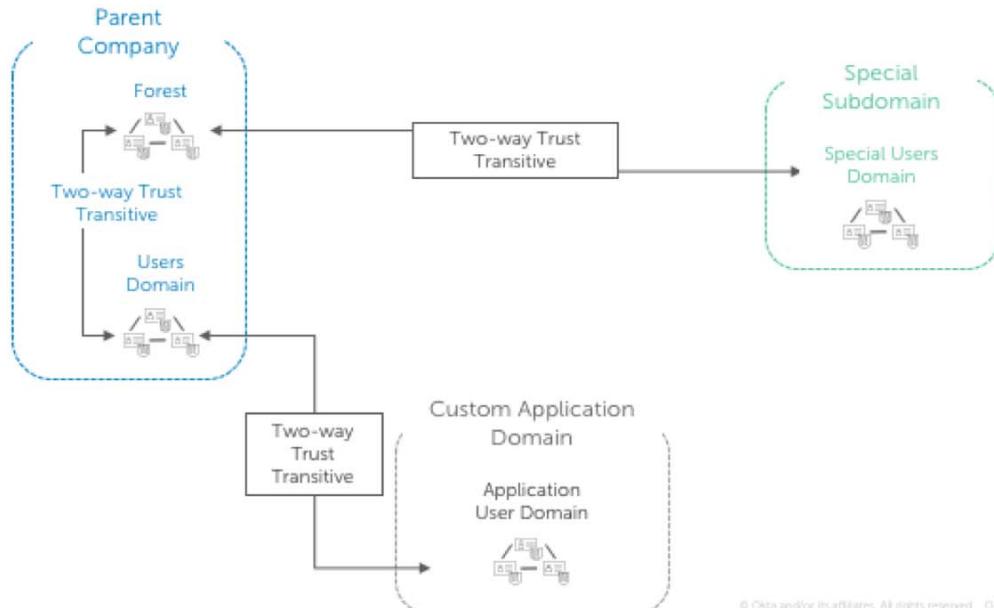
For the slide graphic:

- Active Directory specs:
 - Centralized resource domain
 - Disparate user domains
 - Mixed trust types
- Okta specs:
 - Delegated authentication
 - DSSO to a single Okta org

In this scenario:

- The Active Directory environment consists of more than 20 forests, each with one or more domains.
- There is a corporate parent company maintaining the shared resource forest and domain.
- There are also domain-level trusts between the resource domain and most subsidiary domains.
- In some cases, there are also forest-level trusts.

Scenario 2: Administrative Isolation



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

For the slide graphic:

- Active Directory specs:
 - Single forest
 - Multiple domains
 - Two-way trusts
- Okta specs:
 - Delegated authentication
 - DSSO to a single Okta org

In this scenario:

- The Active Directory environment consists of a single forest, each with one or more domains which contain users for specific purposes accessing an application.
- While this scenario shows two-way trusts use cases might vary.
- Application users are typically copies of existing users.

High Availability in Large Deployments

AD Agents	Number of users
2	First 15,000 user
1	Additional 15,000 users

AD Agents ??	Number of users
	15,000 users
	30,000 users
	90,000 users

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The table is a guide only and does not show every possible combination but should be used when deploying AD agents in a large user base. This is a best practice guideline.

Scenario: Configure a Multiple Forest Environment

Your customer wants to deploy Okta to a multiple domain environment with delegated authentication and DSSO.

Questions:

- How would you approach this scenario?
- What are the first questions to ask to determine the Okta fit?
- Describe some of the required missing data before a solution can be designed.

The Okta logo, which consists of the word "okta" in a lowercase, sans-serif font. The letters are white, and the background behind them is a blurred image of a city street at night with lights and a bridge.

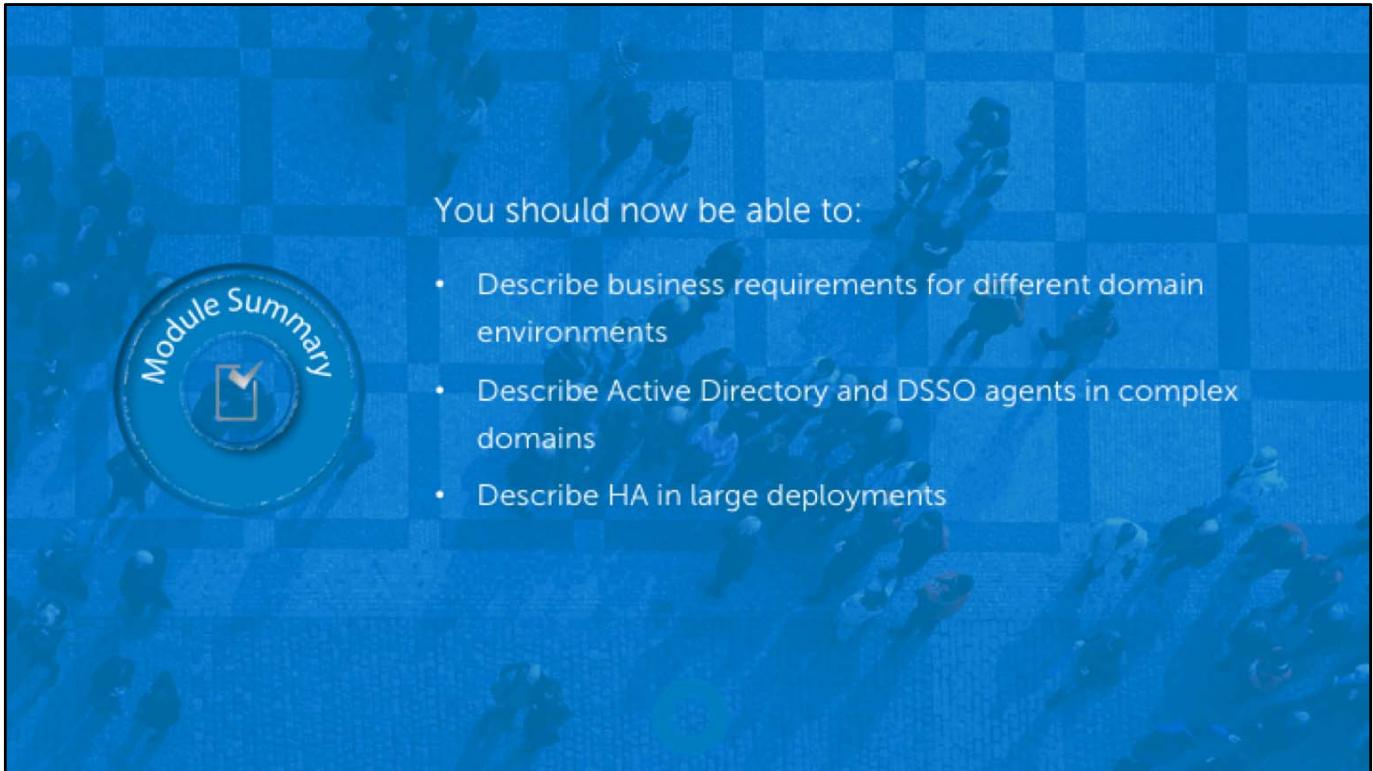
- If you are crossing forests, Active Directory agent cannot communicate. You must configure agents in each forest.
- If you need to push users to Active Directory with passwords with delegated authentication, you probably require two distinct Okta orgs and dedicated Active Directory agents for each direction.
- Use customer SLAs to determine how many domains are to be registered on a single agent server because too many domains cause excessive risk of an outage, while too few are wasteful of resources.
- Evaluate Active Directory trust usage when placing agents because crossing too many trusts for authentication might introduce excessive risk.



When troubleshooting multiple forest and or domain environments:



- Make sure you know exactly which trusts are in place between all domains and forests.
 - MS Article on Trusts: [https://technet.microsoft.com/en-us/library/cc773178\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc773178(v=ws.10).aspx)
- If you install agents at the forest level, the Active Directory service account used for the installation should also be at the forest level or issues might occur.



You should now be able to:

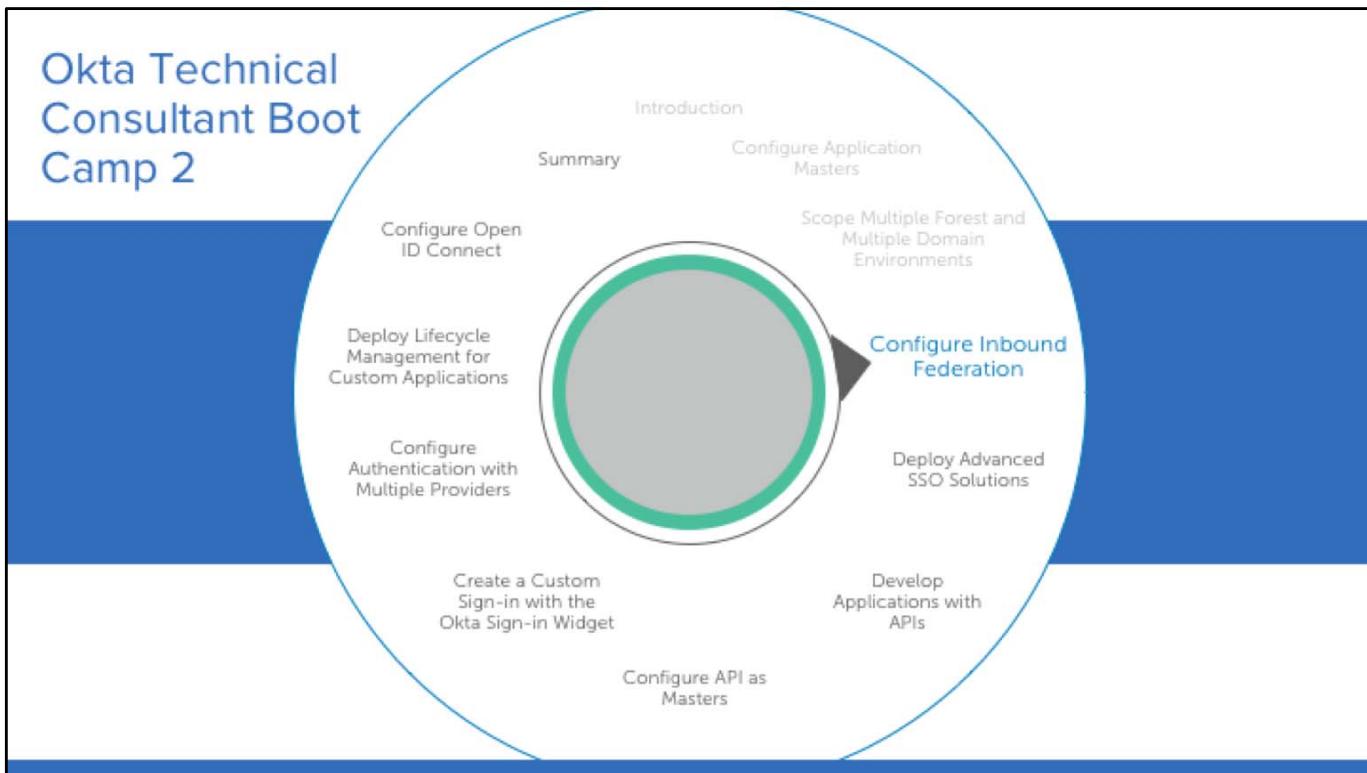
- Describe business requirements for different domain environments
- Describe Active Directory and DSSO agents in complex domains
- Describe HA in large deployments

Module Summary

End of module review question:

1. How does Okta approach a multiple forest and domain customer scenario?
 - a. By configuring a hub-and-spoke model with a parent org
 - b. By creating an Okta-only solution and phasing out Active Directory
 - c. By focusing on the Active Directory trusts and other elements of the infrastructure

Source: Page 35



Configure Inbound Federation

If a company uses more than one Okta org, perhaps because of an acquisition, creating an inbound SAML connection enables you to build federation between the Okta orgs.



Configure Inbound Federation

Scope the Business Requirements
Describe the Okta Solutions
Configure Inbound Federation

Configure Inbound Federation Overview

In this module, you will scope the business requirements to properly configure inbound federation between multiple Okta orgs. With org needs identified, you will then configure inbound federation joining Okta orgs in an IdP/SP relationship.

This module consists of two labs and review questions.

Inbound Federation Overview

Enables you to...

- Properly scope and help deploy inbound federation solutions for Okta customers

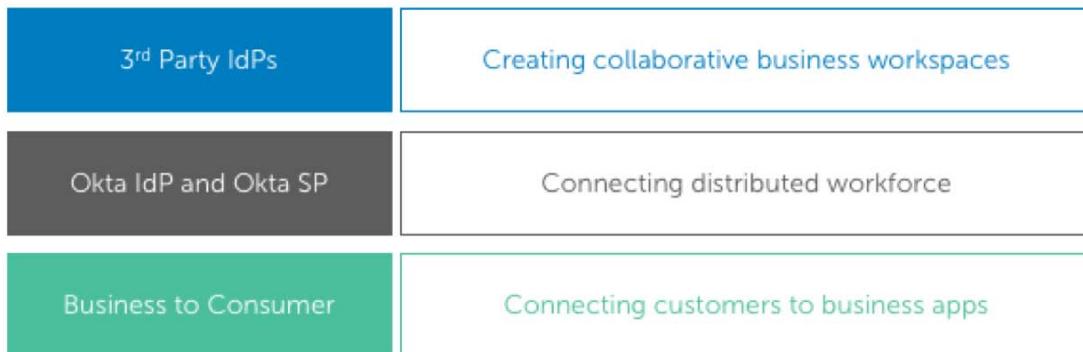
Is important because...

- You work with different Okta customers with different user stores and requirements.
- You need help customers configure Okta based on company structures and policies.

Additional Information

Many Okta customers require specific authentication between different Okta orgs and applications.

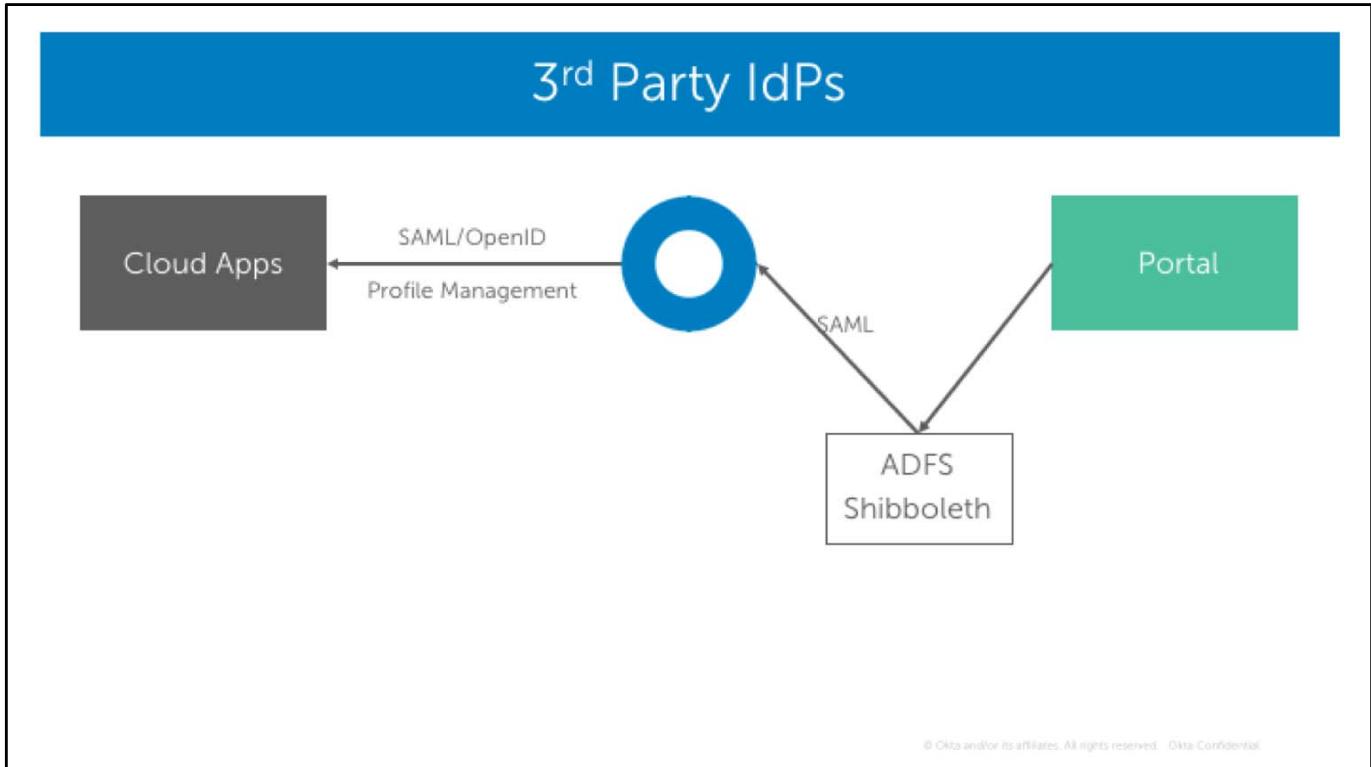
Okta Solutions: Three Distinct Scenarios



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The distinct inbound federation scenarios include configuring Okta to work with another IdP, configuring Okta as an IdP and SP, and configuring Okta in a business to consumer solution.

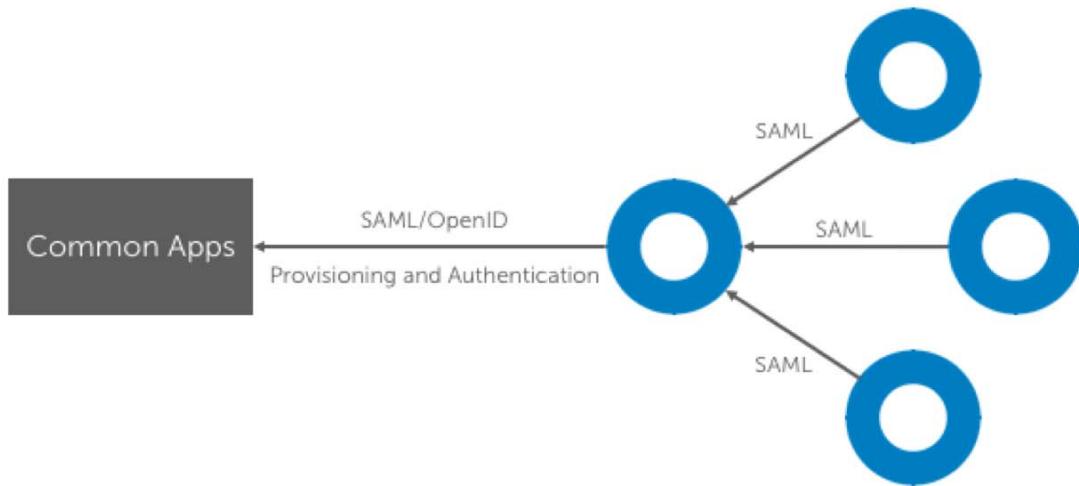


Additional Information

Use Cases:

- As an IdM administrator for my company, I want to create a federation trust with my existing IdP to make the federation server the source of record for credentials and profile attributes. Users are JIT provisioned to my Okta org during federated SSO, I need to be able to map and transform attributes from the existing federation server to Universal Directory user profiles. This mapping and transformation extends my existing IdP to cloud applications for provisioning and SSO.
- As an IdM administrator for my company, I want to JIT provision users to licensed applications such as Box on initial application access so that I do not get charged for licensed that are not required or used.
- As a IT administrator for my company's portal solution, I need to grant my partners access to an application so they can SSO through their existing IdP session.

Okta IdP and Okta SP



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In addition to using Okta as an IdP, you can also configure Okta as an SP. When Okta is used as an SP, it integrates with an IdP using SAML. Inbound SAML allows users from other IdPs to SSO into Okta.

Business to Consumer



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Use Cases: As a business owner for my company's portal or ISV application, I want to allow users to self-register with an existing social profile so that I lower the friction for new user sign-ups and keep my application user profiles in sync with data from the social platform. This use case eliminates the need for another user password by allowing users to connect the account to a social login provider for SSO.

Inbound SAML

- Okta is the SP.
- The administrator can add external SAML 2.0 providers as trusted IdPs with one trusted IdP designated as the default.
 - If this is configured, any traffic from the normal login page of the Okta org is redirected to the SSO endpoint of the IdP.
- SAML JIT is possible for creates and updates.
- The IdP is a profile master in UD and gets an application user profile with mappings.
- Groups can also be passed in the SAML assertion and used to append or overwrite group membership in Okta as part of the federated login.

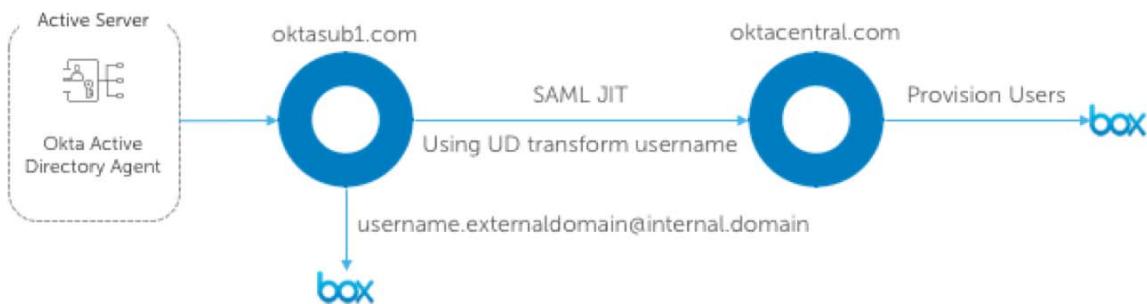
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Design Considerations

- User information complexity
- Group complexity
- Customer expertise
- Complex supported applications

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

In Depth Use Case 1

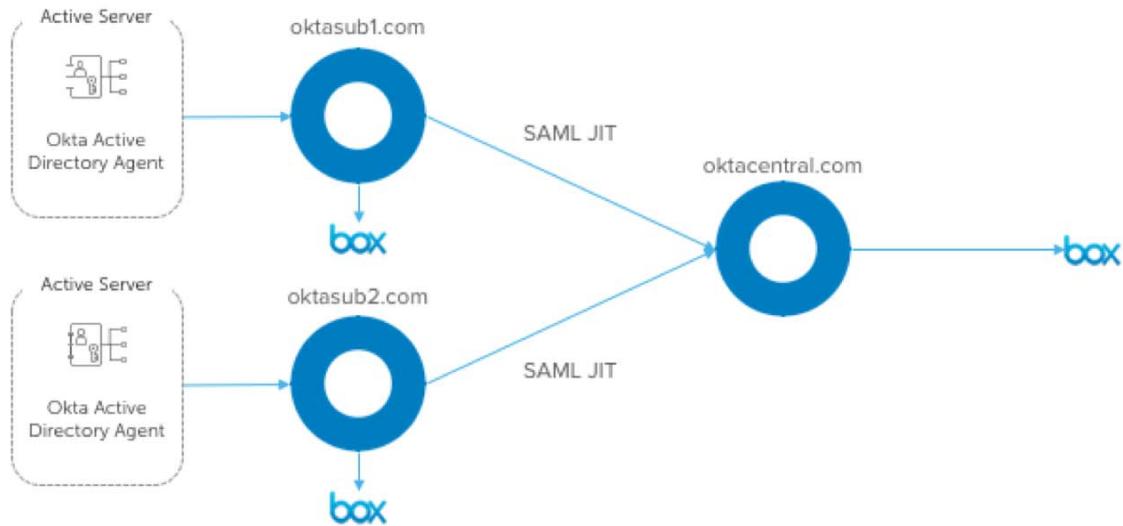


© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In this scenario, the sub org has a distinct Okta tenant which is the IdP using Okta Central as the SP to gain access to the parent Box tenant.

In Depth Use Case 2



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Configure Inbound Federation with JIT

1. Configure shared application in the Okta central org.
2. Configure Inbound SAML in the Okta central org.
 - a. Make sure JIT/Update is enabled.
 - b. Map the attributes.
 - c. (Optional) Transform data.
3. Configure an application in the Okta sub org using the AIW.
 - a. Hide the application.
 - b. Assign groups to the application.
4. Create a bookmark application in the Okta sub org pointing to the shared application in the Okta central org.

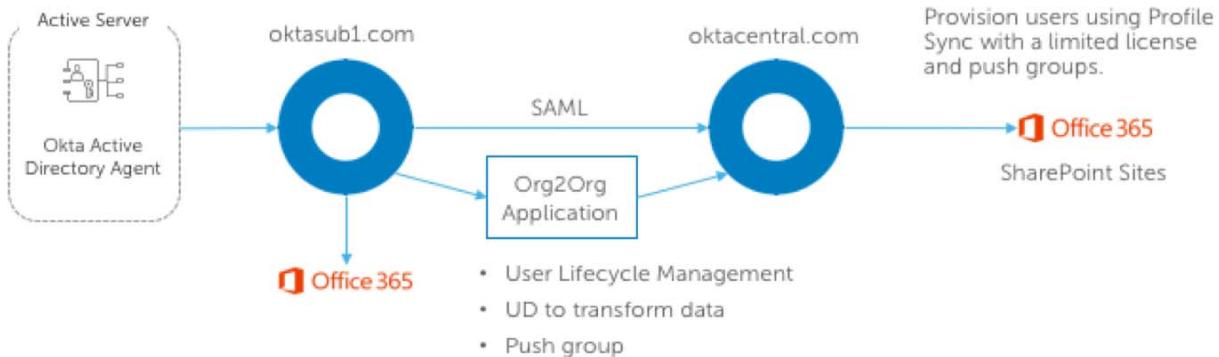
© Okta and/or its affiliates. All rights reserved. Okta Confidential.



Configure Inbound Federation With SAML JIT

- Configure the AIW Application
- Configure Inbound SAML
- Configure the Bookmark Application
- Verify Results

In Depth Use Case: Complex Application



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In this scenario, each sub org has a distinct Okta tenant which is the IdP using Okta Central as the SP to gain access to the parent SharePoint site.

Configure Inbound Federation with Org2Org

1. Configure the shared application in the Okta central org.
2. Configure Inbound SAML in the Okta central org and verify JIT is off.
3. Configure an application in Okta sub org using the AIW.
 - a. Hide the application.
 - b. Assign groups to the application.
4. Configure the Org2Org application.
 - a. Map the attributes.
 - b. (Optional) Transform the data and push groups.
 - c. Assign any groups.
5. Create a bookmark application in the Okta sub org pointing to the shared application in the Okta central org.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.



- **Configure Inbound Federation with Org2Org**
 - Create the API Token
 - Configure the Org2Org Application
 - Transform Data
 - Configure Groups and Group Rules
 - Verify Results

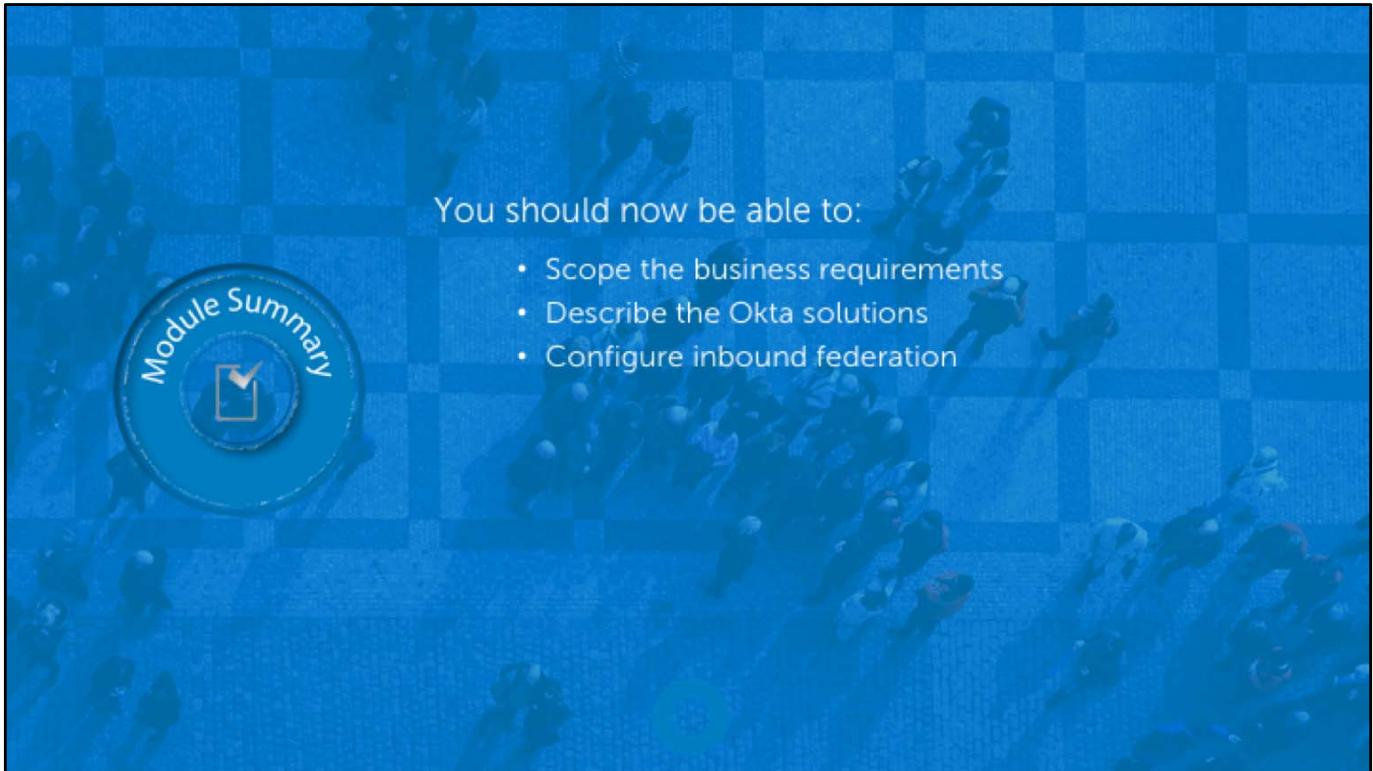


- When deploying O365 in the IdP, if the IdP shares a consolidated email domain in the central org, this might cause issues with non-ADAL client-like mobile applications that cannot authenticate against the IdP. You will have to synchronize the password which is probably not the wanted end user experience.
- G Suite can also have native mail client issues where it cannot authenticate and you have to synchronize the password to the central org.
- Org2Org - Assign using a group set the Initial status to active_with_pass
- Add an attribute to store the users home org in the Hub User Profile and map a static string in org2org, in case IDP Discovery is needed





- Verify data mappings and transformations are correct.
- Verify all required attributes being passed on the SAML assertion.
- Verify if using Org2Org provisioning that the API token has not expired.
- Check the Tasks page for provisioning errors.
- Use a SAML tracer tool to view the SAML assertion.



You should now be able to:

- Scope the business requirements
- Describe the Okta solutions
- Configure inbound federation

End of module review questions:

1. How can you provision users when using inbound federation?

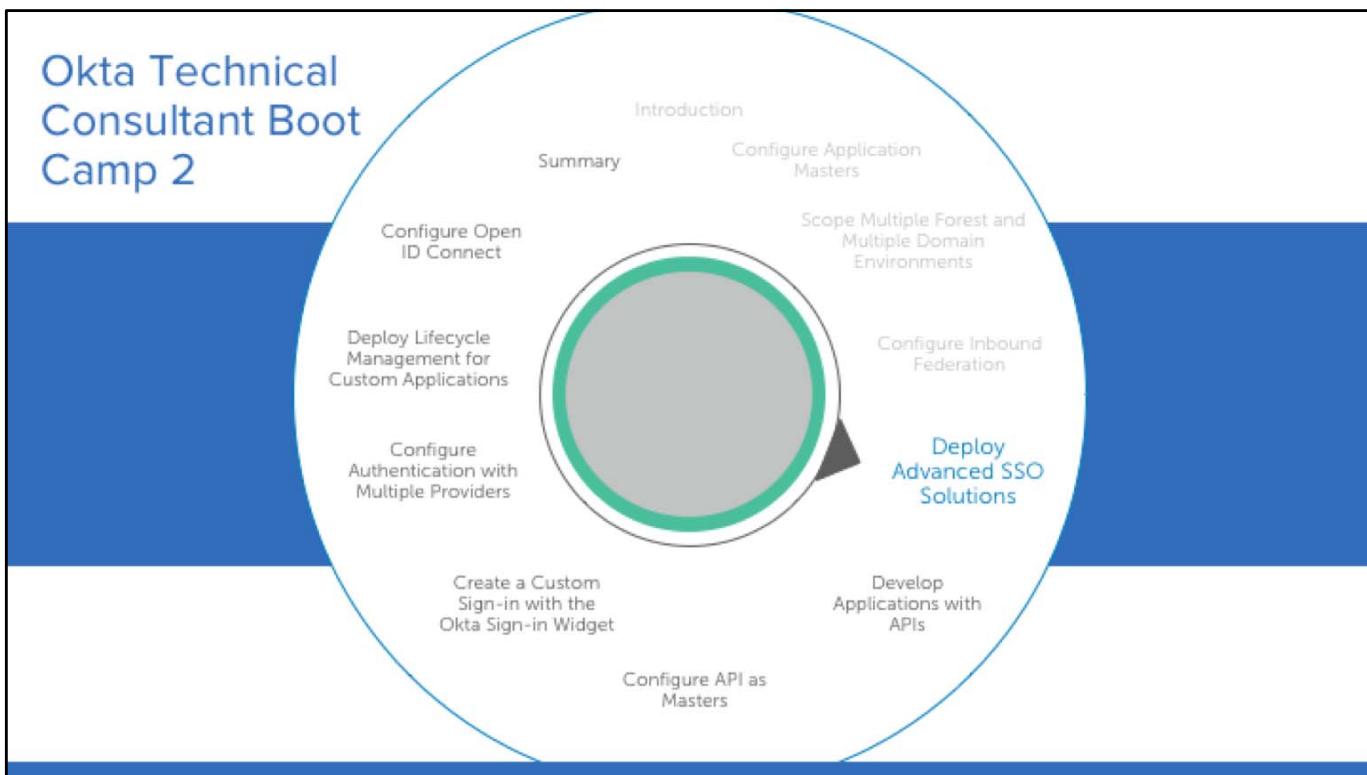
- a. SAML JIT
- b. SAML Encryption
- c. SAML Key Rollover
- d. SAML Single Sign Out

Source: Page 52

2. Which federated inbound protocols does Okta support?

- a. SWA
- b. SAML
- c. OAuth
- d. WS-Fed

Source: Page 52



Deploy Advanced SSO Solutions

If a company needs to use more advanced SAML configuration options, perhaps because of a business requirement, you need to know how to configure those options within Okta. The Okta RADIUS agent is another authentication option supported for a large scale enterprise deployment.

Deploy Advanced SSO Solutions



Describe Reverse Proxy Flows for Header-based Authentication
Configure Advanced SAML Configurations
Configure Radius Agent Configurations

Deploy Advanced SSO Solutions Overview

If a company would like to use more advanced configuration options with SAML, understanding how to implement these features are very important. Also, in large enterprise configurations there could several authentication options, such as RADIUS.

This module consists of labs and review questions.

Advanced SSO Solutions Overview

Enables you to...

- Assist Okta customers with varying and complex SSO solutions

Is important because...

- Not every customer has a common approach to SSO.
- You must help customers configure Okta based on company structures and policies.

Additional Information

Okta provides SSO for more than only public applications - you can also configure SSO with various 3rd-party solutions, such as Shibboleth.

Header-based Authentication

Why Needed

- Legacy applications that do not natively support SAML, OIDC, or OAuth
- Applications that rely on header variables passed for user identity
- Okta does not pass header variables directly
- Other custom SSO integrations that are not options with Okta such as Kerberos
- Translating layer which accepts SAML from Okta and converts the identity to a form that the downstream application understands

3rd Party Solutions

- F5 BIG-IP
- Shibboleth
- Apache using Mod_Auth_Mellon
- ICS

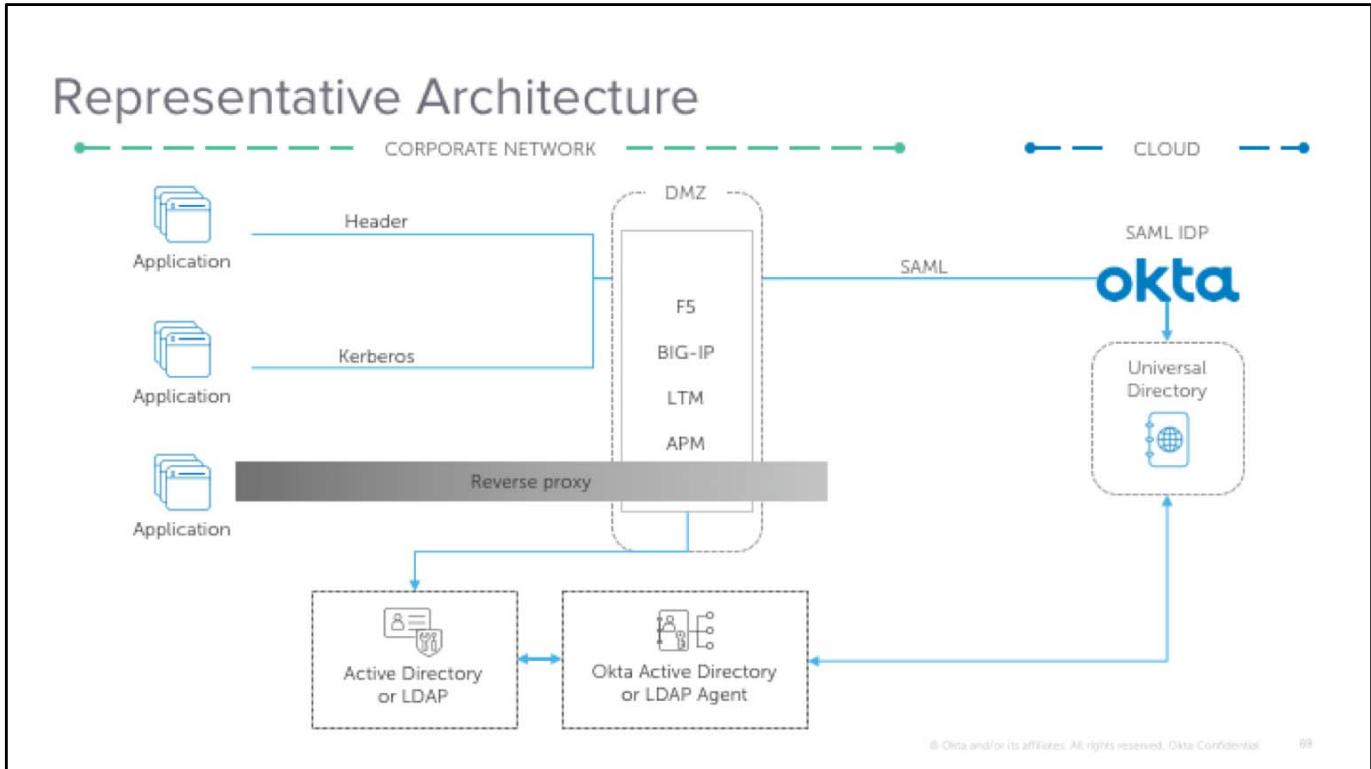
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

68

Additional Information

SAML is the recommended application configuration, but not all applications support SAML. As a result, header-based authentication is required. Header-based authentication supports custom integrations such as:

- MD5 digest-based authentication
- Parameters
- Cookies
- Server-variables
- Kerberos tokens



Additional Information

There are four scenarios for the basic integration that are supported:

1. Okta is the IdP. Users can be defined locally with Okta. In most cases, an on-premise Active Directory or LDAP is the source of the identities and is integrated with Okta through an Okta directory agent.
2. Between Okta and F5 BIG-IP, a SAML trust is built where F5 BIG-IP acts as a SAML SP.
3. The target applications are protected behind F5 BIG-IP. These applications are protected by header-based authentication or Kerberos.
4. SAML assertion from Okta is consumed by F5 BIG-IP which then translates the assertion appropriately for the downstream application based on the authentication scheme.

Okta and F5 BIG-IP

Authentication Mechanism	Okta	F5 BIG-IP
Header-based	Acts as IdP	Receives SAML from Okta – generates header(s) for downstream application
Kerberos	Acts as IdP	Receives SAML from Okta – obtains Kerberos ticket for downstream Kerberos-enabled application
Reverse-proxy to access on-premise application from outside the firewall	Acts as IdP if only authenticated users are allowed	Acts as reverse proxy

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

70

Okta IdP and F5 BIG-IP SP

1. Publish Sample ASP .NET IIS Web Application Through F5 BIG-IP

2. Configure Okta as SAML 2.0 IdP for F5 BIG-IP

3. Configure F5 BIG-IP as SAML 2.0 SP for Okta

4. Test the SSO Integration

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

71

Additional Information

These are the implementation steps for a F5 BIG-IP. These steps could be more generic:

1. A .Net web application that uses header-based authentication.
2. Configure Okta as SAML 2.0 IdP for the reverse proxy.
3. Configure the reverse proxy as SAML 2.0 SP for Okta.
4. Test the SSO Integrations.

Sample .Net Web Application

The screenshot illustrates the configuration of a sample .Net web application on an F5 BIG-IP system. On the left, the 'iApp Services' interface shows the creation of a new iApp named 'SSONetApp' based on the 'f5.microsoft_iis' template. The template selection screen includes sections for 'SSL Encryption Questions' (asking if the BIG-IP system should offload SSL processing) and 'Virtual Server Questions' (specifying the virtual server IP as 12.12.1.12, port 80, and route traffic to application clients via the BIG-IP system). On the right, a detailed configuration dialog for 'Server Pool, Load Balancing, and Service Monitor Questions' is displayed, with the IP address '11.11.1.11' and port '80' highlighted in red.

Additional Information

Creating the sample .Net web application in the F5 BIG-IP would involve:

1. Creating an iApp.
2. Using the f5.microsoft_iis template.
3. Providing the virtual server IP address.
4. Providing the IP address of the test web server.
5. Providing the listening port.
6. Providing an FQDN for the web server hostname.

Sample .Net Web Application Continued



Additional Information

Next, test your sample .Net web application by performing the following:

1. Open a browser on the host machine.
2. Point to the external IP address.
3. Render the backend webserver page.

Sample .Net Web Application Continued

www.democorp.co/headers.aspx	
HTTPS	off
HTTPS__KEYSIZE	
HTTPS__SECRETKEYSIZE	
HTTPS__SERVER_ISSUER	
HTTPS__SERVER_SUBJECT	
INSTANCE_ID	1
INSTANCE_META_PATH	/LM/W3SVC/1
LOCAL_ADDR	11.11.1.11
PATH_INFO	/headers.aspx
PATH_TRANSLATED	C:\inetpub\wwwroot\headers.aspx
QUERY_STRING	
REMOTE_ADDR	11.11.1.2
REMOTE_HOST	11.11.1.2
REMOTE_PORT	62644
REQUEST_METHOD	GET
SCRIPT_NAME	/headers.aspx
SERVER_NAME	www.democorp.co
SERVER_PORT	80
SERVER_PORT_SECURE	0
SERVER_PROTOCOL	HTTP/1.0
SERVER_SOFTWARE	Microsoft-IIS/8.0
URL	/headers.aspx
HTTP_CONNECTION	keep-alive
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.8
HTTP_COOKIE	BIGipServerSSOWebApp.app~SSOWebApp_pool=184617739.20480.0000
HTTP_HOST	www.democorp.co
HTTP_USER_AGENT	Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
HTTP_UPGRADE_INSECURE_REQUESTS	1
HTTP_DNT	1

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

74

Additional Information

Next, configure two files:

1. Place a hosts file entry to point to a test Homepage (e.g. www.democorp.co) to point to the backend application IP address.
2. Place a headers.aspx in the webserver root folder with the following line to display all headers:

```
<%@Page!Language="C#"!Trace="true"%>
```

Configure Okta as SAML 2.0 IdP for F5 BIG-IP

Create a New Application Integration

Platform: Web

Sign on method:

- Secure Web Authentication (SWA)
- SAML 2.0 (selected)
- OpenID Connect

Create Cancel

SAML Settings

GENERAL

Single sign on URL: <https://www.democorp.com/saml/sp/profile/post/acs> (highlighted with red box)

Audience URI (SP Entity ID): <https://www.democorp.com/sp> (highlighted with red box)

Default RelayState: (empty)

Name ID format: Unspecified

Application username: Okta username

Response: Signed

Assertion Signature: Signed

Signature Algorithm: RSA-SHA256 (highlighted with red box)

Digest Algorithm: SHA256

Assertion Encryption: Unencrypted

Enable Single Logout: (unchecked)

Hide Advanced Settings

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

75

Additional Information

Next you must create an SAML AIW application in Okta:

1. Create a SAML application in Okta using the AIW.
2. Provide the following Single Sign On URL:
<https://external-f5-hostname/saml/sp/profile/acs>
3. Provide the following Audience URI:
<https://external-f5-hostname/sp>
4. Verify the Signature Algorithm is set to:
RSA-SHA256

Configure Okta as SAML 2.0 IdP for F5 BIG-IP Continued

The screenshot shows the Okta configuration interface for setting up SAML 2.0 attributes. It includes two main sections:

- ATTRIBUTE STATEMENTS (OPTIONAL):** This section lists four attributes: FirstName, LastName, EmailAddress, and City. Each attribute has its name, name format (Unspecified), and value defined. For example, FirstName is mapped to user.firstName.
- GROUP ATTRIBUTE STATEMENTS (OPTIONAL):** This section allows defining a group attribute with a name, name format (Unspecified), and a filter (e.g., Starts with).

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

76

Additional Information

Next, configure the attribute statement to pass the correct custom attributes in the SAML assertion to the ASP .NET application. In the slide example, the first name, last name, email, and city are being passed.

Configure Okta as SAML 2.0 IdP for F5 BIG-IP Continued

The screenshot shows the Okta Assignments page for the 'F5_BIG_IP' application. The top navigation bar includes a gear icon, the application name 'F5_BIG_IP', a status dropdown set to 'Active', and a 'View Logs' button. Below the navigation are tabs: General, Sign On, Mobile, Import, and Assignments, with 'Assignments' being the active tab. A sub-header 'Assign' with a dropdown arrow is visible. The main area displays a table titled 'Assignment'. The left column is labeled 'FILTERS' with options 'People' and 'Groups' (which is selected and highlighted in blue). The middle column is 'Priority' (showing '1') and the right column is 'Assignment'. One assignment row is present, labeled 'Employees' with a note 'No Description'. To the right of the table are edit and delete icons. At the bottom of the table are search and filter buttons labeled 'Search...' and 'People'.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

77

Additional Information

Next, in Okta assign the authorized users or groups. In this example, the Employees group is being assigned to the SAML F5_BIG_IP application.

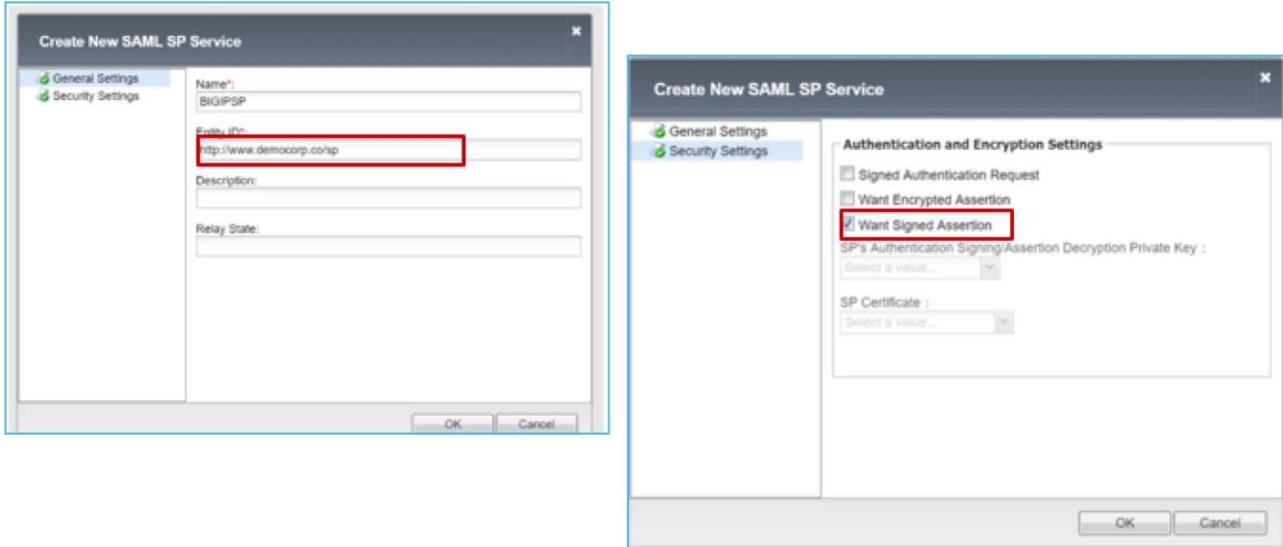
Configure Okta as SAML 2.0 IdP for F5 BIG-IP

The screenshot shows the Okta Sign On Settings page. The 'Sign On' tab is selected. Under the 'SIGN ON METHODS' section, 'SAML 2.0' is listed. A yellow callout box highlights the 'Identity Provider metadata' link, which is enclosed in a red rectangle. Below the 'CREDENTIALS DETAILS' section, there is a note: '© Okta and/or its affiliates. All rights reserved. Okta Confidential. 78'

Additional Information

The final required item is the Identity Provider Metadata (metadata.xml) file, which you must create by clicking the link.

Configure F5 BIG-IP as SAML 2.0 SP for Okta



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

79

Additional Information

Next, configure F5 BIG-IP as SAML 2.0 SP for Okta by creating a SAML SP service in your F5 BIG-IP Access Policy Manager:

1. Provide a name for the endpoint. Provide the SP Entity id:
<https://external-f5-hostname/sp>
2. Verify the Want Signed Assertion is selected.

Configure F5 BIG-IP as SAML 2.0 SP for Okta Continued

The left screenshot shows the F5 BIG-IP Local IP Services interface with the 'SAML IdP Connectors' tab selected. It lists a single connector named 'BIGIPSP'. The right screenshot shows the 'Edit SAML IDP's that use this SP' dialog box, which contains a table for managing SAML IdP Connectors. A dropdown menu on the right side of the dialog box is open, showing options: 'Custom', 'From Metadata', and 'From Template'.

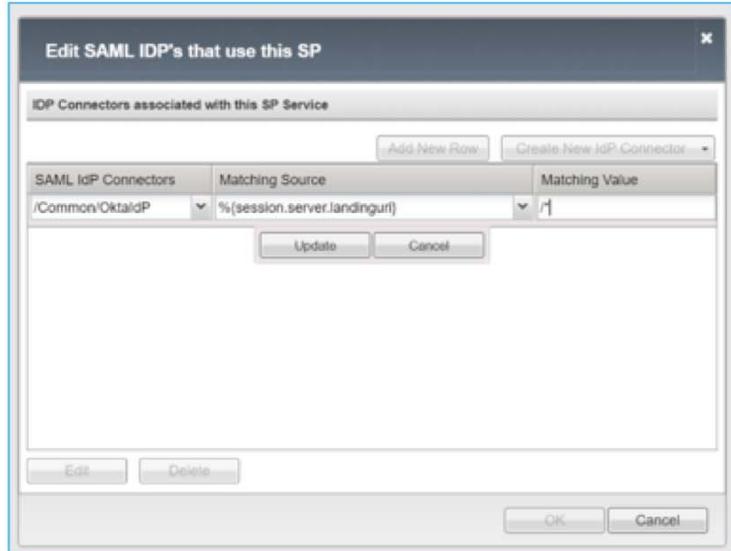
80

Additional Information

Next, upload the Identity Provider metadata file to the F5 BIG-IP:

1. On the **Main** tab, select your endpoint.
2. Select **Bind/Unbind IdP Connectors**.
3. Click **Create New IdP Connector > From Metadata**.
4. Browse to and select the **metadata.xml** file.

Configure F5 BIG-IP as SAML 2.0 SP for Okta Continued



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

81

Additional Information

This creates an Okta IdP connector and imports the signing certificate.

1. Click **Add New Row**.
2. Select **OktaldP** as the **SAML IdP Connect**.
3. Select the following **Matching Source**:
`%{session.server.landingurl}`
4. Select the following **Matching Value**:
`/*`

Configure F5 BIG-IP as SAML 2.0 SP for Okta Continued

The left screenshot shows the Okta Access Policy Editor interface. An access policy named 'OktaSAMLAuth' is being created. The flowchart starts with 'Start' (default), followed by 'Okta SAML Auth' (default), then 'Successful' (green box), and finally 'Logout' (red box). There is also a 'Failure' (red box) path. A note at the bottom explains the components of an access policy: a start point, actions, and one or more endings. It provides instructions for inserting actions, editing endings, deleting actions, and adding macros. It also mentions the 'Add Macro' button for pre-defined access policy items.

The right screenshot shows the F5 BIG-IP Local Traffic interface. Under 'Virtual Servers', a list of virtual servers is shown, including 'SSOWebApp_Http_Virtual'. The interface includes tabs for Main, Help, and About, and sections for Statistics, iApp, Wizards, and Local Traffic (Network Map, Virtual Servers, Profiles, iRules).

Additional Information

Next, create an access policy:

1. Create an Access Policy in F5 BIG-IP.
2. Change the default policy to SAML Auth.
3. Attach the Access Policy to the Virtual Server.

Configure F5 BIG-IP as SAML 2.0 SP for Okta Continued

```
when RULE_INIT {
set static::debug 0
}
when ACCESS_ACL_ALLOWED {
    set oktaUser [ACCESS::session data get "session.saml.last.identity"]
    if { $static::debug } { log local0. "id is $oktaUser" }
    if { !{[HTTP::header exists "OKTA_USER"]} } {
        HTTP::header insert "OKTA_USER" $oktaUser }

    set oktaFirstName [ACCESS::session data get "session.saml.last.attr.name.FirstName"]
    if { $static::debug } { log local0. "id is $oktaFirstName" }
    if { !{[HTTP::header exists "OKTA_FIRSTNAME"]} } {
        HTTP::header insert "OKTA_FIRSTNAME" $oktaFirstName }

    set oktaLastName [ACCESS::session data get "session.saml.last.attr.name.LastName"]
    if { $static::debug } { log local0. "id is $oktaLastName" }
    if { !{[HTTP::header exists "OKTA_LASTNAME"]} } {
        HTTP::header insert "OKTA_LASTNAME" $oktaLastName }

    set oktaCity [ACCESS::session data get "session.saml.last.attr.name.City"]
    if { $static::debug } { log local0. "id is $oktaCity" }
    if { !{[HTTP::header exists "OKTA_CITY"]} } {
        HTTP::header insert "OKTA_CITY" $oktaCity}
}
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

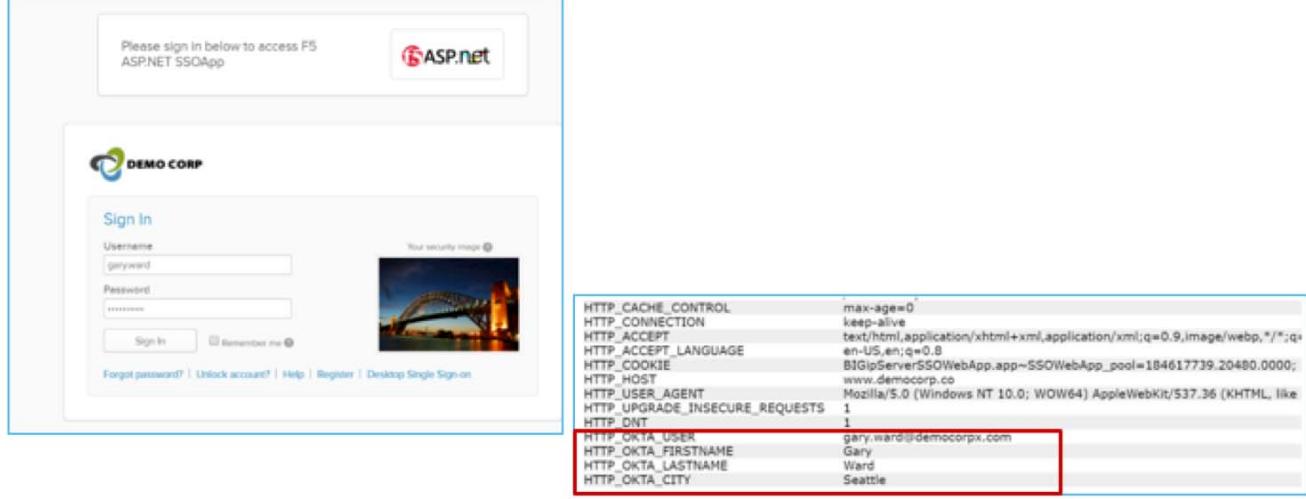
83

Additional Information

Next, create an F5 BIG-IP iRule to extract the custom SAML attributes from the incoming assertion and pass them as HTTP headers to the backed ASP .NET application.

Apply this F5 BIG-IP iRule to the virtual server.

Testing the SSO Integration



The screenshot shows the Okta sign-in interface and the resulting F5 BIG-IP log output.

Okta Sign-In Page:

- Header: "Please sign in below to access F5 ASP.NET SSOApp".
- Logo: ASP.NET.
- Form fields: Username (garyward), Password, Sign In button, Remember me checkbox, and links for Forgot password?, Unlock account!, Help, Register, and Desktop Single Sign-on.
- Background image: A bridge at night.

F5 BIG-IP Log Output:

HTTP_CACHE_CONTROL	max-age=0
HTTP_CONNECTION	keep-alive
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.8
HTTP_COOKIE	BIGIPServerSSOWebApp.app=SSOWebApp_pool=184617739.20480.0000; www.democorp.co
HTTP_HOST	Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like
HTTP_USER_AGENT	1
HTTP_UPGRADE_INSECURE_REQUESTS	1
HTTP_DNT	1
HTTP_OKTA_USER	gary.ward@democorp.com
HTTP_OKTA_FIRSTNAME	Gary
HTTP_OKTA_LASTNAME	Ward
HTTP_OKTA_CITY	Seattle

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 84

Additional Information

Navigate to the published application URL in our example it is:
<https://www.democorp.co/headers.aspx>

F5 BIG-IP should redirect the request to Okta for authentication. After Okta authenticates the user, it should be redirected to the published application web page. Note the **HTTP_OKTA_*** headers indicating successful extraction of SAML headers.

Advanced SAML Configurations

Assertion Encryption

SAML Single Logout (SLO)

Force Authentication

Key Rollover

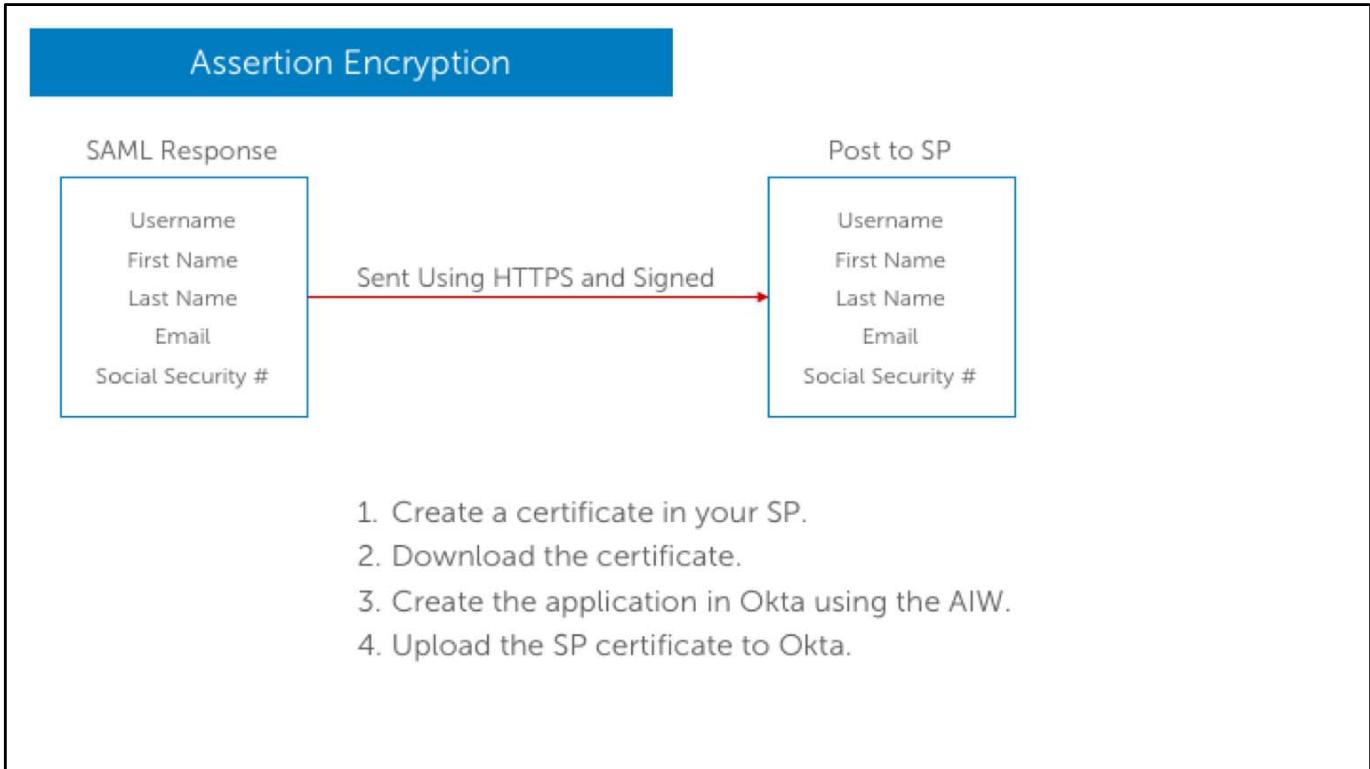
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

85

Additional Information

Advanced SAML configuration options can be used to change settings for advanced, alternative deployment scenarios or to test different SSO configurations.

- Assertion encryption could be used to protect the privacy of the data after it arrives at the SP, if the SP supports encryption.
- The SAML protocol is popular for enabling SSO and contains a built-in feature known as SLO. This additional protocol helps address the issue of orphaned logins. SLO allows a user to terminate all server sessions established through SAML SSO by initiating the logout process once. SLO is initiated from the IdP or any of the involved SP.
- For SAML 2.0 IdPs that support it, you can pass ForceAuthn="true" as an attribute for the AuthnRequest. This tells the IdP to not use any previous security context when authenticating the user.
- The term key rollover refers to a process when one key is systematically replaced by another key in SAML metadata. Because entities (and therefore SAML metadata) are distributed, key rollover must be deliberate, so as not to break the key operations of a relying party.



Additional Information

If the SAML response contains claims and assertions that contain private data, and the receiver of the response will be holding onto the SAML assertion for an indefinite period or passing the SAML assertions through intermediate parties you do not trust, then yes, the SAML assertions should be encrypted and the response signed, regardless of whether it is transmitted by SSL or not. Encryption is to protect the privacy of the data after it arrives at the Service Provider.

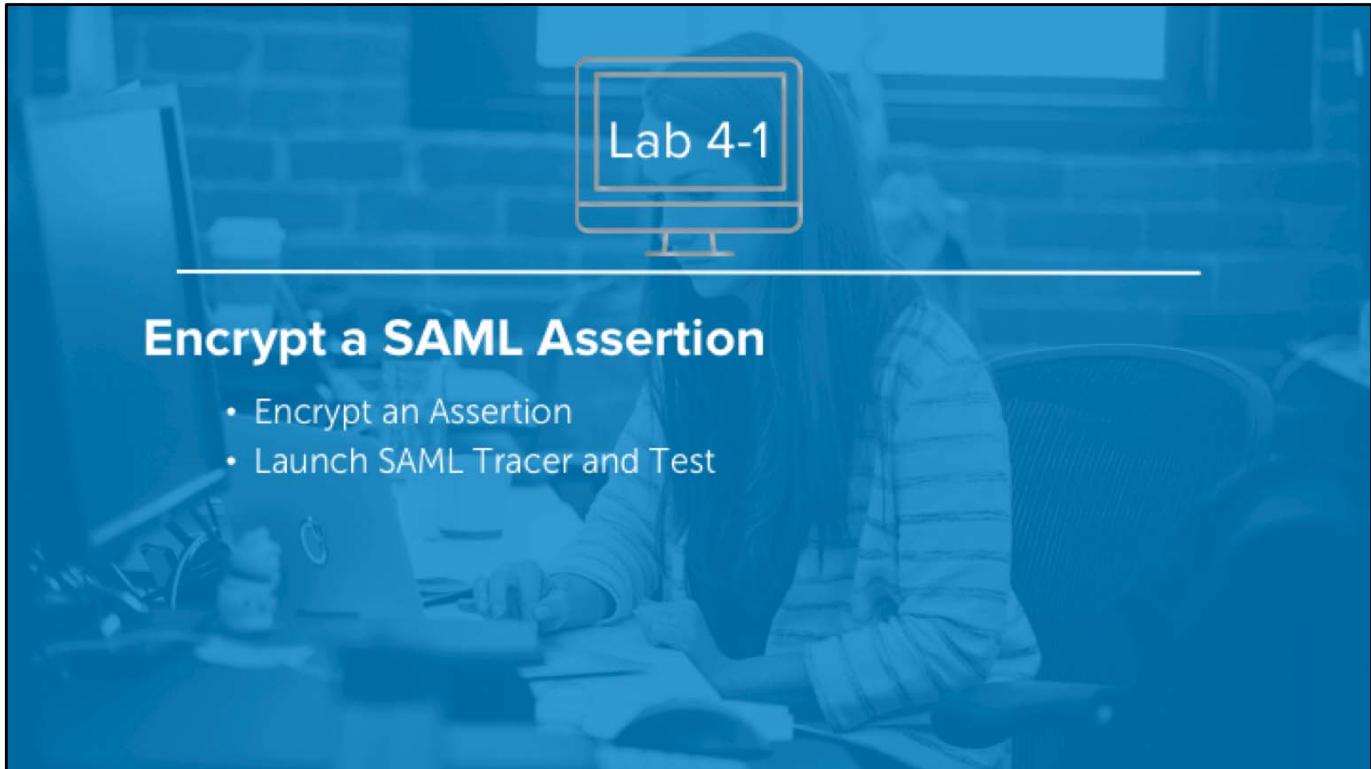
The screenshot displays two Okta interfaces side-by-side. On the left is the 'Assertion Encryption' section of the 'SAML Settings' page, which includes fields for General, Response, Assertion Signature, Signature Algorithm, Digest Algorithm, and Assertion Encryption. The 'Assertion Encryption' section is highlighted with a red border. On the right is the 'SAML tracer' interface, showing a POST request to 'https://www.example.com/' with the 'SAML' tab selected. The response body contains a large, complex XML document representing an encrypted SAML assertion.

Additional Information

Assertion contents:

- Are configurable through the Okta AIW
- Are sent as cipher text within the SAMLResponse
- Are encrypted using the SP public key
- Helps ensure confidentiality
- Can only be decrypted using the private key by the intended recipient

Troubleshooting is more difficult because the assertion contents are not decipherable through browser plugins such as SAML Tracer.



The background of the slide features a photograph of a person sitting at a desk in an office environment, facing a computer monitor. The monitor displays the text "Lab 4-1". The person is wearing a light-colored shirt and appears to be looking at the screen. The overall color tone of the background is blue.

Encrypt a SAML Assertion

- Encrypt an Assertion
- Launch SAML Tracer and Test

Single Logout (SLO)

- Terminates session for all federated application and the IdP using SAML protocol messages.
- Is configured through the Okta AIW

The screenshot shows the 'Single Logout (SLO)' configuration page in the Okta AIW. It includes fields for Response (Signed), Assertion Signature (Signed), Signature Algorithm (RSA-SHA256), Digest Algorithm (SHA256), Assertion Encryption (Encrypted), Encryption Algorithm (AES256-CBC), Key Transport Algorithm (RSA-GAEP), and Encryption Certificate (Browse files...). A red box highlights the 'Enable Single Logout' section, which contains a checked checkbox for 'Allow application to initiate Single Logout', a 'Single Logout URL' field, an 'SP Issuer' field, and a 'Signature Certificate' field with a 'Upload Certificate' button. Below this are fields for 'Authentication context class' (PasswordProtectedTransport), 'Honor Force Authentication' (Yes), and 'SAML Issuer ID' (http://www.okta.com/\$org.externalKey).

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

89

Additional Information

The Okta SAML implementation provides limited support for applications implementing SLO.

The SAML application created in Okta can be configured to accept SAML LogoutRequest messages from the SP and after terminating the Okta session; sending back a SAML LogoutResponse to the SP SLO endpoint.

After configuring the SLO on an Okta application, the Sign On tab instructions provide the IdP SLO endpoint which is also included in the IdP metadata.

Okta does not track all applications federated during a session, so all application sessions, other than the application initiating the SLO, remain valid.

Force Authentication

- Is typically a business logic requirement in SP applications
- Is a method in SAML for an SP to force a re-authentication when user has a valid session
- Is initiated by the SP through as an attribute for the SAML AuthnRequest: Set ForceAuthn="true"
- Forces the IdP to respond by re-authenticating the user

The screenshot shows the Okta configuration interface for a SAML application. The 'Force Authentication' setting is highlighted with a red border. Other settings shown include Response (Signed), Assertion Signature (Signed), Signature Algorithm (RSA-SHA256), Digest Algorithm (SHA256), Assertion Encryption (Encrypted), Encryption Algorithm (AES256-CBC), Key Transport Algorithm (RSA-GAEP), and Encryption Certificate (Browse files...). The 'Honor Force Authentication' field is set to 'Yes'. The 'SAML Issuer ID' field contains the URL [http://www.okta.com/\\$org.externalKey](http://www.okta.com/$org.externalKey).

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

90

Additional Information

For the Okta configuration, Honor Force Authentication must be set to yes.

When activated, even with Desktop Single Sign-on (DSSO) configured for valid DSSO users, there is a form authentication prompt.

Key Rollover	
Information	Steps to Complete
<ul style="list-style-type: none">• Is only currently available through the API• Is useful when SP has validation limitations for SAML response and assertions signed by a certificate with 30 or 10 year validity	<ol style="list-style-type: none">1. Generate the new credential for the source application.2. Update the source application to use the new certificate.3. Share the source application key credential with the target application.4. Update the target application to use the newly shared credential.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

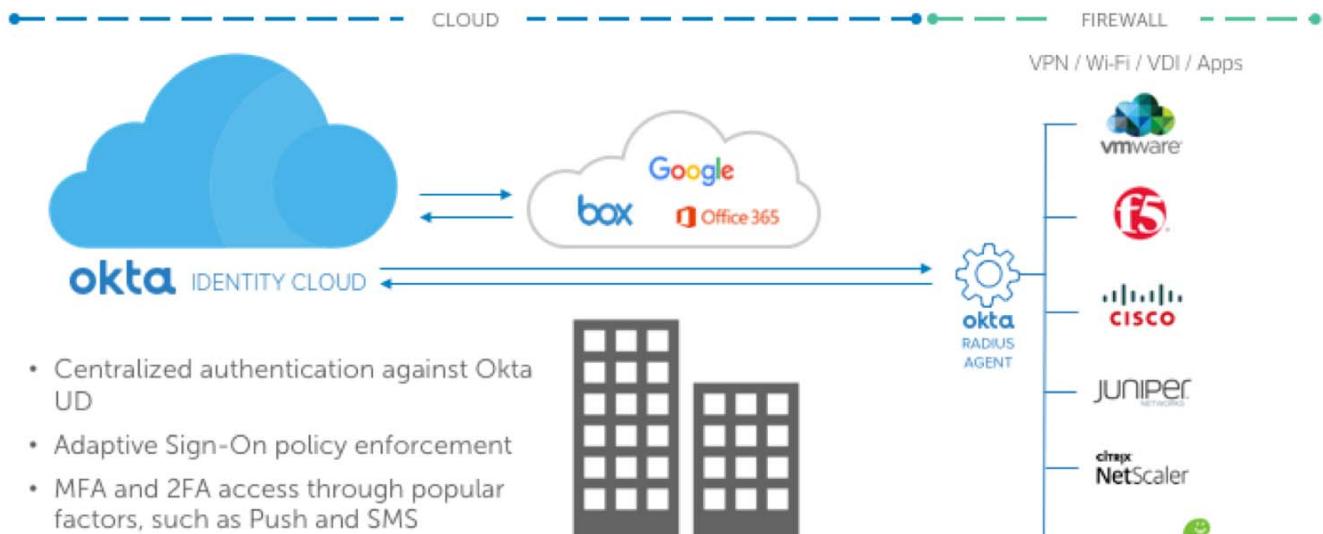
91

Additional Information

The term key rollover refers to a process where one key is systematically replaced by another key in SAML metadata. Because SAML entities (and therefore SAML metadata) are distributed, key rollover must be deliberate, so as not to break the key operations of a relying party.

To use the key rollover feature, you must generate an additional keypair and use a different key for SAML signing. Okta does not currently support using 3rd-party CA issued certs, so you must use self-signed certificates.

RADIUS Integrations Extend Okta IAM On-Premises

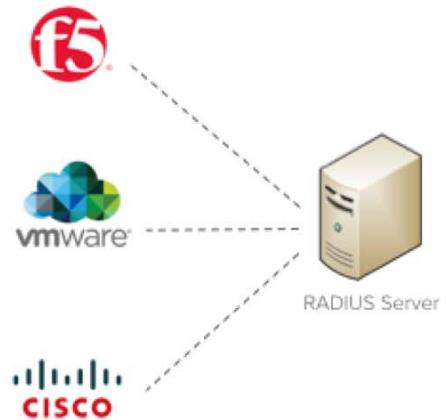


© Okta and/or its affiliates. All rights reserved. Okta Confidential.

92

How Does It Work?

- RADIUS has been around since 1991 and has been widely adopted on-premise for:
 - VPNs
 - VDI
 - Reverse Proxies
 - Wi-Fi (802.1x)



© Okta and/or its affiliates. All rights reserved. Okta Confidential. 93

Additional Information

RADIUS serves as a secure way to centralize authentication, authorization, and accounting, or AAA.

How Does It Work? Continued

The Okta RADIUS Agent can be used in place of a traditional RADIUS server, which routes an on-premise application authentication to Okta similar to cloud applications.



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

94

How Does It Work? Continued

- There are three authentication modes that can be used when an application or infrastructure is connected to the Okta RADIUS agent:
 - Primary Authentication
 - Multi-Factor Authentication
 - 2FA (Second Factor) Only



Please log in with your ronaldhouse credentials

[Forgot Password?](#)

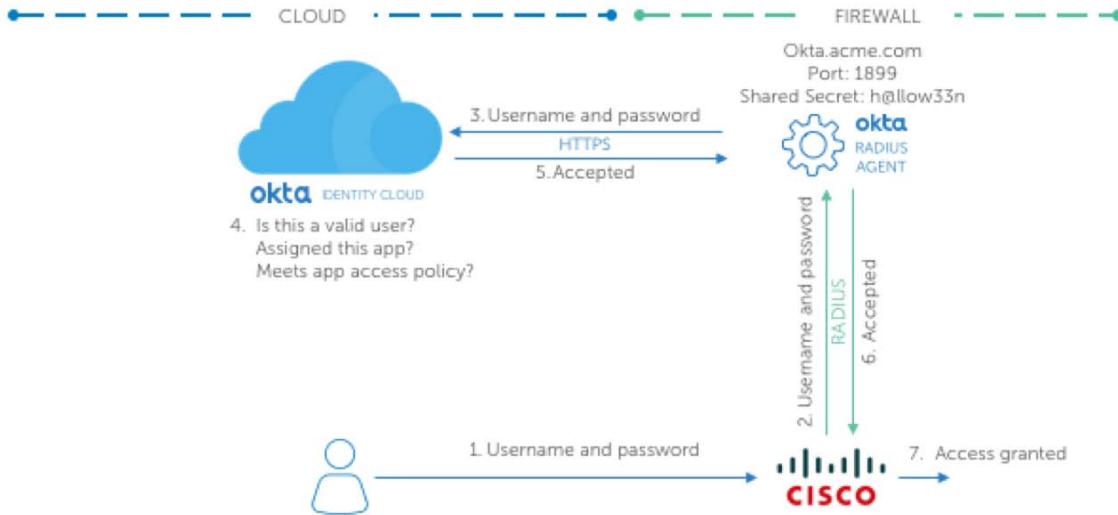
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

95

Additional Information

Support for these modes vary for each application or infrastructure.

How Does It Work? Continued



Primary Auth for Cisco AnyConnect VPN

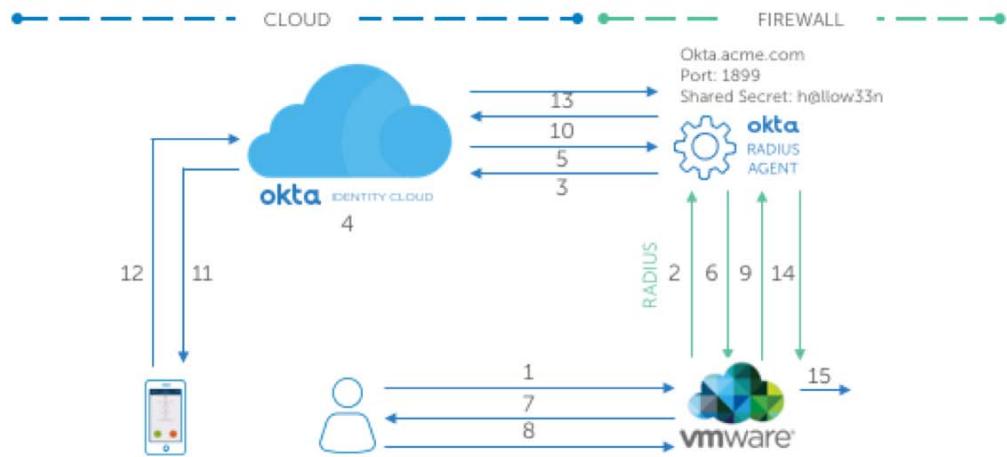
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

96

Additional Information

1. Users log into the Cisco VPN Client with a username and password.
2. Cisco ASA issues a RADIUS ACCESS-REQUEST to the configured RADIUS server (Okta RADIUS Agent) using the specified host, port, and shared secret.
3. Okta RADIUS Agent receives the request and communicates with the Okta cloud through outbound HTTPS.
4. Okta validates the user credentials and access policy for the application.
5. Okta replies and permits access.
6. Okta RADIUS Agent replies with an ACCESS-ACCEPT message to the ASA.
7. User is permitted into the VPN.

How Does It Work? Continued



MFA for VMware View

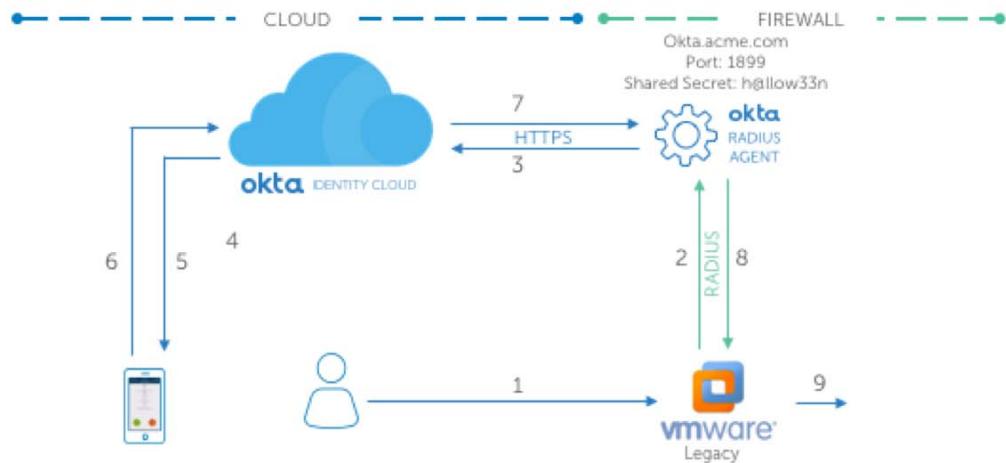
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

97

Additional Information

- Users log into VMware Horizon View client with username and password.
- VMware issues a RADIUS ACCESS-REQUEST to the configured RADIUS server (Okta RADIUS Agent) using the specified host, port, and shared secret.
- Okta RADIUS Agent receives the request and communicates with the Okta cloud through an outbound HTTPS.
- Okta validates the user credentials and realizes the user requires MFA.
- Okta replies with a MFA Challenge – requesting for the user to specify a factor.
- Okta RADIUS Agent passes a RADIUS ACCESS-CHALLENGE to VMware.
- VMware presents the ACCESS-CHALLENGE an input box message to the user.
- User types 1 (Verify), 2 (Push), 3 (SMS) or 0 (Cancel) to indicate the MFA to use.
- VMware passes the response as another RADIUS ACCESS-REQUEST to the RADIUS Agent.
- The RADIUS Agent passes the response to Okta.
- Okta issues a push command to Okta Verify on the user's mobile device.
- User taps Accept.
- Okta replies with Access-Accepted to the RADIUS Agent.
- The RADIUS Agent replies to VMware with the ACCESS-ACCEPT RADIUS response.
- Users is permitted into the Virtual Desktop.

How Does It Work? Continued



2FA Only for Legacy VMware View

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

98

Additional Information

1. Users log into VMware Horizon View with username and Push, SMS, or an Okta Verify code instead of a password.
2. VMware issues a RADIUS ACCESS-REQUEST to the configured RADIUS server (Okta RADIUS Agent) using the specified host, port, and shared secret.
3. Okta RADIUS Agent receives the request and communicates with the Okta cloud through an outbound HTTPS.
4. Okta validates the username and interprets the 2FA code.
5. Okta issues a push command to Okta Verify on the user's mobile device.
6. User taps Accept.
7. Okta replies with Access-Accepted to the RADIUS agent.
8. Okta RADIUS Agent replies with an ACCESS-ACCEPT message to VMware.
9. User is permitted into the VDI.

Remember

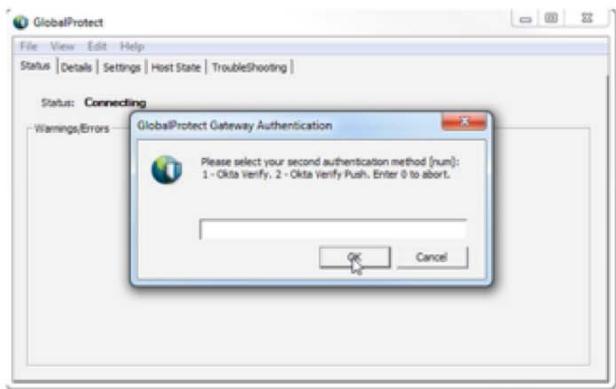
- Ongoing Considerations:
 - Okta RADIUS Agent only uses PAP protocol – 802.1x wi-fi uses EAP
 - MFA flow requires infrastructure to understand RADIUS ACCESS-CHALLENGE requests. (Most common VPNs & VDI support this but it varies)

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

99

Remember Continued

RADIUS ACCESS-CHALLENGE



- Presents a second authentication form during application or infrastructure log in.
- Is not universally supported, but most modern VPNs do support it
- Parses text from Okta's RADIUS CHALLENGE-RESPONSE message
- Response is done by the user with a # indicating the MFA preference.

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 100

An Easy Way to Put It All Together

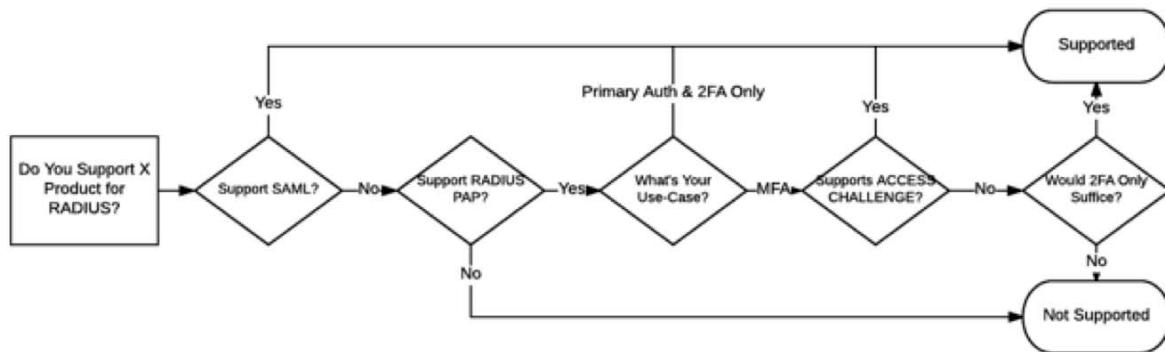
At a high-level, Okta connects to any RADIUS-enabled application that supports the PAP protocol.

Though it varies from product to product, Okta does support:

- Primary, MFA, and 2F-only authentication
- Into most VPNs, VDI, and reverse proxies

An Easy Way to Put It All Together Continued

In Depth – Vetting Okta RADIUS Support for a Given Product (Today)



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

102



Install the Okta RADIUS Agent

- Configure the Okta Sign-on Policy
- Install the Okta RADIUS Agent
- Test Using a RADIUS test Client

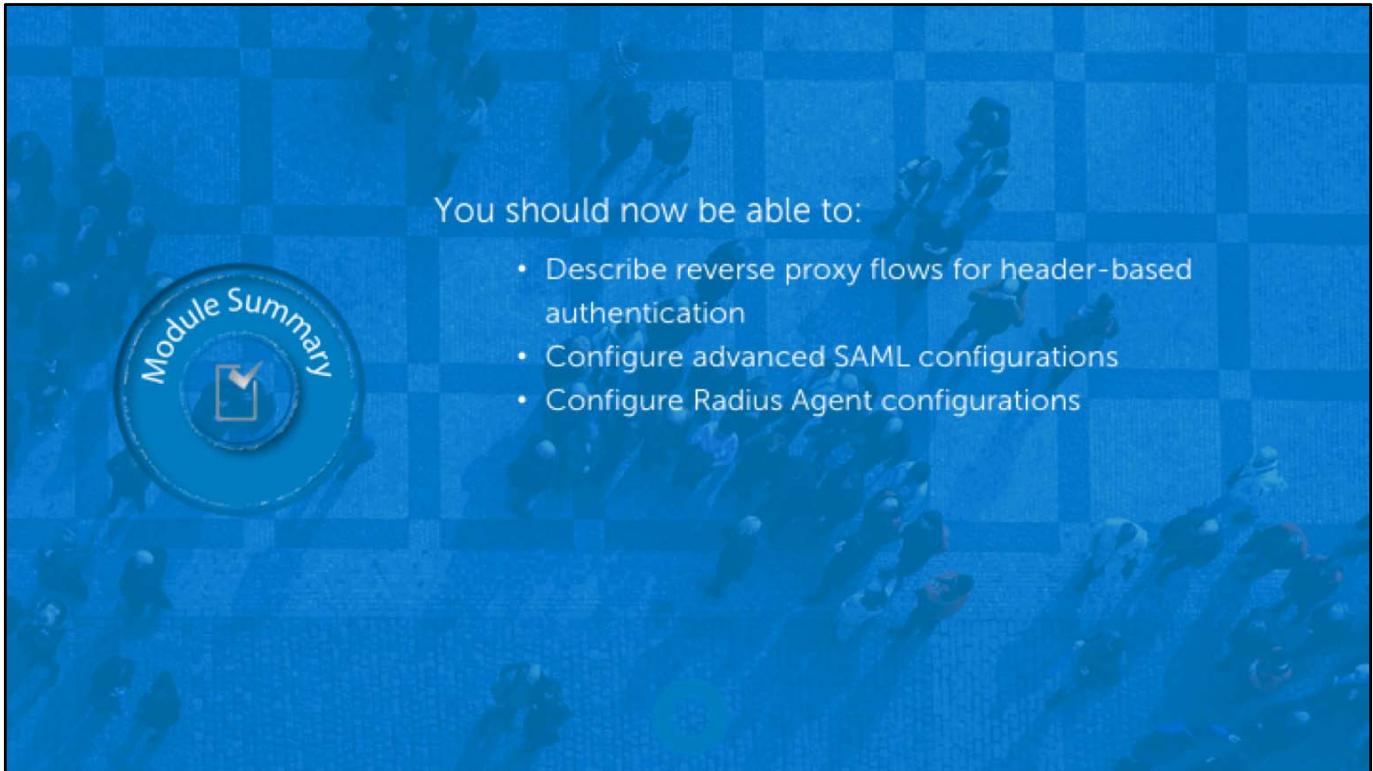


- Advanced SAML configurations are require custom integrations and should be used when appropriate for business needs
- Configure at least 2 RADIUS agents using the VPN for failover.
- Understand the limitations of how Single Sign-out works with Okta



When troubleshooting SSO solutions issues:

- Verify the Encryption certificate is valid
- Verify that ForceAuthn="true" is set for the AuthnRequest on the SAML Assertion using a SAML tracing tool
- Check to make sure the RADIUS port is 1812
- Verify the Share secret is configured correctly



You should now be able to:

- Describe reverse proxy flows for header-based authentication
- Configure advanced SAML configurations
- Configure Radius Agent configurations

End of module review questions:

1. How does Okta support advanced SAML configurations?

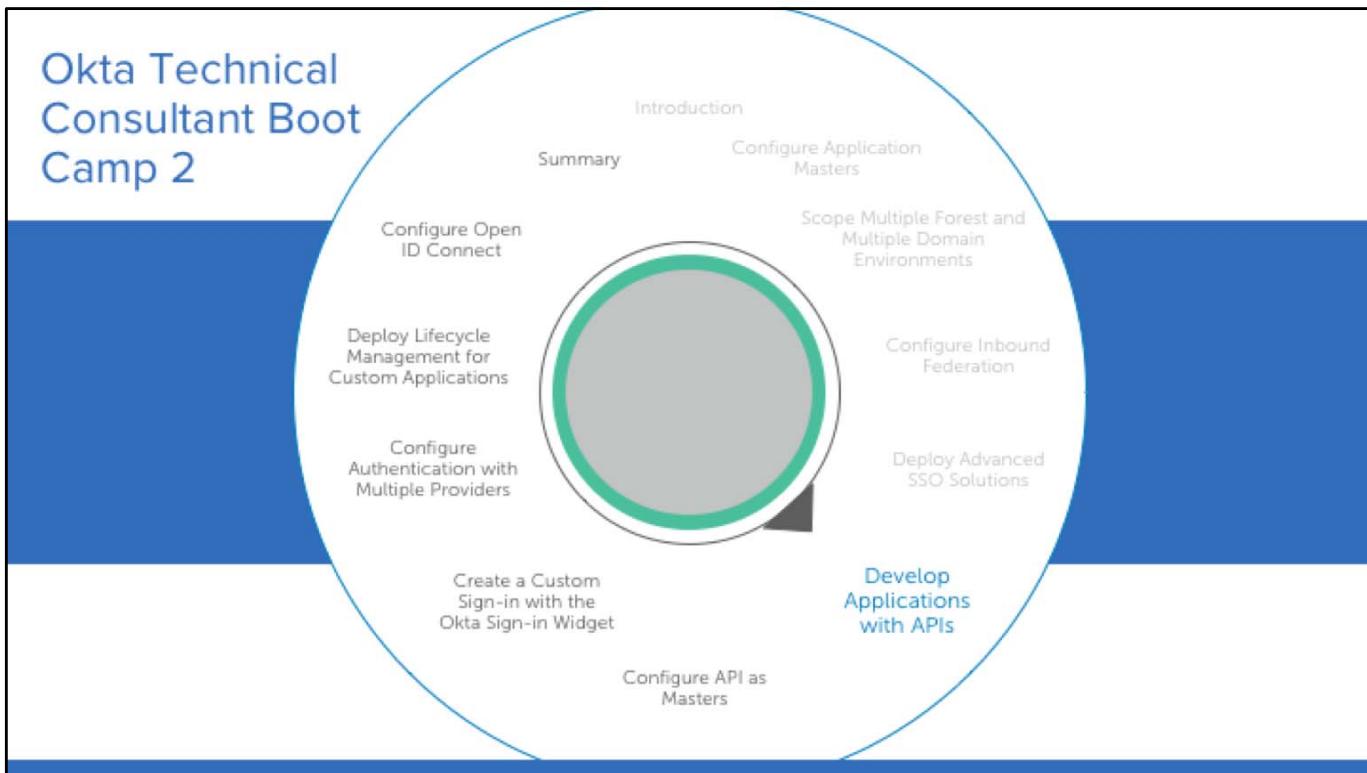
- a. OAN
- b. Not Supported
- c. WS-Fed Template
- d. AIW SAML Application

Source: Page 85

2. Which Radius port is used for the Okta Radius Agent?

- a. 80
- b. 466
- c. 1812
- d. 9000

Source: Page 96



Develop Applications with APIs

To help customers integrate custom applications with Okta there are many API options. Before integrating an application, you must perform some discovery and design a document to fully scope and successfully integrate the application.



Develop Applications with APIs

Scope the Business Requirements
Implement Okta API and SDK Solutions

Develop Applications with APIs Overview

In this module, you will learn about how to scope and implement solutions that use Okta APIs and SDKs.

Develop Applications with APIs

Overview

Enables you to...

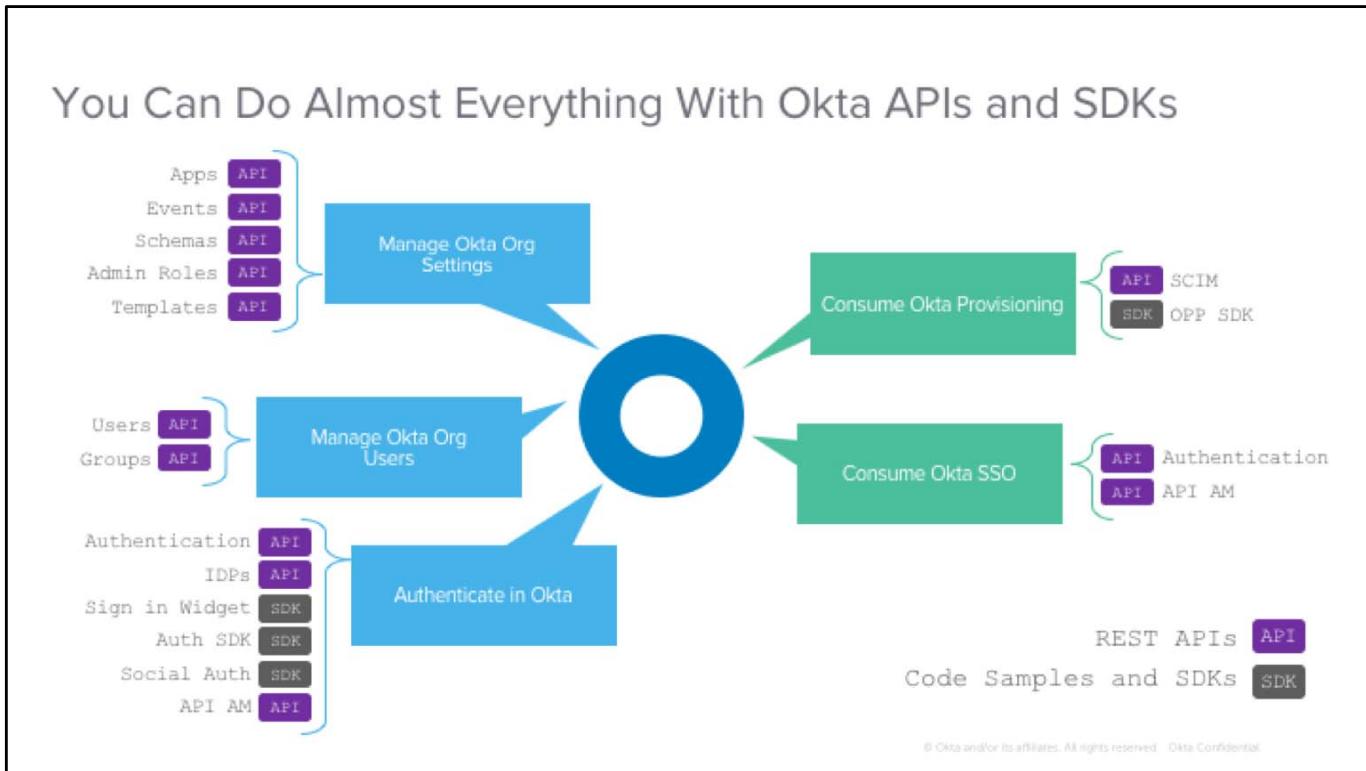
- Scope advanced projects that require custom development using Okta APIs.

Is important because...

- It sets the right expectations – effort, resources required, and final results – with customers.
- It reduces the risk of delivering solutions with reduced value.

Additional Information

This module focuses on scoping advanced projects that require custom development using Okta APIs and SDKs.



Additional Information

Okta Identity Cloud provides you with an extensive list of APIs, Codes, and SDKs that you can use for managing your Okta org from outside or for integrating consuming Okta services as a target app. With Okta APIs and SDKs, you can:

- **Manage Okta org settings** such as Applications, UD Schemas, or access the Okta Org logs via REST APIs.
- **Manage Okta org users** and groups via external applications. In this scenario, the external application "masters" the Okta users and groups.
- **Authenticate in Okta** or access Okta after log into another IDPs or applications.
- **Consume Okta Provisioning** in your custom application using SCIM APIs and Provisioning SDKs.
- **Consume Okta SSO** in your custom application using Authentication and API AM.

Because with great power comes great responsibility, your role as consultant is to implement solutions that leverage Okta APIs and SDKs responsibly.

Tips:

- You further explore most of the Okta APIs and SDKs later in this course.
- The Okta developer website – <https://developer.okta.com> – provides resources and sample code to help customers integrate applications and use extended Okta features through API requests.

What Okta APIs and SDKs Implementations Require

- Tailor-made solutions outside Okta.
- Examples:
 - An integration with a customer's custom or proprietary system that will consume Okta APIs.
 - A development of custom code, consuming Okta's SDKs and APIs.
- An implementation that includes:
 - Discover use-cases for implementation.
 - Design a solution and application architecture.
 - Develop and validate solution.
 - Deploy.
 - Support and handover to the customer.

aka,
App Development
Lifecycle

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Most of the use-cases where Okta APIs and SDKs are used involve either making a customer custom/proprietary system consume Okta APIs or develop applications that will consume Okta SDKs and APIs.

Integrating systems that belong to your customer or developing solutions from scratch are considered advanced projects.

Implementing advanced projects require a structured approach to ensure the business value is delivered, while setting the right expectations.

This structure approach is the Application Development Lifecycle.

App Development Lifecycle



1. Discover Use Case(s)
2. System Architecture
3. Custom Application Design
4. Custom Application Code
5. Custom Application Unit Test
6. Deployment
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

When helping customers integrate any application, especially using the Okta API, it is important to follow an established flow and document all customer requirements and expectations.

1. Discover Use Cases



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

To properly scope the integration, you must perform use case discovery. As part of the discovery:

- There is a handoff between Sales and Technical Engagement Managers (TEMs)
- The Statement of Work is created
- A customer workshop is conducted

Customer Workshop - Sample Agenda

Current State	Desired Future State
<ul style="list-style-type: none"> • High level review of existing customer environment • Existing custom applications • Existing target applications <ul style="list-style-type: none"> • SSO protocol (SAML/OIDC) • Downstream provisioning 	<ul style="list-style-type: none"> • End-user experience • Custom applications: <ul style="list-style-type: none"> • Environments • Users • UD attributes • Management: Admin user profile, Self-service user profile, Self-service password, Self-service lockout, Session • Group membership <ul style="list-style-type: none"> • Integrated Okta target applications <ul style="list-style-type: none"> • SSO protocol • Downstream provisioning

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The agenda for the customer workshop should include a review of the current customer configuration. With the current state outlined, move on to discuss the desired future state with Okta. In this view, be sure to include the following for each custom application:

- **Environments:** Prepare Okta and application sandboxes and Okta production environments.
- **Users:** Have a migration plan for existing users outlining how to handle custom import applications, password configurations, and account activations. For new user accounts, you must have a plan around custom registrations, password configurations, and account activations.
- **UD attributes:** Scope and provide a comprehensive attribute including the attribute source.
- **Admin user profile:** Have plans for the user management.
- **Self-service user profile:** Create a plan for self service options.
- **Self-service password:** Create a plan for user-enablement of password management (change password and forgot password flows).
- **Self-service lockout:** Develop a workflow on how to handle user management for locked out accounts.
- **Session management:** Provide plans around login flows, password policies, sign-on policies, and MFA factors and policies.
- **Group membership:** Identify potential group memberships and rules to help with user application assignment.

App Development Lifecycle



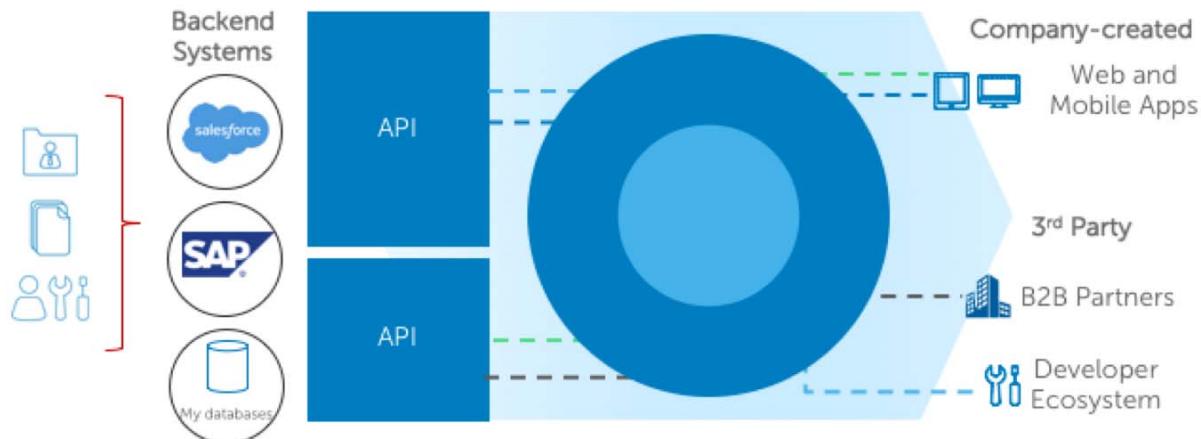
1. Discover Use Case(s)
- 2. System Architecture**
3. Custom Application Design
4. Custom Application Code
5. Custom Application Unit Test
6. Deployment
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

2. System Architecture



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Based on the use cases, SOW, and workshop there will be an overall system architecture which includes anything to do with or that Okta touches. Depending on the size and complexity of the project, this might be a deliverable. Existing customers should have a documented system architecture from the previous deployment; this might require updating if the customer has performed changes without Okta involvement.

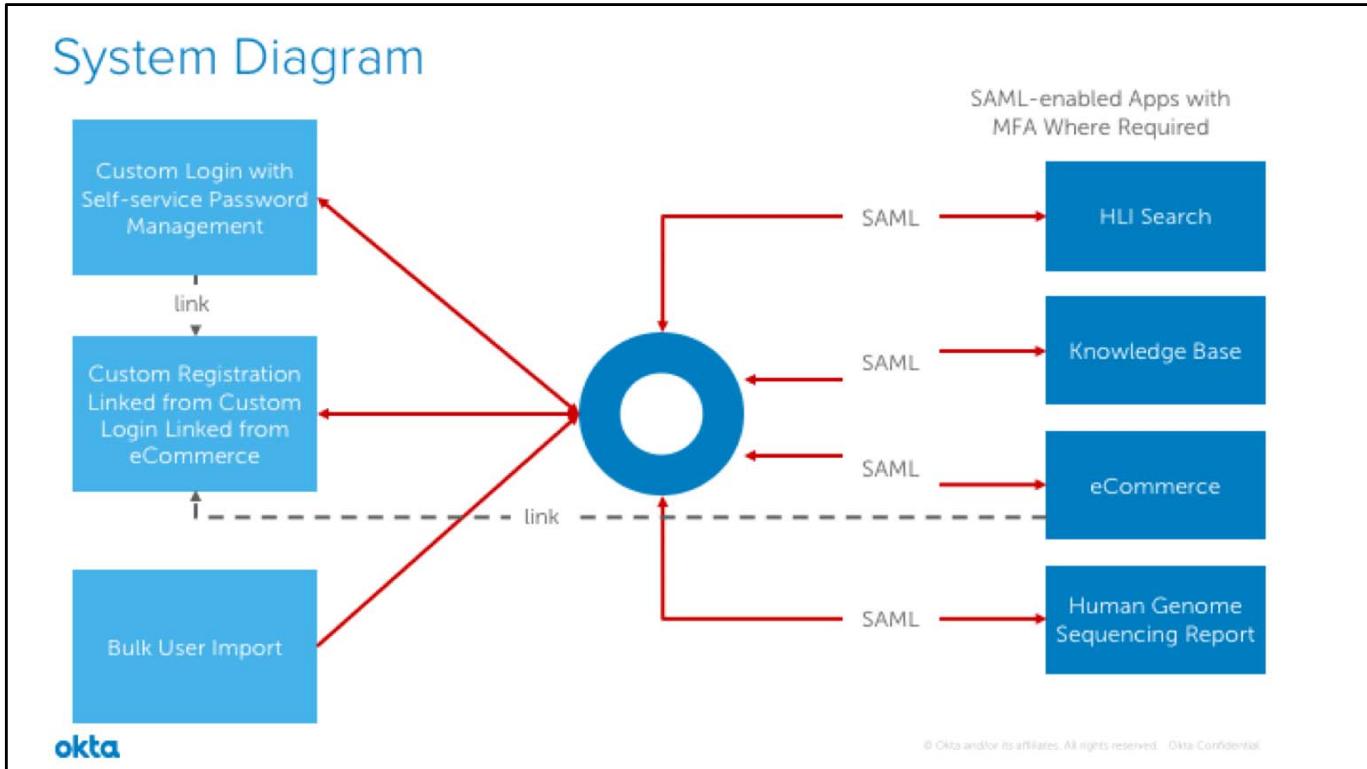
In a broad scope, the system architecture includes:

- Okta Org Configuration
- Custom Applications, their purpose and features
- A system diagram with high-level representations of primary components.

It also shows where and how the custom applications interface with the overall system. For smaller projects the system architecture is standard and does not need much documentation. For example, some projects can contain complicated group structures with sub containers of user which are used to guide associations, mappings, and so on. This information must be documented somewhere.

A sample Okta org configuration might include a group structure, a user source, MFA factors, a password policy, a sign-on policy, a 3rd party IdP, and application provisioning.

Custom applications might include a login application, a portal application, a registration application, and self service options such as a forgot password flow.



Additional Information

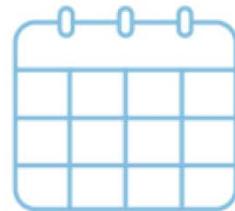
For the system diagram, provide enough detail to understand the relationship and data flow for all major components.

Also consider multiple diagrams showing:

- Current state
- Transition state
- Proposed end state

App Development Lifecycle

1. Discover Use Case(s)
2. System Architecture
- 3. Custom Application Design**
4. Custom Application Code
5. Custom Application Unit Test
6. Deployment
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

3. Custom Application Design Document (CADD)

Why it is needed:

- CADDs show detailed information around the application design
- Each custom application requires a distinct CADD for clear separation of requirements and expectations
- When a template is used, each CADD helps ensure complete custom application scoping

After creation, the CADD:

- Must be reviewed and signed off by the customer
- Becomes the source of truth for:
 - Use case information
 - Business logic
 - Attributes
- Is revised when questions or impasses arise
- Is reviewed & revised and a change order is initiated if a customer initiates a change request

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

For each custom application in the system architecture document the specifics in the CADD. When documenting each custom application separately, it might help to split up the work

Separate CADDs also help the project - if one application implementation gets stalled it might have less of effect on other application integrations because the unit tests and deployments are kept distinct from other applications.

The CADD is where the requirements from SOW and details from workshop are documented. While some projects have multiple custom applications and it is a best practice to keep separate CADDs, there can be reason to combine, including:

- The customer might be comprised of many teams with each team having a specific focus
- Often the contract is written by management, but now you are working with engineering; many things could have gotten missed and not contracted

Post CADD creation, you might also deal with:

- Time has passed since the contract was written and things are dynamic; people are busy and might not have spent enough time on the design, so items are later uncovered.
- Scope creep, new requirements, or a new perspective emerges about a spec and it changes your design as well as the subsequent code and test
- If a change is identified that will cost you time, schedule, or both and it is driven by customer changes, but not something you overlooked, you will must justify a change order.

CADD Sections (Sample Text)

INTRODUCTION

The following document outlines the requirements and expectations of the company application integration with Okta.

SCOPE

This application configuration meets the following conditions:

- Full user CRUD for on and off boarding
- Okta Verify for MFA
- SSO through SAML for no user-defined passwords
- Employee access to company information
- No mobile native application access

USE CASES

All employees require access to company records.

Mobile web application is supported to read information, but native application is not available.

When off network, employees must be prompted for Okta Verify.

OKTA ORG CONFIGURATION

The following features must be enabled:

- Core
- SSO - 5 apps
- Universal Directory
- Provisioning
- Adaptive MFA

APPLICATION DESIGN



USER ATTRIBUTES

The following attributes must be passed:

- Username
- First name
- Last name
- Employment status

CONFIGURATION FILE

Parameters:

Environment requirements:

Acceptable values:

LOGGING

Debug log requirements:

Audit log requirements:



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

When completing the CADD, identify the following:

- The use cases this application implementing.
- The required configurations in the Okta org to support the application.
- For the application design, as much detail as possible to help with coding.
- The various attributes, including:
 - User attributes
 - Okta Universal Directory attributes
 - Source system attributes
 - Target system attributes
 - Required attribute transformations
 - Required and optional attributes
- For the configuration file, state the available parameters without having to open the code, what is needed to move between test, QA, and production, and the legal or acceptable values for each entry.

For logging, identify the ideal information that will help people debug any issues including the log levels and log location.

For audit purposes, scope the audit log to include helpful information around logins and imports including the exact information to capture.

Scenario: Populate a CADD

A customer wants to integrate an application that does not require usernames or passwords for authentication. Access to the app must be gated by the sales manager team. The customer has multiple Active Directory servers and Universal Directory as user stores.

Based on this information:

- Which Okta features should be enabled?
- What are possible use cases?
- What can you initially scope?



okta

Custom Application Design Approach

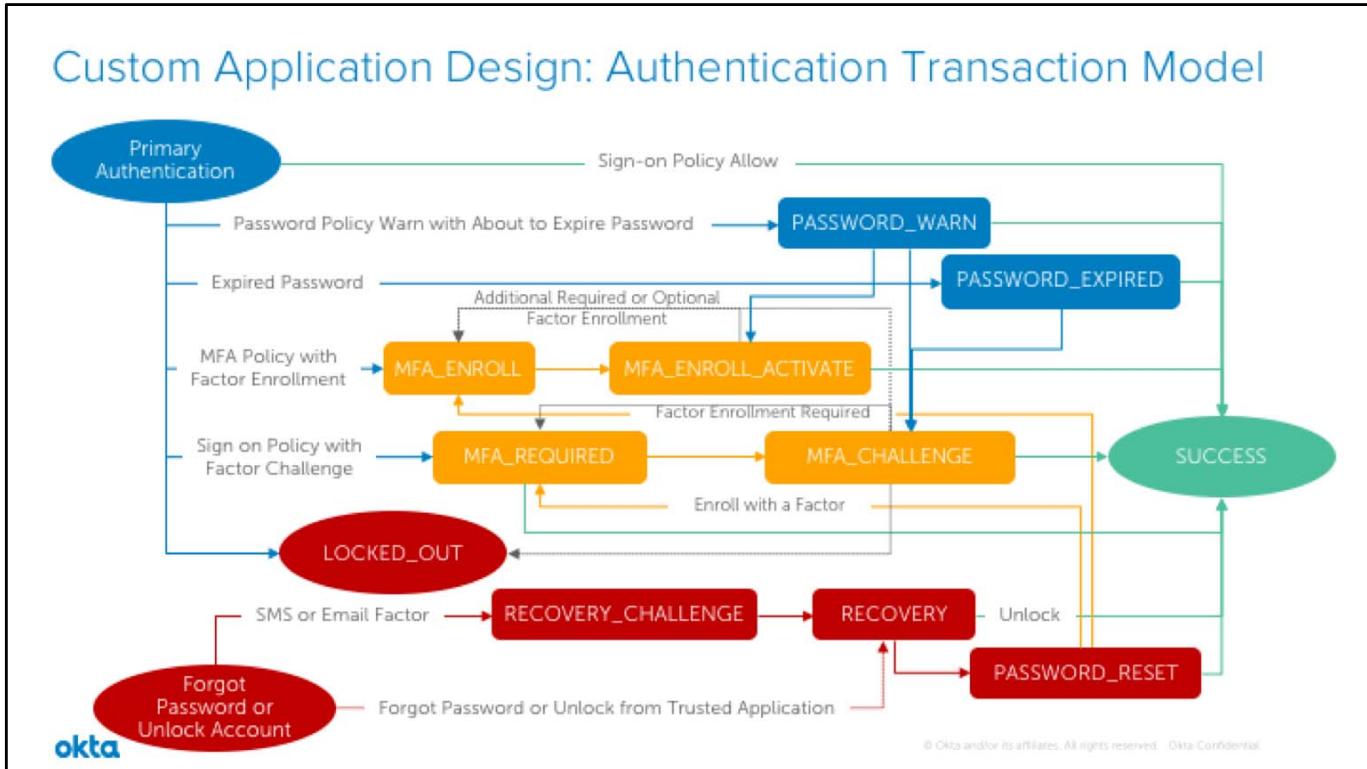
- Okta Platform is set of REST APIs
 - POSTMAN is great tool to use before coding to become familiar with the API
 - Because several Okta workflows are implemented and enforced as a state machine, you must honor transitions
 - stateToken is passed with API to maintain state
- Okta platform API applications:
 - Handle branding and custom logic
 - Embed into existing code
 - Have been previously created

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

When designing applications, remember that applications to solve common issues have probably been written before, so save time by reusing an existing application as a template.



Additional Information

The Authentication API is a stateful API that implements a finite state machine with defined states and transitions. Each authentication or recovery transaction is issued a unique state token that must be passed with each subsequent request until the transaction is complete or canceled.

On the slide is a complete transaction model view. When actions are performed, APIs must be called in sequence. For example, you cannot call a forgotten password reset without first answering security question.

Rate Limit Checks

- Okta does rate limiting on all incoming APIs regardless of origin
- The SDK will trap error, but you can configure custom code that if the rate limit is reached, wait then retry API call.

Response Header	Description	Example
X-Okta-Request-Id	<ul style="list-style-type: none"> The unique identifier for the API request. 	WJHi5Ozxwu3pGCLFXDRanwAAaY
X-Rate-Limit-Limit	<ul style="list-style-type: none"> The rate limit ceiling (configurable) that is applicable for the current request. 	1200
X-Rate-Limit-Remaining	<ul style="list-style-type: none"> The number of requests for the current rate-limit window. 	1199
X-Rate-Limit-Reset	<ul style="list-style-type: none"> The remaining time in the rate-limit window before the rate limit resets. This is expressed in UTC epoch seconds. 	1485955872

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

API endpoint throttling limits have been created to protect the Okta service from load spikes or service interruptions caused by submitted requests - whether unintentional or as a Denial of Service (DDOS) attack. This strategy has been adopted across the industry in many SaaS applications and platforms to maintain consistent service levels for customers. Requests that hit the rate limits return a "429 Too Many Requests" HTTP status code.

When working with the Okta API, it is important to know how many users you are working with. For many customers, the default rate limits are acceptable. On a case-by-case basis, rate limits for an Okta org can be increased temporarily or permanently by operations.

Configuration Files

- Are necessary to deploy to test, QA, and production
- Can be setup differently
- Expose details of application for changes without recompiling

File Sample:

```
<add key="okta.ApiToken" value="00A11INlImjHB2KXmodqWwpHjASS3jnwxfxW" />
<add key="okta.ApiUrl" value="https://subdomain.okta.com" />
<add key="custom.MfaRequiredGroup" value="MfaRequired" />
<add key="custom.IpGateway" value="72.222.777.505" />
<add key="custom.OktaPushWaitTimer_ms" value="10000" />
<add key="custom.OktaPushTimerCycle" value="3" />
<add key="custom.rsaUsernamePrefix" value="true" />
<add key="custom.countryCode_json" value="countryCode.json" />
<add key="custom.helpUrl" value="/Home/Help" />
<add key="custom.selfServeLinkExpiry" value="59" />
<add key="PswdComplexity" value="Passwords must be at least 8 characters long"/>
```

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

When working with configuration files, older asp.net MVC 5 and Web API 2 used an xml file, while newer asp.net core uses JSON.

Some customers use the server environment variable to help keep secrets, such as apikey, out of source code.

Logging

- Are needed for debugging code after deployment
- Might be the only method to debug site-specific issues
- Should:
 - Be run early and often
 - Be made readable so customers can debug their own issues
 - Have configurable log levels for better performance
 - DEBUG
 - INFO
 - WARNING
 - ERROR

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

A popular logging library is log4net.

App Development Lifecycle

1. Discover Use Case(s)
2. System Architecture
3. Custom Application Design
- 4. Custom Application Code**
5. Custom Application Unit Test
6. Deployment
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

4. Custom Application Code

Technique and Styles	Code Reuse and SDKs
<ul style="list-style-type: none">• Okta Platform API code can be written in any language that supports HTTP.• The Okta Core library (NuGet) is freely available.• Most common development languages are:<ul style="list-style-type: none">• dotNet/C#• Java• JavaScript• PowerShell	<ul style="list-style-type: none">• Okta provides SDKs for common languages.• Not all SDKs are complete.• The API Helper classes extend SDK capabilities; inquire with Okta Professional Services.• SDKs support common features.• If not using the SDK, then custom code is required for:<ul style="list-style-type: none">• JSON serialization/deserialization• Rate-limit checks• Okta exception handling• Model classes

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

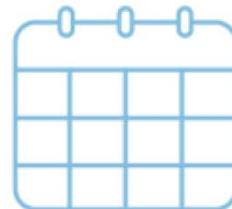
Additional Information

The .NET/C# SDK is currently the most complete, while the Java and PowerShell SDKs are limited.

The application type used depends on the use case. In the Configure API as Masters module, you will see how to use built-in timer capabilities and in the Configure Authentication with Multiple Providers module, you will configure a web application using ASP Net MVC 5 because it is an interactive web based transaction.

App Development Lifecycle

1. Discover Use Case(s)
2. System Architecture
3. Custom Application Design
4. Custom Application Code
- 5. Custom Application Unit Test**
6. Deployment
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

5. Custom Application Unit Test

- Test each designed feature.
- Create a test document to track the following:
 - Tests performed
 - Data used
 - Pass criteria
 - Results
- Make the test traceable from the design document to implementation to unit test.

okta

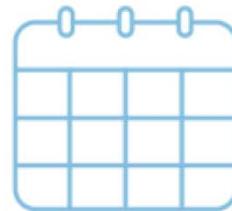
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

While you probably will not be provided enough project time to do very formal test plans, you must record the tests completed and the results.

App Development Lifecycle

1. Discover Use Case(s)
2. System Architecture
3. Custom Application Design
4. Custom Application Code
5. Custom Application Unit Test
- 6. Deployment**
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

6. Deployment

- Deliver source and compiled code; the configuration file makes compiled code portable.
- Provide customer document for application deployment; this can be addendum to the design document
- Be ready if you are asked by customer to deploy to their infrastructure:
 - Try to limit to test environment only; limit your liability
 - Get VPN access
 - Get local credentials
 - Deploy on the case by case basis

okta

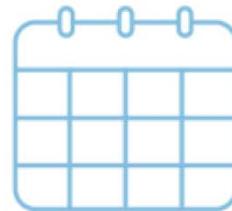
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

It might be difficult to get access to and schedule VPN or server access, so do not attempt to personally deploy, but rather assist the customer.

App Development Lifecycle

1. Discover Use Case(s)
2. System Architecture
3. Custom Application Design
4. Custom Application Code
5. Custom Application Unit Test
6. Deployment
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

7 and 8. Customer Support

Test

- The customer is responsible for final end to end testing, but Okta provide assistance creating test plan.
- The developer assists investigating test failures; debug logs are helpful.
- The customer documents failures and submits information.
- Applicable unit tests are repeated for revised code and then redeployed.
- The code is delivered 'as-is', Okta does not warranty custom applications.
- After signing off for application acceptance the customer owns the code.

Go Live

- Typically, by the time you are ready for Go-Live, the confidence level should be very high.
- You should make yourself available with 30 minutes notice during Go-Live, but not to participate in the full process.
- The customer should be aware of the steps to revert to previous configuration if the need arises.

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Test:

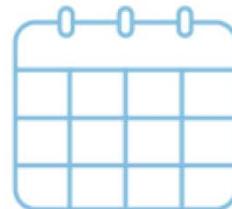
- Set customer expectations that they perform the testing, but that you are available to help investigate issues.
- Proper testing of a complex application will take days (or longer) and usually require a larger team coordination, which can get very time consuming:
 - End-users
 - App owners
 - Business owner
 - IT (infrastructure)

Go Live:

- Set customer expectations to confirm that everything should be tested, including deployment, before go-live.
- Check the SOW for what was promised, but similar to testing this can be very time consuming.

App Development Lifecycle

1. Discover Use Case(s)
2. System Architecture
3. Custom Application Design
4. Custom Application Code
5. Custom Application Unit Test
6. Deployment
7. Customer Test Support
8. Customer Go Live Support
9. Customer Follow-up



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

9. Customer Follow-up

- After the customer signs off for the final tests, they:
 - own the code
 - own the source
 - are responsible for maintenance
- Okta Customer Support will have little insight into custom applications and might reach out if there is an issue.
- If the customer demand becomes too great for you, refer to a TEM for new PS SOW.
- Remember that the design and test documents are a great resource to settle discussions.

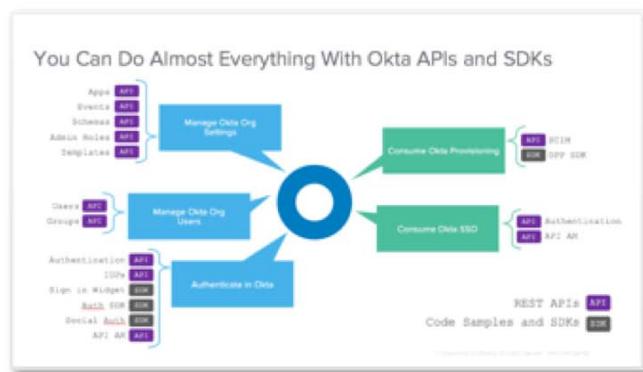
okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

- Document your design, review, and revise documents with customers.
- Have the customer sign to approve the CADD.
- Remember that changes to documented design are possible by issuing change orders.
- Raise issues as soon as possible - preferably during design phase
- Log everything
- Externalize data into a configuration file.
- Remember the Okta API rate limits on endpoints and code accordingly
- Have customer take responsibility for the look of the UI



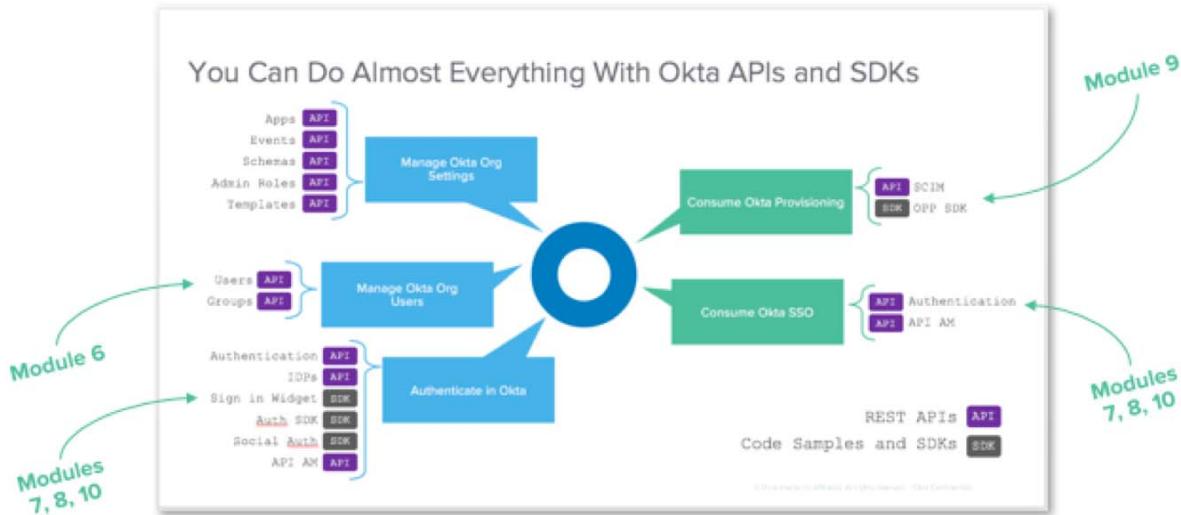
To Learn More About APIs and SDKs

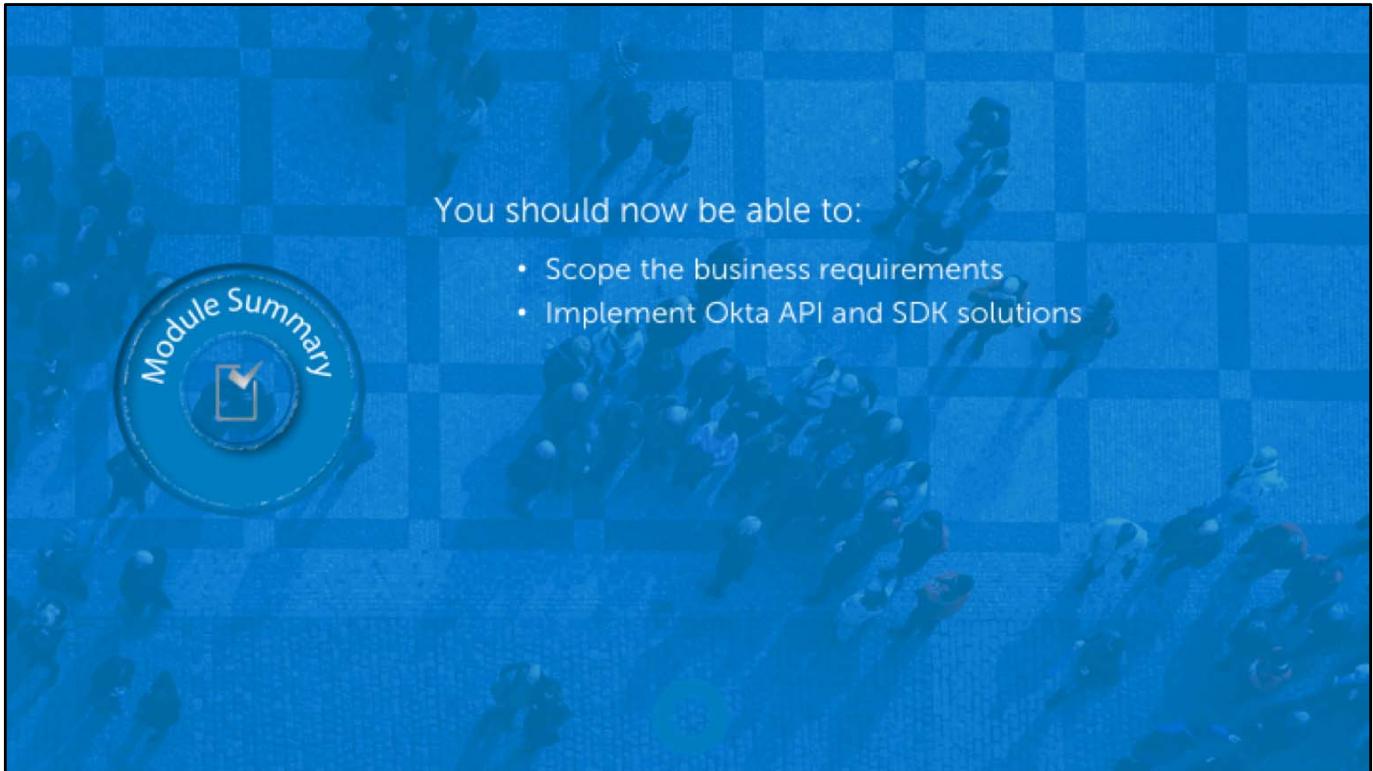


<http://developer.okta.com>
<https://github.com/okta>

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Coming Up Next....





You should now be able to:

- Scope the business requirements
- Implement Okta API and SDK solutions

End of module review questions:

1. When do you work out the specifics of the use cases and solidify the contents of the SOW with the customer?

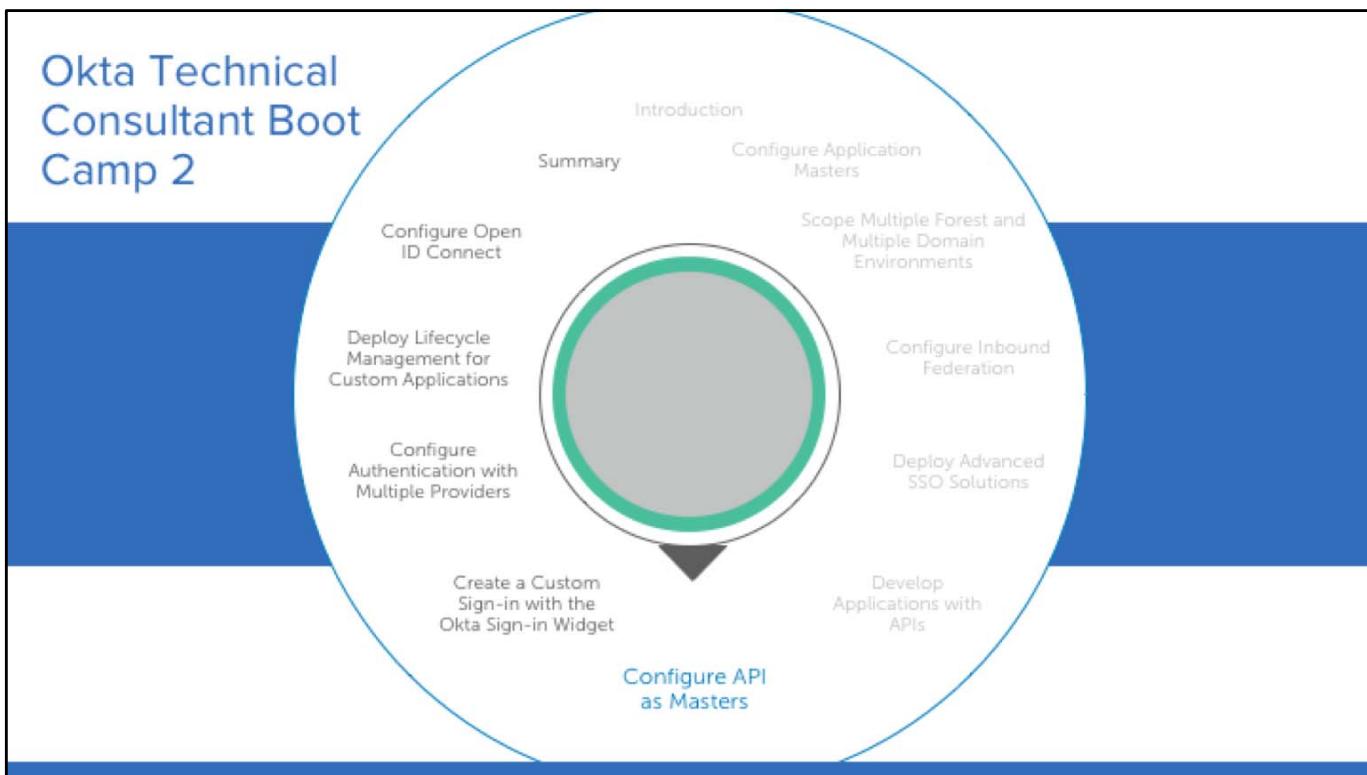
- a. When creating the CADD
- b. When creating the unit tests
- c. During the handoff portion of the use case discovery
- d. During the customer workshop portion of the use case discovery

Source: Page 119

2. What is true of customer support during the test phase?

- a. Okta documents failures and submits information.
- b. The custom application code is verified and warrantied by Okta.
- c. After signing off for application acceptance the customer owns the code.
- d. The customer is responsible for creating the test plan and completing the final end to end testing.

Source: Page 134



Configure API as Masters

To help customers quickly integrate applications or external users with an Okta org, Okta provides the ability to configure APIs as masters. Solutions for using APIs as masters include:

- Preloading users into Okta
- Integrating users managed through a non-standard (not Active Directory or LDAP) directory service, such as a local database
- Provisioning users from an Oracle database
- Using SQL Server Integration Services to perform CRUD into Okta based on SQL events



Configure API as Masters

Scope the Business Requirements
Describe the Okta Solutions
Configure API Masters

Configure API as Masters Overview

In this module, you will scope the business requirements, match Okta solutions, and configure an API master.

This module consists of labs and review questions.

API as Masters Overview

Enables you to...

- Assist Okta customers with a custom app-as-a-master integration using the API

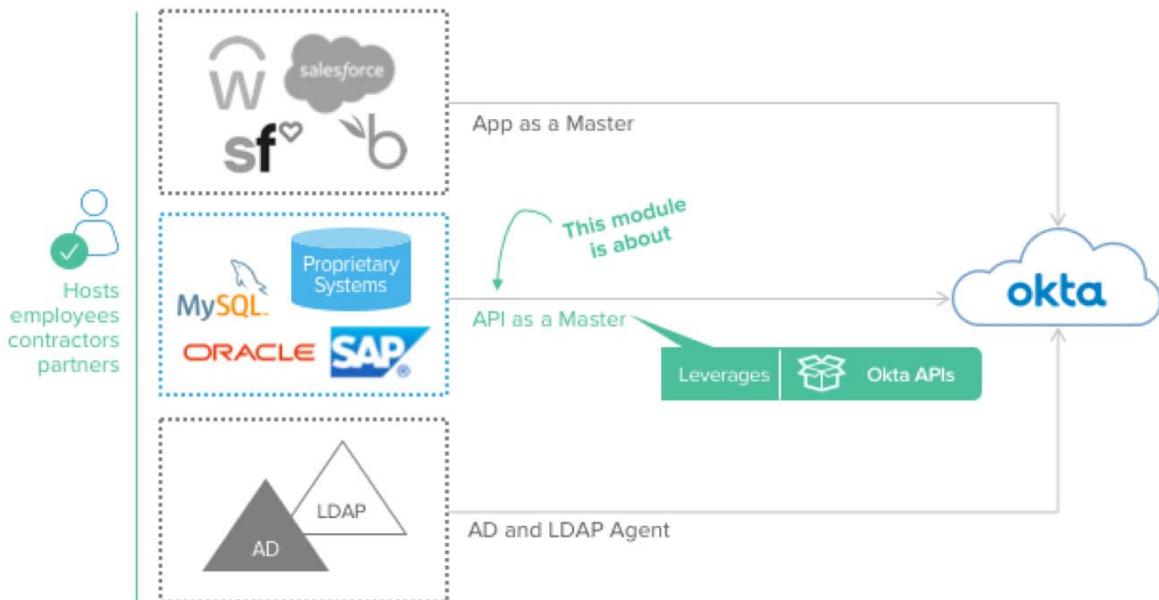
Is important because...

- Some customers have proprietary applications to integrate with Okta.
- Customers with diverse user-bases might need help using the API to integrate a user store

Additional Information

In the previous modules, you learned about using applications and Active Directory as a master in advanced scenarios. This module focuses in API as a Master.

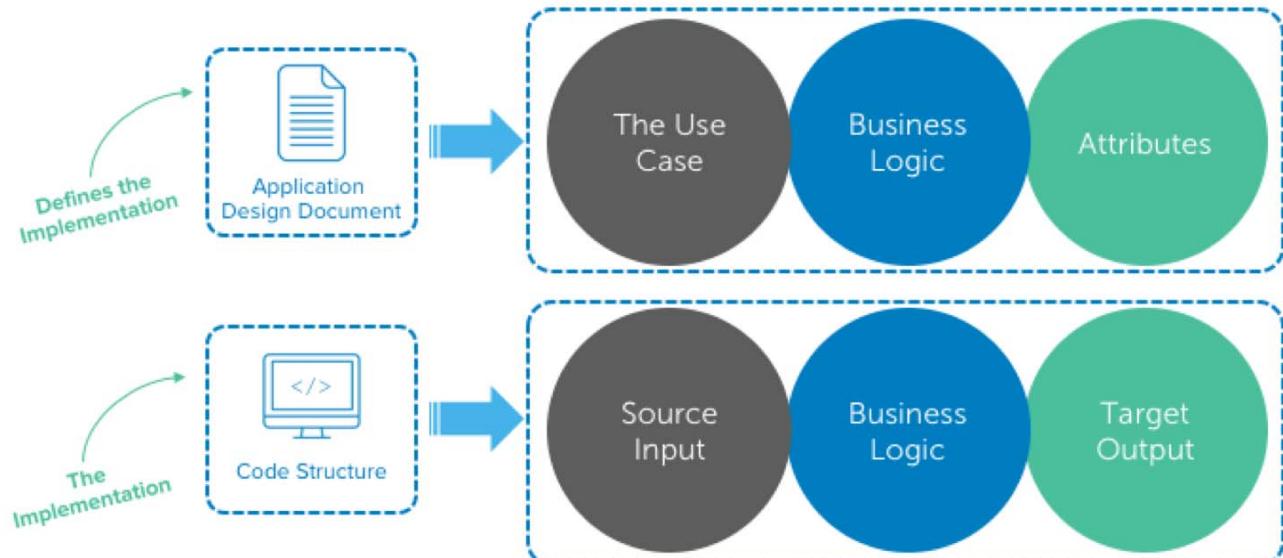
API as a Master: Overview



Additional Information

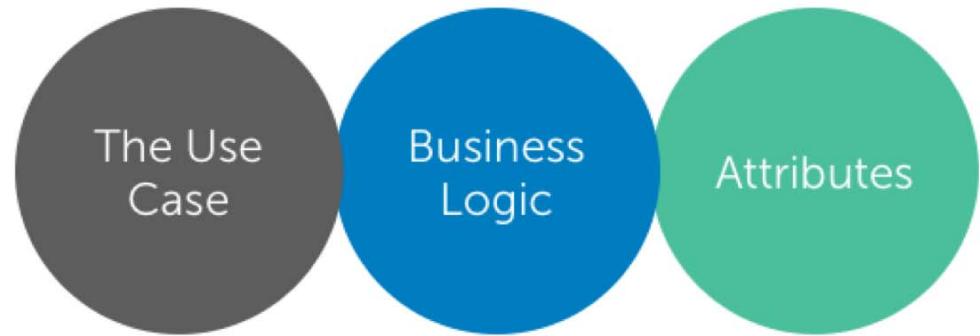
API as a Master is used when you need to onboard users from systems that are not supported using an agent or application as a master integration.

API as a Master: Implementation Components



© Okta and/or its affiliates. All rights reserved. Okta Confidential. 145

Application Design Document



Application
Design Document

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 146

Additional Information

To properly scope the integration, create an application design document to determine:

- The use case
- Business logic
- Attributes

The Use Case

- Basic CRUD operations**
- How are you activating new users?
• How are you creating passwords?
• Is a security question required?

- Application creation and assignments**
- Which authentication method will be used?
• If not SAML, what are the auth requirements?
• What happens after a successful auth?
• What happens for a failed auth?
• Who requires access to the app?
• Who will manage the app?

- Okta group creation and assignments**
- Are Okta groups required to help with provisioning or application assignment?
• Should group membership rules be established to help with application assignment?

- Auditing requirements**
- Which information requires tracking or monitoring?
• How often should the information be audited?
• Who requires the ability to audit the information?



Application Design Document

© Okta and/or its affiliates. All rights reserved. Okta Confidential 147

Additional Information

Before integrating applications using the API, you must know the factors behind the integration. For example, is this application simply being integrated for SSO access? If provisioning is required, which functions are supported?

Also worth scoping are the application and Okta group creations and assignments. If groups or roles are required as part of an assignment to a role or function to use the application, perhaps attribute mapping or group membership rules must be considered.

To help IT manage the application, there also might be auditing concerns that should be considered for reports. Ask what the concerns are and write them down.

Business Logic

Search criteria

- What is being used to search for users?
- Are complex searches, such as a combination of attributes, required?

Source input to current state

- Which user stores are in place?
- If there are multiple user stores, which is the source of truth?
- How are duplicate matches to be handled?

Error reporting

- How are application handling errors reported?
- Is write-back to the data source available?
- How is logging done and can it be captured through Okta?



Application Design Document

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 148

Additional Information

With the use case established, the business logic or handling of information must be scoped.

For example, what is the information required for searching and reporting? If multiple user stores are in place, de-duplication of user accounts should occur across the user stores or leveraging Universal Directory attribute-level mastering might be necessary.

To minimize help desk impact of errors or anomalies of user states and make the application more robust, it is also important to scope how handling occurs for users in unknown states.

Attributes

Mapping

- What are the required source attributes?
- Which attributes should be mapped to Okta attributes?

Custom attributes

- How should Universal Directory be used to handle custom attributes?

Attribute transformations

- Are there multiple attributes that guide data transformation?
- Are there any required:
 - JSON/XML lookup tables
 - RegEX manipulations
 - External references



Application Design Document

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 149

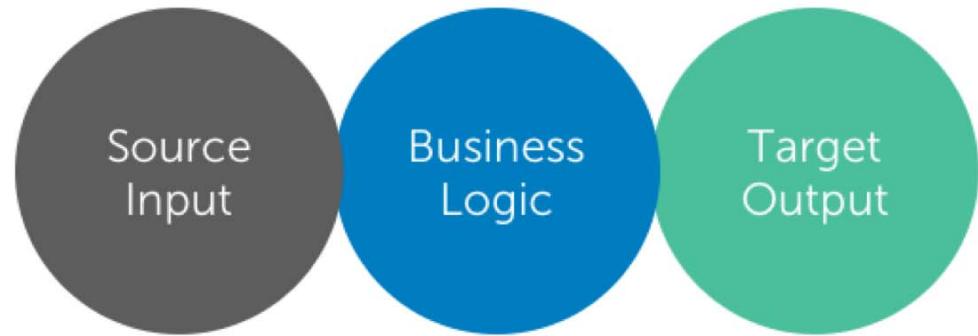
Additional Information

With the use case and business logic established, the specific information around attribute handling needs to be scoped.

For example, what are the required user attributes for the user to authenticate to the app?

If provisioning is a factor, which attributes are provisioning enabled and which operations can be performed? For example, can all user information be updated between Okta and the application or are only a few attributes modifiable, such as first name, last name, and username?

Code Structure



Code Structure

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 150

Additional Information

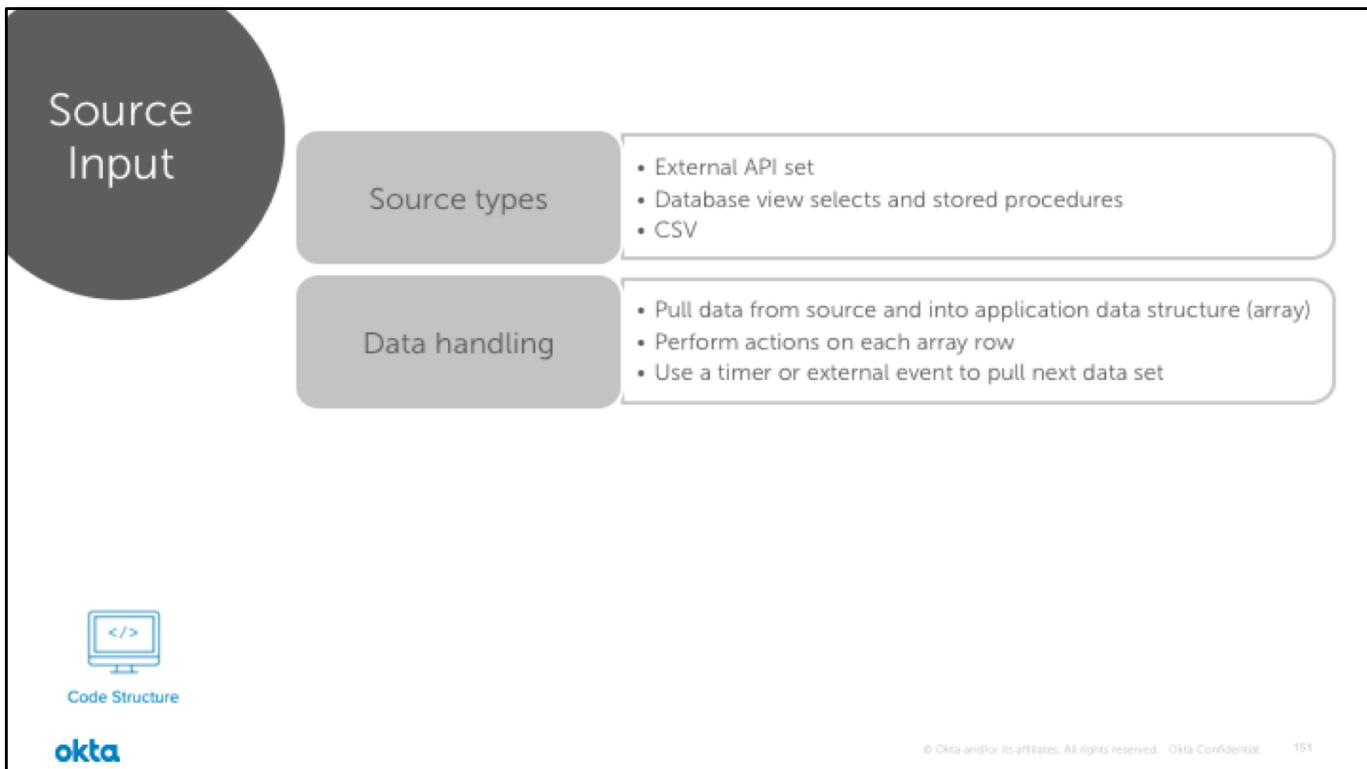
When working with Okta and an application code structure, use the Okta SDK or write modules that implement Okta APIs.

Notes:

- Any HTTP Client can execute commands and receive replies
- You should use any existing code (SDK or otherwise) to make coding easier
- Application type (Windows service or CronTab) is important

Separate your code as follows:

- Source input
- Business logic
- Target output



Additional Information

Working with the code structure requires an understanding of the source input, including types and data handling.

Which source information can be imported to Okta?

- Can the information be read, polled, queried, or called against?
- Are only the basic attributes required?

When processing each row, you must:

- Determine the retry method
- Determine how to handle halt executions
- Establish the vent trigger for the next data set process
- Show for each loop and switch statement

Business Logic

- foreach loop**
 - Is the main core of your code
 - Is where you should be exerting the most effort
 - Is where Okta resources to implement target actions are called
 - Is typically implemented as a switch statement(s)
- Implementation**
 - Check the user state in Okta before attempting to process
 - What is the source action? (create, update, or deactivate)
 - Adjust the action based on comparison between source and target
 - Instantiate the target attribute model based on source attributes
 - Call appropriate methods from library to effect target (Okta)
 - Check for errors executing API calls
 - Depending on error, wait and retry call

 Code Structure

 okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 152

Additional Information

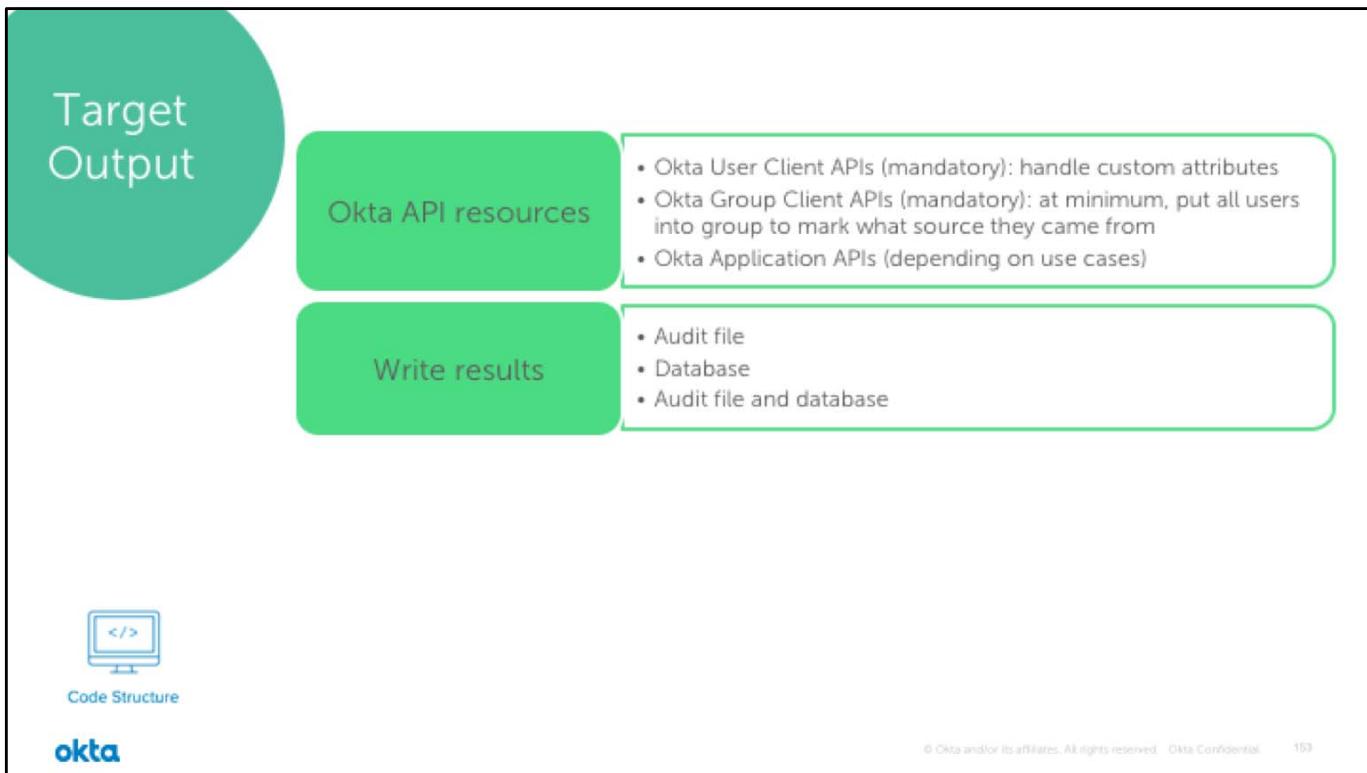
For the business logic implementation:

What is the user state in Okta? Does the user exist and could the user be a duplicate? (elastic search)

Your design document should state:

- How to adjust each action based on source and target comparisons.
- Transformations instantiated by the target attribute model based on source attributes

For error processing, if the source is a database can write into source table and retry on next pass.



Additional Information

The target output receives calls from the business logic main core.

Okta provides API templates to help you create your “as a Master” application; the user and group client APIs are mandatory components of the output, while the application API is a variable depending on the use case.

When writing results to the output, you can write to an audit file, a database, or both.

Okta APIs

- Search for user

```
GET {{url}}/api/v1/users?search=profile.customId eq "12348765" and status eq  
"ACTIVE"&limit=8
```

Response body is none or many Okta user models

- Create user with password or activation

```
POST {{url}}/api/v1/users?activate=false
```

Response body is Okta user model

- Set hardcoded password with using admin privileges

```
PUT {{url}}/api/v1/users/{{userId}}
```

```
request body: { "credentials": { "password" : { "value": "{{password}}" } } }
```



Code Structure

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential. 154

Additional Information

The slide contains examples of some Okta APIs in raw form.

You can use Postman and the developer.okta.com website to access the latest collections.

Coding Your Application

You know your design and the use cases to implement.

Choose your coding language and environment.

Use your assembled methods to pull from source and push to target.

Use your designed models for your custom and base UD attributes.

Bring it all together with coding business logic in main core



Code Structure



© Okta and/or its affiliates. All rights reserved. Okta Confidential. 155

API as a Master: When to use?

Onboarding users from:	API as a Master?
Workday, Salesforce, Successfactor, Netsuite, Ultipro AD or LDAP	
PeopleSoft or SAP HR A MySQL, Postgres, or Oracle DB table	
A SaaS solution without App as Master A heterogeneous integration landscape	

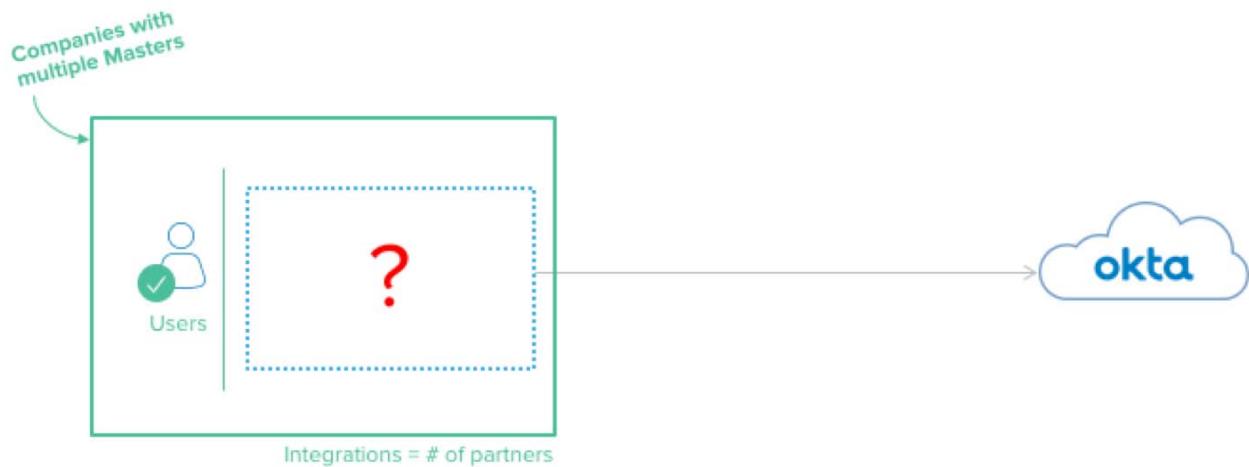
© Okta and/or its affiliates. All rights reserved. Okta Confidential. 156

Additional Information

Answers:

- **Workday, Salesforce, Successfactor, Netsuite, Ultipro:** Use OAN and App as a Master.
- **AD or LDAP:** Use LDAP and AD agents with Directory as Master
- **PeopleSoft or SAP HR:** These are examples of proprietary systems that you can integrate with API as Master.
- **A MySQL, Postgres, or Oracle DB table:** These are examples of user stores that you can integrate with API as Master.
- **A SaaS solution without App as Master:** Consider asking Okta and your SaaS provider to support App as Master in OAN before using API as a Master.
- **A heterogeneous integration landscape:** API as a Master may be an option when you have several different Masters. However, it's not the only option available.

Identifying Heterogeneous Integration Scenarios



© Okta and/or its affiliates. All rights reserved. Okta Confidential. 157

Additional Information

Some companies work with an extensive network of third party contractors. Each contractor may have a list of users that need to access systems through Okta to do work.

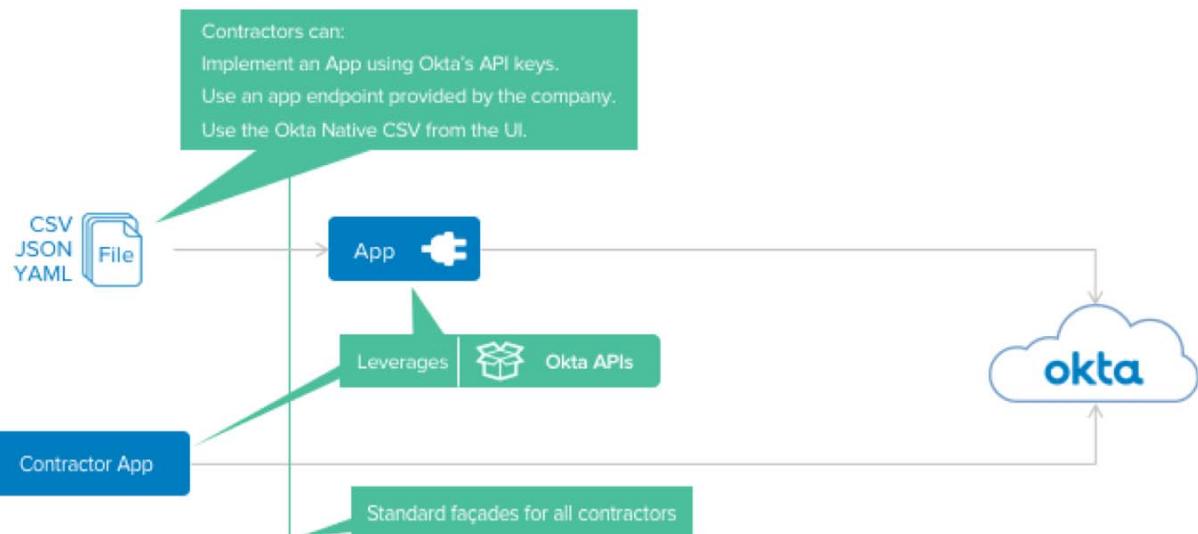
Companies with this scenario typically:

- Outsource services to multiple Help desk or Supply chain companies: Telecoms, Supermarket/Retail chains.
- Require extra workforce to support seasonality peaks: Ice cream business (summer), NFL (Super bowl, Draft).
- Depends on several contractors to deliver a product: cinema studios, car manufacture.
- Have an aggressive M&A strategy.

These companies share the same requirements:

- Support **multiple and heterogeneous masters**.
- Provide simple ways to integrate with new contractors **fast**.
- Provide a standard façade for consistent integration with reduced compatibility issues.

API as a Master: For Heterogeneous Integrations



© Okta and/or its affiliates. All rights reserved. Okta Confidential. 158

Additional Information

Depending on the requirements to support an extensive contractor network, you can define a standard façade leveraging the Okta APIs for all contractors integration.

Few examples:

- You can provide an apikey for each contractor, so they can develop their own app using Okta APIs to load users.
- You can develop a custom app to receive information from contractors in a pre-established format.

Important:

- The ideal solution depends on the customer business, security, and availability requirements.
- Besides the API Key based solutions, customers can also use the native CSV import provided with Okta Admin Console to quickly and conveniently load users.



Configure the Marketing Structure for Contractors

- Register LinkedIn and Pinterest Apps
- Create a Marketing Contractors Group and Rule



A person wearing a plaid shirt is sitting at a desk, looking at a computer screen. The screen displays a white window with a black border and the text "Lab 6-2".

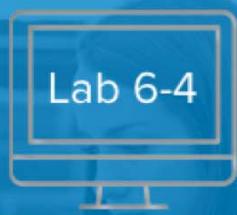
Configure API as a Master

- Get an API Token
- Configure Postman
- Explore API Requests



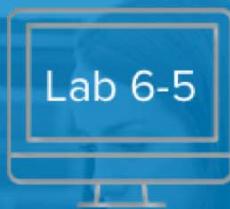
Onboard Users with API as a Master

- Onboard Users
- Verify the Results
- Access an App as a Contractor



Update Users with API as a Master

- Update Users
- Verify the Results



Deactivate and Delete Users with API as a Master

- Deactivate Users
- Verify the Results
- Delete Users

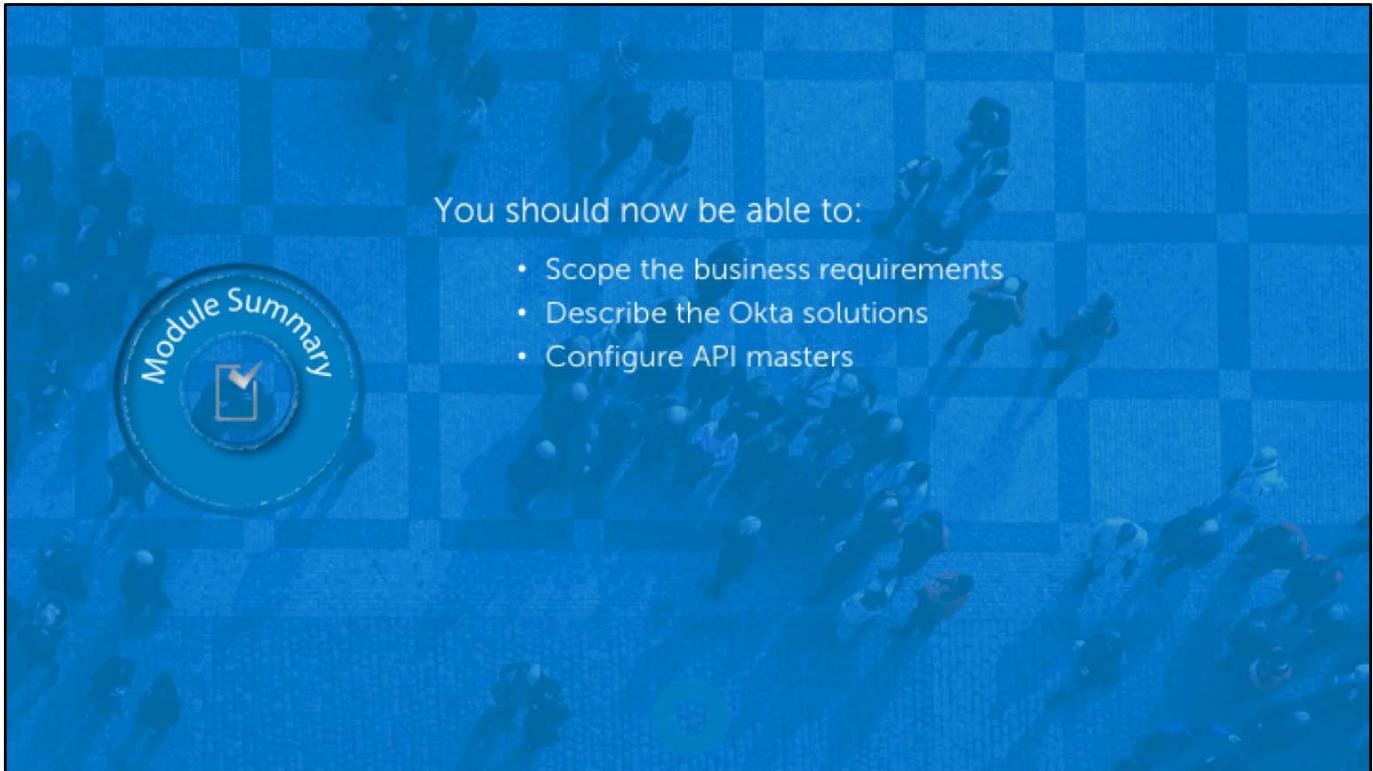
- Follow General Coding Best Practices
- Use Existing Code Libraries or Build your Own
- Separate Okta Implementation from Use Case Business Logic
- Check for Users first
- Compare Source Action to Current State
- Allow for retries of each entry
- Allow for retries of each data set
- Use Okta Groups to Identify User Source
- Audit Log



When troubleshooting API master issues:

- source code debugger
- debug log files
- check the configuration files
- maybe misconfigured for the environment
- POSTMAN
- check API syntax and JSON body





You should now be able to:

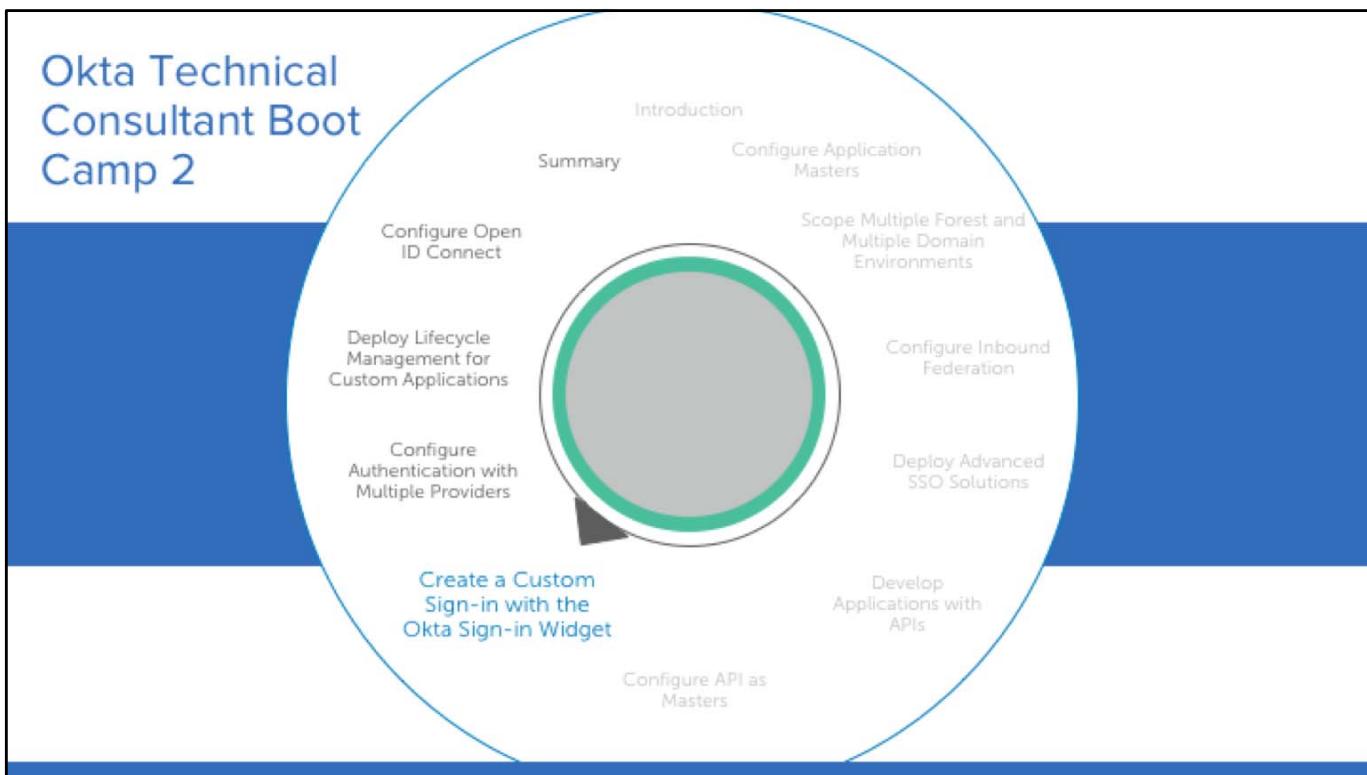
- Scope the business requirements
- Describe the Okta solutions
- Configure API masters

End of module review questions:

1. When you should consider using API as a Master?
 - a. When importing users from BambooHR.
 - b. When importing users from a proprietary system.
 - c. When importing users from AD with multiple forests.
 - d. When provisioning users to Workday, Successfactors, or other apps.

Source: Page 144
2. Why might it be a good idea to use a simplified standard for masters integration?
 - a. Using a simplified standard is the best option when you have only one master.
 - b. Using a simplified standard in scenarios with multiple partners/contractors reduces complexity and integration pain points.
 - c. Using a simplified standard is the best option when you don't have partners and keeps all your master data in Salesforce and Bamboo HR.
 - d. This is never a good idea. The best thing to do is integrate with each master using a specific technology. This include developing custom integrations, even if you have several partners or contractor companies.

Source: Page 158



Create a Custom Sign In with the Okta Sign-in Widget

Okta has created a drop-in widget, known as the Okta Sign-In Widget. It contains custom UI capabilities with the ability to sign-in users into an Okta org.

Create a Custom Sign In with the Okta Sign-in Widget



- Describe the Okta Sign-in Widget Features
- Develop Authentication Workflows
- Configure the Okta Sign-In Widget

Custom Sign-In Widget Overview

With the Okta Sign-In widget, you can easily create a custom sign in user experience.

In this module, you will start planning a sign on page by discussing and outlining business requirements.

This module consists of a lab and review question.

Custom Sign In with the Okta Sign-in Widget Overview

Enables you to...

- Understand the business needs and persona requiring access
- Define the expected user experience and workflows
- Create an effective and functional sign on page quickly

Is important because...

- You must know the business drivers and user requirements behind the web page needs to develop a meaningful and function user experience.

Additional Information

To design a sign on page using the Okta Sign-In widget, still requires an understanding of the business requirements for the page.

Okta Sign-In Widget

The JavaScript widget from Okta provides a fully featured and customizable login experience which can be used to authenticate users to any website.

Authentication

MFA

Self-service
Password Reset

Password
Expiration

Validation and
Error Handling

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The main, and most visible, solution for the Okta Sign-In Widget is validating a user using a username and password. In addition to credential validation, the Okta Sign-In Widget handles validation of password complexity requirements and displays common error messages for items such as invalid passwords and blank fields.

The Okta Sign-In Widget also comes with full support for MFA user flows. It handles enrollment and verification of factors and comes with built-in support for all factors compatible with Okta.

When it comes to password management, the widget comes with support for sending reset notifications, verifying users using a security question, and notifying and prompting users for password changes when the passwords expire.

Okta Sign On Widget Scoping

Similar to the approach for the sign-in page creation, you must answer several questions for:



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Regardless of how you create a sign on page, you must scope the business needs to help shape the design and flow of the page contents.

Sign-In Widget Flow

When working with the widget, you perform the following:

1. Start with the login-to-okta.html file.

2. Configure CORS support.

3. Test the widget.

4. Customize the style with CSS.

5. Customize features and labels with JavaScript.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

With the sign-in widget, you can easily customize aspects of the sign on page. In this task, you will use the sign-in widget for a faster sign on page configuration.

To work with the Sign-In widget, you must start at the Okta developer site.

1. Copy and save the HTML code from the http://developer.okta.com/docs/guides/okta_sign-in_widget.html#creating-an-html-file-with-the-widget-code page into a file named:
login-to-okta.html
2. With the file saved, you then need to change all instances of example.okta.com to your Okta org, for example oktaice987.okta.com.
3. With the updates made and the file saved, you then need to place the file on a web server
4. Configure CORS to allows JavaScript hosted on your websites to make an XHR to the Okta API with the Okta session cookie

HTML File

```
<head>
    <script src="https://oktastatic.oktacdn.com/assets/js/sdk/okta-signin-
widget/1.7.0/js/okta-sign-in.min.js" type="text/javascript"></script>
    <link href="https://oktastatic.oktacdn.com/assets/js/sdk/okta-signin-
widget/1.7.0/css/okta-sign-in.min.css" type="text/css" rel="stylesheet">
    <link href="https://oktastatic.oktacdn.com/assets/js/sdk/okta-signin-
widget/1.7.0/css/okta-theme.css" type="text/css" rel="stylesheet">
</head>
<body>
    <div id="okta-login-container"></div>
    <script type="text/javascript">
        var orgUrl = 'https://example.okta.com';
        var oktaSignIn = new OktaSignIn({baseUrl: orgUrl});

        oktaSignIn.renderEl(
            { el: '#okta-login-container' },
            function (res) {
                if (res.status === 'SUCCESS') { res.session.setCookieAndRedirect(orgUrl); }
            }
        );
    </script>
</body>
```

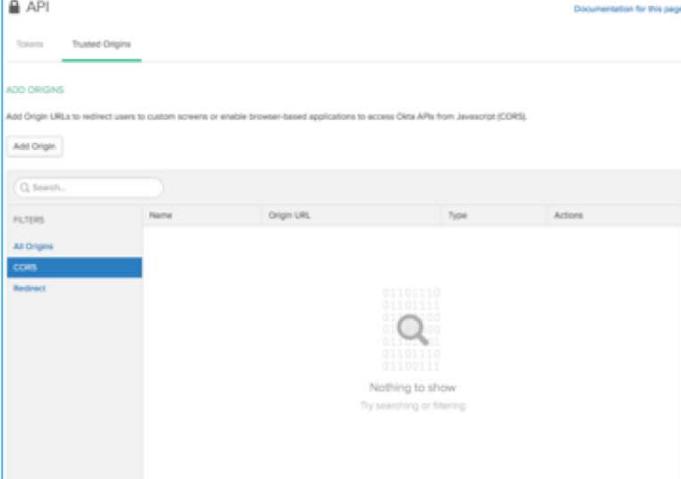
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The slide shows a basic version of the widget.

CORS

Cross-origin resource sharing (CORS) is a standard browser feature that allows JavaScript hosted on your websites to make an XMLHttpRequest (XHR) to the Okta API using the Okta session cookie.



The screenshot shows the Okta API interface with the 'Trusted Origins' tab selected. A search bar and a 'Search...' button are at the top. Below is a table with columns: Name, Origin URL, Type, and Actions. A sidebar on the left has a 'FILTERS' section with 'All Origins' and 'CORS' selected. A message at the bottom right says 'Nothing to show' and 'Try searching or filtering.'

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Because CORS is a standard browser feature that allows JavaScript hosted on your websites to make an XHR to the Okta API with the Okta session cookie, only grant access to specific websites you control and trust to access the Okta API.

Customize CSS Example

```
#okta-sign-in.main-container {  
    /* -- Fonts and Text Colors -- */ font-family:  
    "montserrat", Arial, Helvetica, sans-serif; color: #777;  
}  
  
#okta-sign-in h2, #okta-sign-in h3 {  
    /* -- Fonts and Text Colors -- */  
    font-weight: bold; color: #5e5e5e;  
}  
  
#okta-sign-in .okta-sign-in-header {  
    /* -- Fonts and Text Colors -- */ color: #5e5e5e;  
}
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The CSS file is located at: <https://ok1static.oktacdn.com/assets/js/sdk/okta-signin-widget/1.7.0/css/okta-theme.css>

Customize Features

```
var oktaSignIn = new OktaSignIn({
  baseUrl: baseUrl,
  logo: 'https://upload.wikimedia.org/wikipedia/en/thumb/7/7e/OldacmeLogo.png/200px-OldacmeLogo.png',

  features: {
    rememberMe: true,
    smsRecovery: true,
    selfServiceUnlock: true
  },

  helpLinks: {
    help: 'http://acme.example.com/custom/help/page',
    forgotPassword: 'http://acme.example.com/custom/forgot/pass/page',
    unlock: 'http://acme.example.com/custom/unlock/page',
    custom: [
      { text: 'Dehydrated Boulders Support', href: 'http://acme.example.com/support/dehydrated-boulders' },
      { text: 'Rocket Sled Questions', href: 'http://acme.example.com/questions/rocket-sled' }
    ]
  },

  // See the contents of the 'okta-theme-1.0.2.css' file for a full list of labels.
  labels: {
    'primaryauth.title': 'Acme Partner Login',
    'primaryauth.username': 'Partner ID',
    'primaryauth.username.tooltip': 'Enter your @ partner.com ID',
    'primaryauth.password': 'Password',
    'primaryauth.password.tooltip': 'Super secret password'
  }
});
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The slide shows a customized version of the widget. Within the JavaScript code, the enabled features include Remember Me, so the credentials can be saved on the page; SMS Recovery so that users can change passwords through SMS texts; and self-service unlock so that users can unlock their accounts instead of requiring administrator assistance.

The help links section contains URLs for users to get help, reset passwords, and unlock accounts.

The labels section contains the labels for the predefined fields.



Configure the Okta Sign-in Widget

- Configure the Login Widget
- Expose the Local Host to the Internet
- Enable CORS in Your Okta Org and Test
- Perform Additional Challenges with the Login Widget



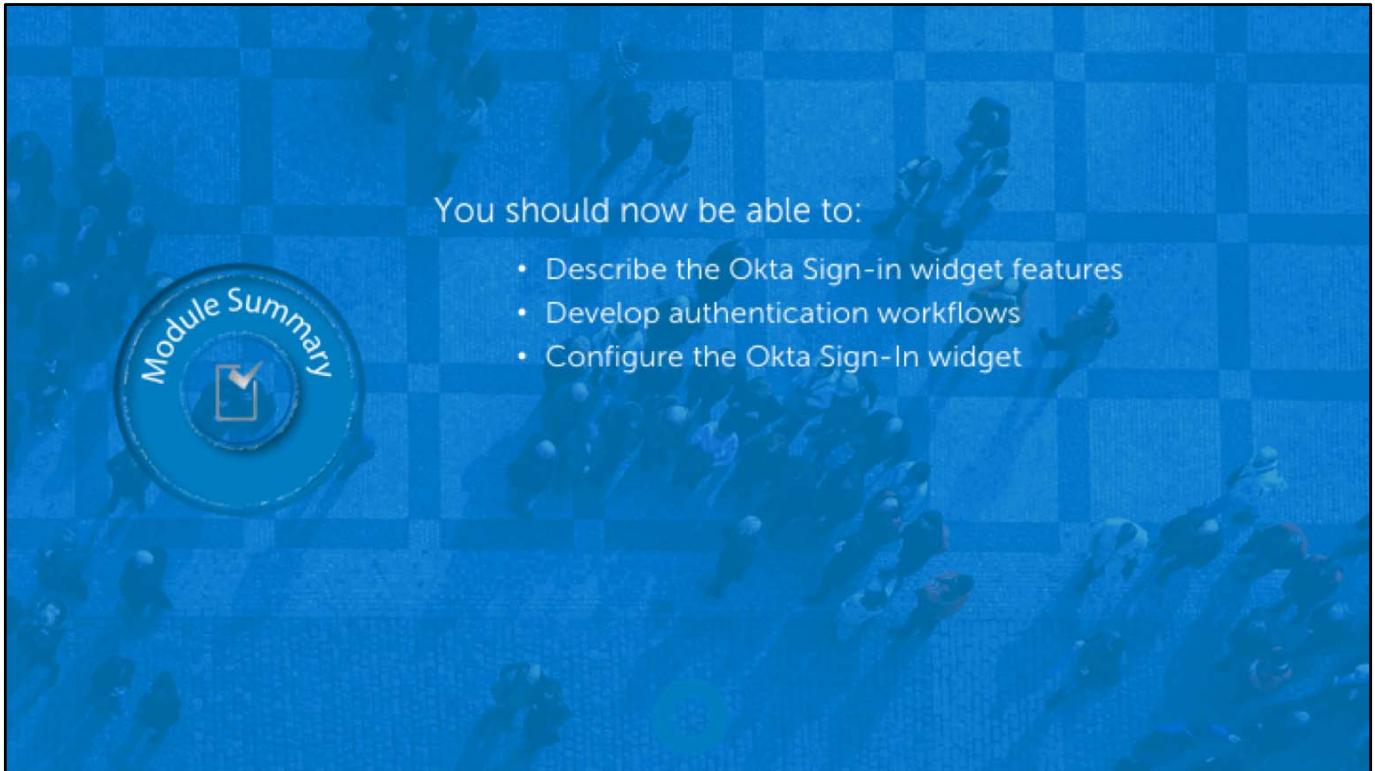
Approach the creation of a sign on page using the Okta Sign-In widget the same way you would the creation of any page by determining the:

- What
- Who
- Where
- How



When troubleshooting custom sign in issues:

- Verify CORS is configured
- baseURL is configured correctly



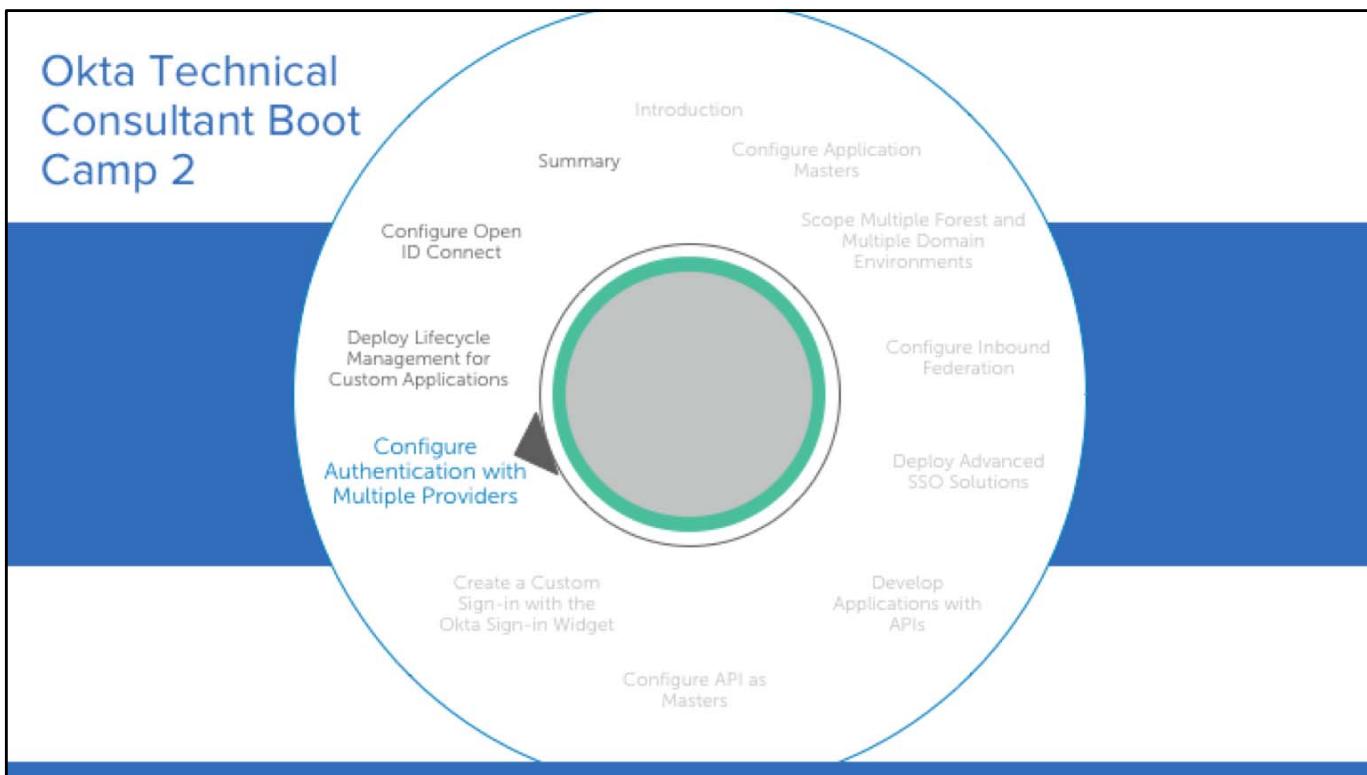
You should now be able to:

- Describe the Okta Sign-in widget features
- Develop authentication workflows
- Configure the Okta Sign-In widget

End of module review question:

1. When working with the Okta Sign-In Widget, after you get the login-to-okta.html file, what should you do next?
 - a. Test the widget
 - b. Customize the CSS
 - c. Configure CORS support
 - d. Customize features and labels in JavaScript

Source: Page 172



Configure Authentication with Multiple Providers

To help customers integrate user stores with external IdPs, you must know the configuration requirements. Before integrating an external IdP to work with Okta as the SP, you must perform some discovery and fully scope the integration.



Authentication with Multiple Providers

Scope the Business Requirements
Describe the Okta Solutions
Configure Multiple Provider Authentication

Develop Applications with APIs Overview

In this module, you will scope business requirements, share Okta solutions, and help customers configure multiple authentication providers.

This module consists of demonstrations, a lab, and review questions.

Authentication with Multiple Providers Overview

Enables you to...

- Assist Okta customers properly configure their Okta orgs with multiple authentication providers

Is important because...

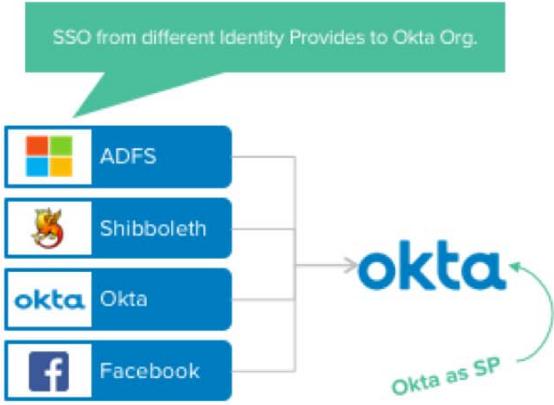
- Some customers have a diverse user base and stores with different authentication providers

Additional Information

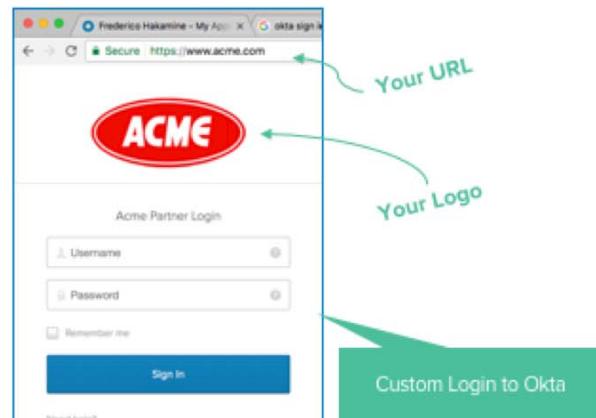
If customers are integrating applications or users currently configured to authenticate through another IdP, you need to know what and how to configure the information in the Okta org.

At this point, You know about

Inbound Federation



Sign-In Widget



[okta](#)

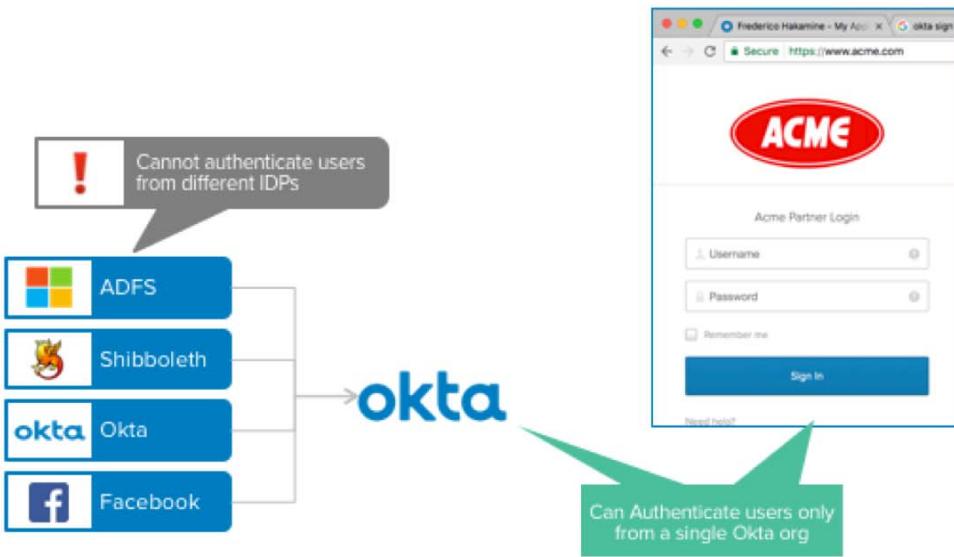
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In the previous modules, you learned about:

- **Inbound Federation:** Provides Federated SSO from other Identity Providers (IDPs) – such as ADFS or even another Okta org – to an Okta as SP, so users don't need to login twice.
- **Sign-In Widget:** A JavaScript widget from Okta that gives you a fully featured and customizable login experience which can be used to authenticate users on any web site.

Multiple IDPs with Sign-In widget



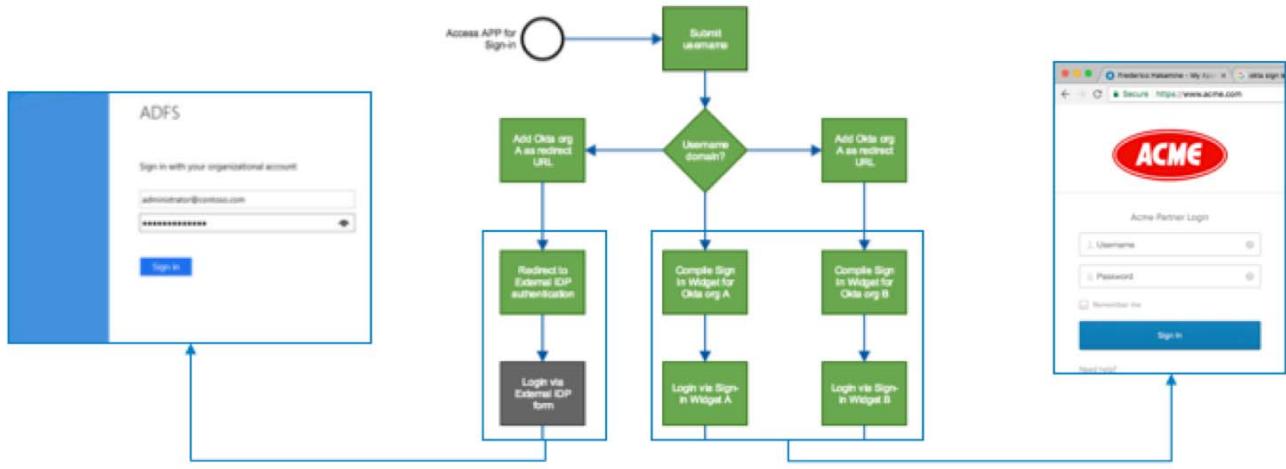
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In scenarios with multiple IDPs, the Okta Sign-In Widget out of the box is not able to:

- **Serve as UI for other vendors' authentication such as ADFS, Shibboleth, Facebook, Oracle, or IBM** because these vendors do not support authentication with the Okta Sign-In widget.
- **Serve as UI for more than one Okta org**, because the widget requires the `orgUrl` variable to be fulfilled with a unique okta org url for sign in.

IDP Discovery: Overview



A process that figures out in what IDP a user should authenticate, and route the user to the correct login process.

okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

To attend a multiple IdP scenario, you must **discover in what IDP** users should be authenticated and route users to the appropriate authentication with a redirect back to the Okta org SP.

If a user is:

- **from a different vendor**, you must redirect them to their appropriate sign in page and signal that the user must be redirect to your Okta org SP after authentication.
- **from a different Okta org**, you must update the sign-in widget `orgUrl` variable with the correct Okta org IDP and update the `redirectUrl` with a link to return to the Okta org SP.

IDP Discovery: Example of Determining the IdP

The appropriate IDP can be determined with information provided users before authentication.

As an example, if user provides its email prior authentication:

IF the user email ends with:

@okta.com => compile sign in widget for oktacentral.oktapreview.com
@oktaice.com => compile sign in widget for oktaice.oktapreview.com
@partnera.com => redirect to ADFS IDP log in: intranet.partnera.com
@partnerb.com => redirect to Oracle IDP log in: oam.partnera.com



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

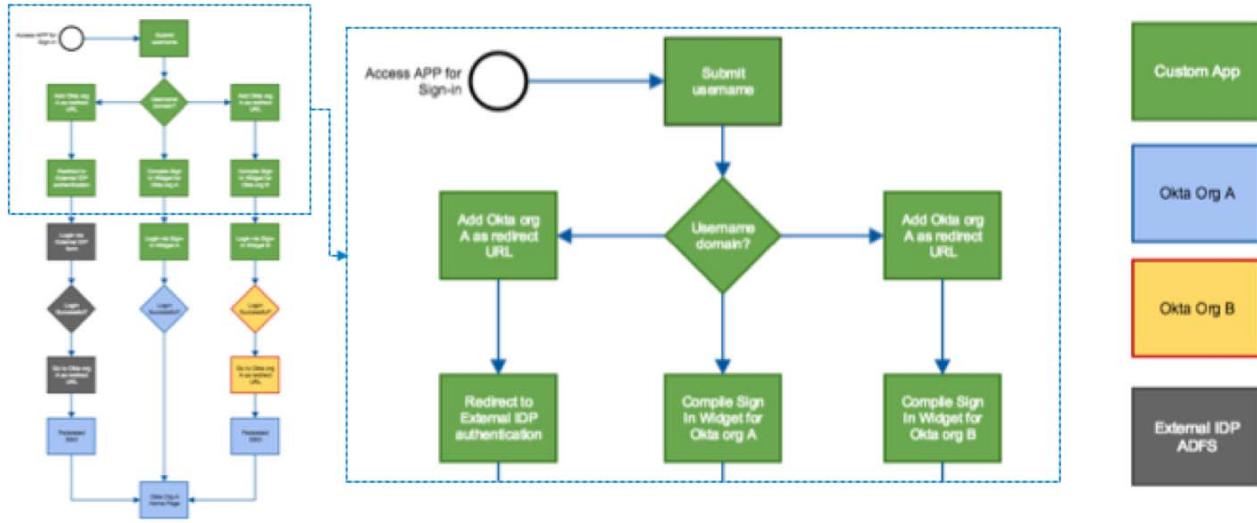
This slide shows an example on how the IDP can be discovered by gathering the user email prior authentication.

The IDP Discovery process depends on how your user data is organized and the information your user possess during the authentication process.

Additional examples of gathering information for IDP discovery:

- Users select their company in a form.
- Network routing and proxies.
- Sticky cookies.

Sign In Process with IDP Discovery



okta

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

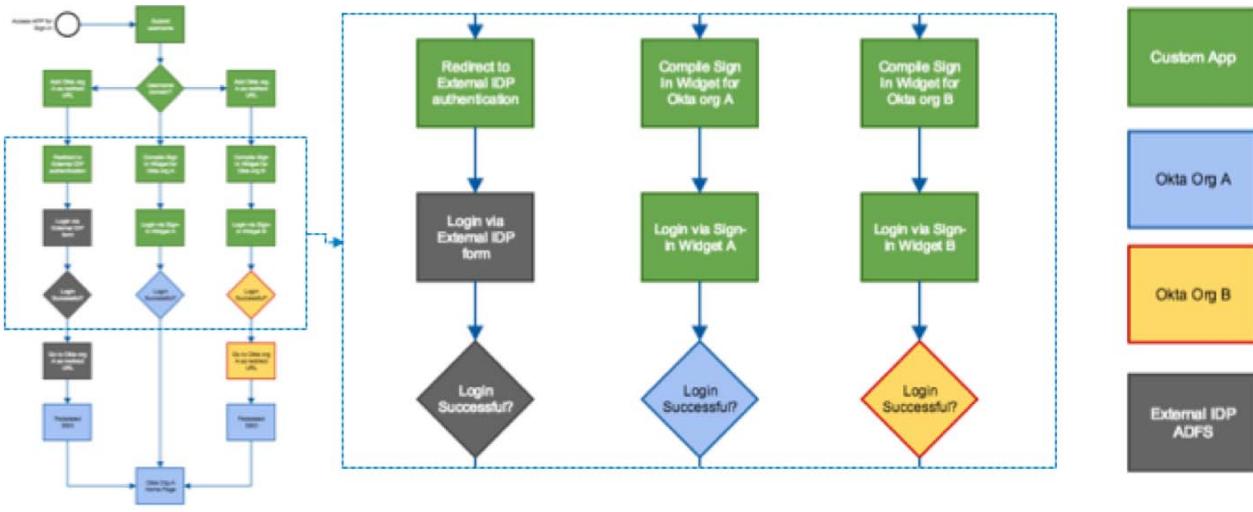
This and the next few slides cover the entire sign in process with IDP Discovery. The colors identify what component is in charge of processing each step.

1. The process starts when a user access a page for sign-in.
2. A Custom app – in charge of performing the IDP discovery – presents a form to collect the username (email format).
3. The user submits his/her username for the custom app that route the user to one of 3 options:
 - Sign in an external vendor IDP (**left**)
 - Sign in Okta SP directly (**center**), or
 - Sign in another Okta org as IDP (**right**)

When the sign in is in Okta, the user is redirected to the sign-in widget with a specific `orgUrl`.

When the sign in is in an external vendor, the user should be redirected to the vendor log in page.

Sign In Process with IDP Discovery



Additional Information

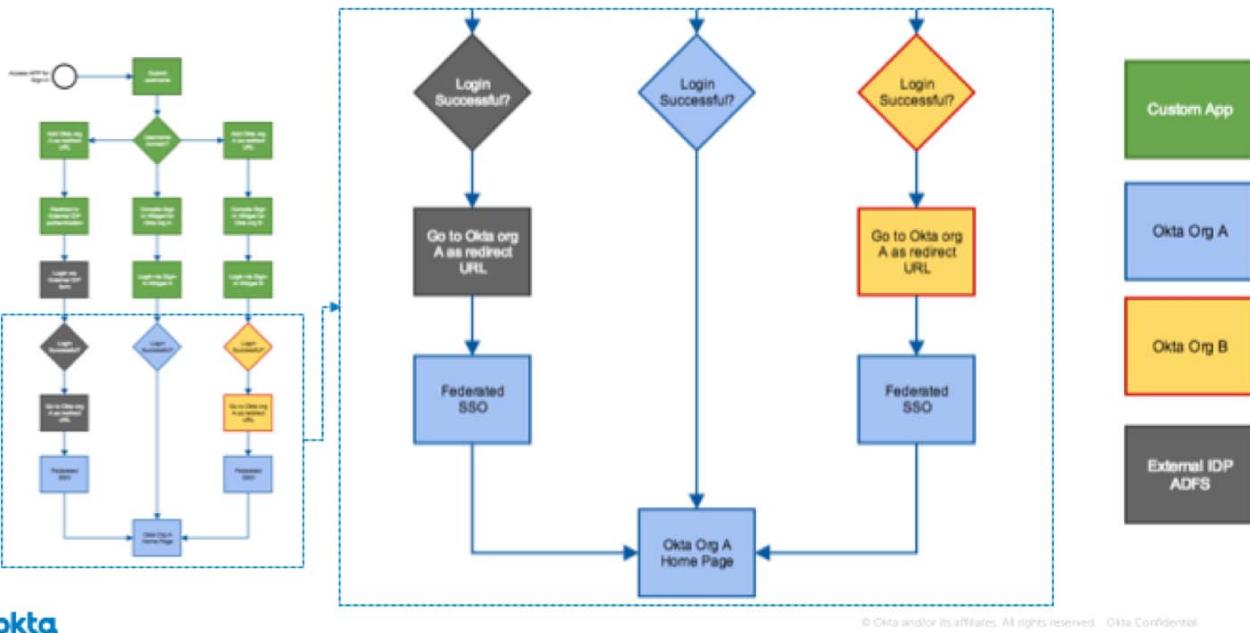
This slide shows how the authentication is performed in each IDP after the discovery process. Observe that:

- All authentication options require a log in validation in a specific IDP – External vendor IDP (grey), Okta org SP (blue), and Okta org IDP (yellow).
- The Custom App is in charge of presenting the sign-in widget when authentication happens in Okta.

Process:

4. The custom app either redirects the user for authentication in an external vendor page or prepares the sign-in widget for Okta's authentication.
5. The user submits his/her credentials.
6. The IDP – External vendor IDP (grey), Okta org SP (blue), and Okta org IDP (yellow) – authenticates the user.

Sign In Process with IDP Discovery



Additional Information

This slide covers the end of the sign-in process with IDP Discovery. At this point, the user is authenticated in one of the IDPs: External vendor IDP (grey), Okta org SP (blue), or Okta org IDP (yellow).

- If the user is authenticated in Okta org SP (blue), he goes straight to the home page.
- If the user is authenticated in another IDP – External vendor IDP (grey) or Okta org IDP (yellow), this IDP redirects the user to Okta org SP (blue) using the url for the SP-initiated SSO.

Discovery IDP: Configuration steps

1. Integrate external IDPs in the Okta SP org.
2. For each Okta org (IDP or SP), enable CORS for the Sign-In widget.
3. For the Okta orgs acting as IDP, enable redirect for the Okta SP org (this is required to avoid authorization errors).
4. Configure the IDP discovery app with the proper information.



© Okta and/or its affiliates. All rights reserved. Okta Confidential.



Perform IdP Discovery

- Walk through code implementation
- Configure and test IdP Discovery



Configure IdP Discovery

- Open and Launch the Code Sample
- Configure the Okta Orgs
- Configure the Code Sampel



Test the IdP Discovery

- Test Access as an Okta Central User
- Test Access as an Okta Ice User

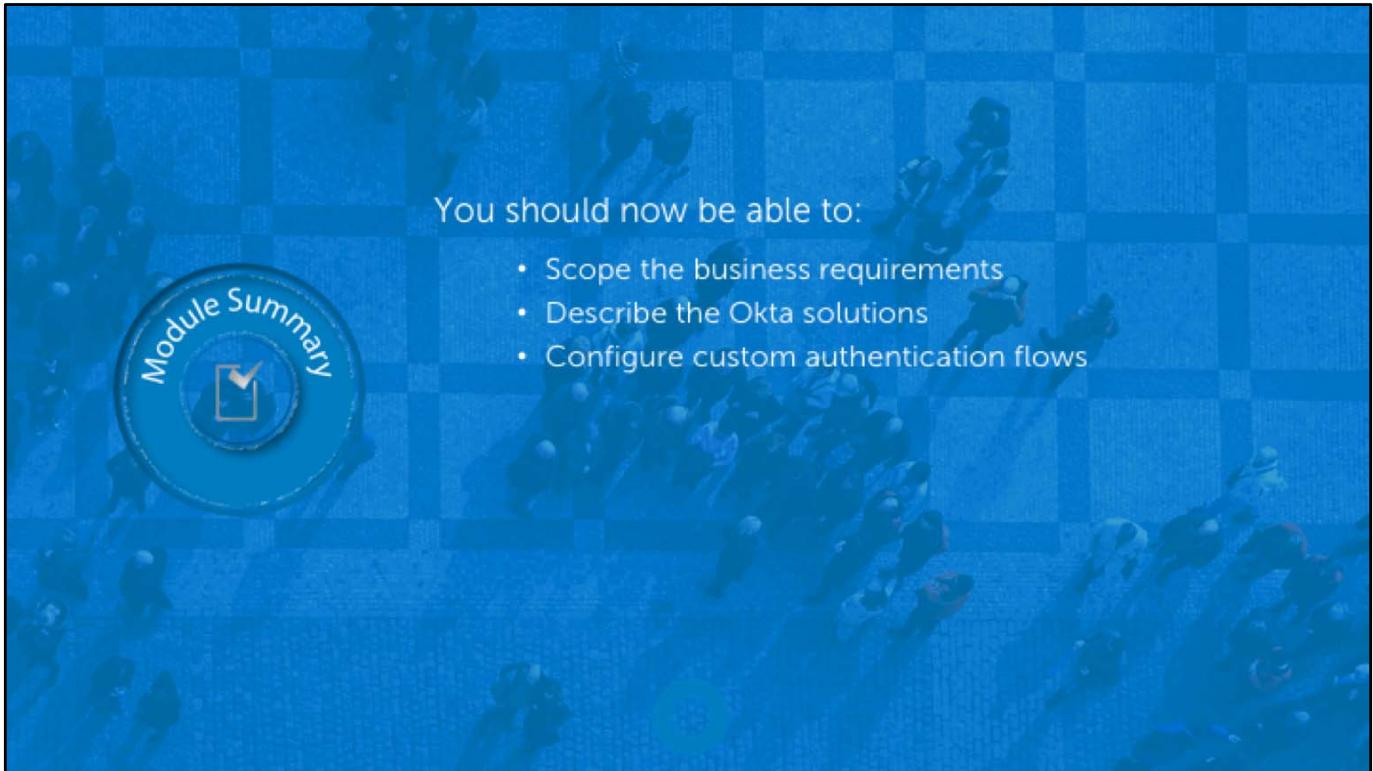


- Follow general coding best practices
- Use existing code libraries or build your own
- Use the Okta Sign-in Widget if possible
- Use Okta groups to IdP source
- Know the configuration files
- Configure and consult audit and debug logs

When troubleshooting Okta as the SP issues:

- Check the following:
 - Source code debugger
 - debug log files
 - Configuration files; maybe misconfigured for the environment
- Verify the user can authenticate into 3rd-party IdP independent of integration with Okta
- Use browser to hit and check if ACS redirects properly





You should now be able to:

- Scope the business requirements
- Describe the Okta solutions
- Configure custom authentication flows

End of module review questions:

1. When configuring a 3rd-party IdP with SAML, how is authentication delegated to the IdP?

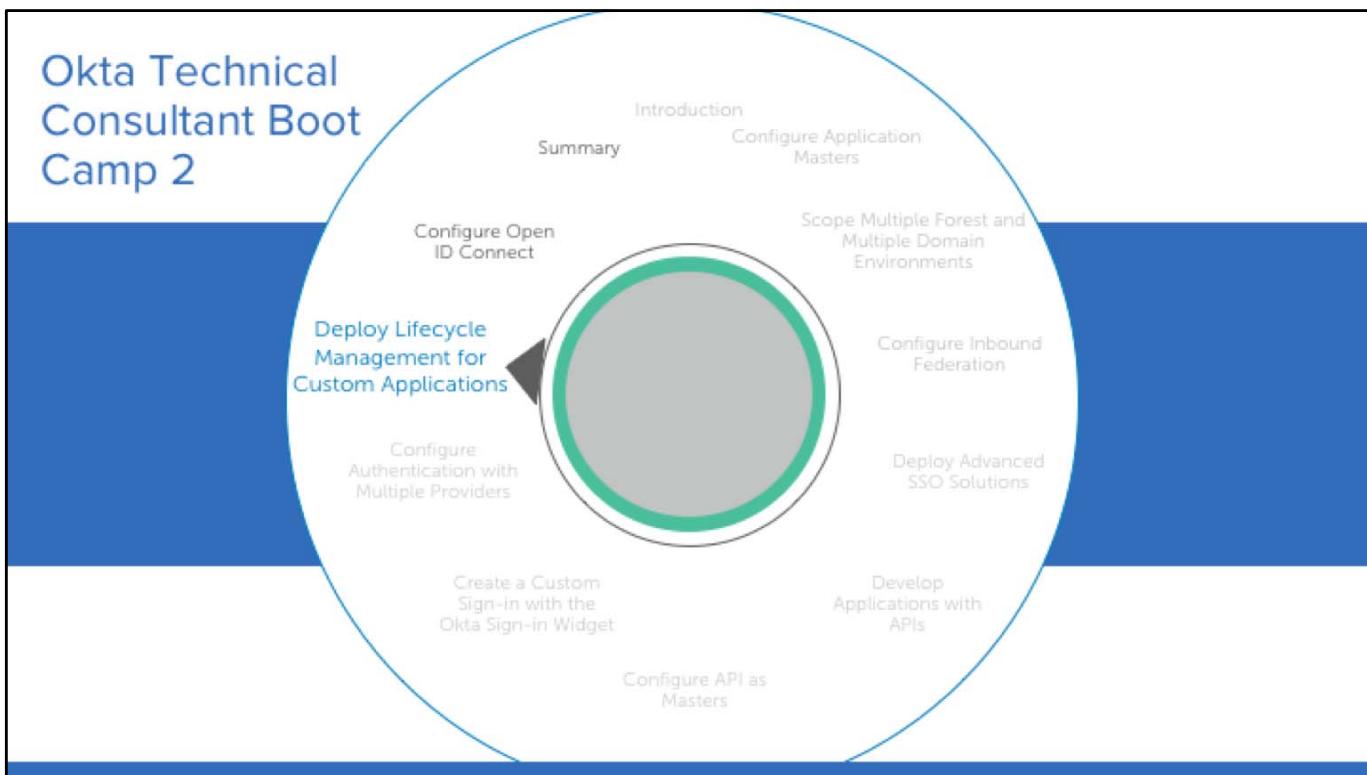
- a. It is the IdP Issuer URI.
- b. It is pushed through the ACS URL.
- c. It is located in the IdP signature certificate.
- d. You configure it through Universal Directory attribute mappings.

Source: Page 186

2. What must be configured in Okta to support external IdP authentication for an application?

- a. A SAML certificate
- b. The IdP configured using the AIW and set as a master
- c. The IdP within Okta and attribute mappings in Universal Directory
- d. The IdP set as a master and attribute mappings in Universal Directory

Source: Page 191



Deploy Lifecycle Management for Custom Applications

With Okta implement, you can provide SSO to and manage the account lifecycle for custom applications.

With UD, you can also perform advanced attribute-level mappings and transformation for customized application workflows.



Deploy Lifecycle Management for Custom Applications

- Describe Lifecycle Management
- Deploy Native SCIM
- Deploy On Premise Provisioning

Lifecycle Management for Custom Applications Overview

In this module, you use Okta's lifecycle management features to manage user accounts in custom applications.

This module consists of six labs and review questions.

Deploy Lifecycle Management for Custom Applications

Enables you to...

- Help Okta customers integrate homegrown apps with Okta for automated lifecycle management

Is important because...

- Customers might have proprietary applications with user provisioning requirements for integration with Okta.

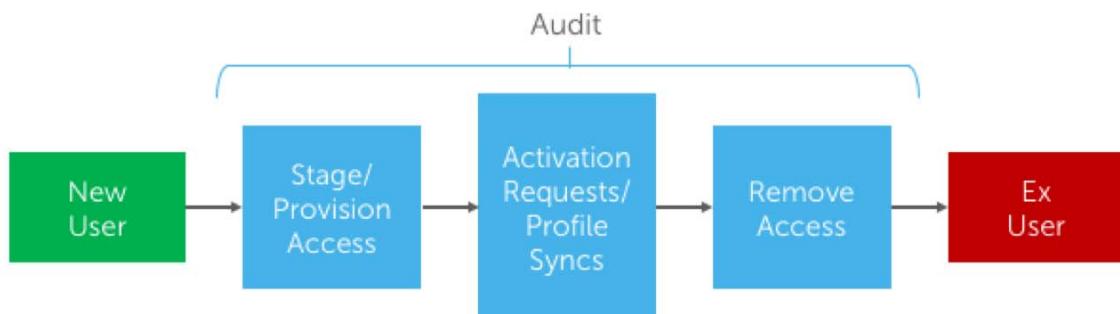
Additional Information

Okta can provide SSO and manage the account lifecycle in custom applications.

With lifecycle management, you can manage accounts in custom applications directly from Okta.

You can also take advantage of advanced Okta features related to Lifecycle Management, such as the UD's attribute mappings/transformations and approval workflows.

Okta Lifecycle Management: Overview



- Accelerate onboarding
- Reduce costs
- Meet auditing requirements
- Increase adoption
- Real-time off-boarding
- Greater data quality

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The user lifecycle is more than the provisioning or creation of a user in endpoint systems. As a part of ongoing updates, there are requests to activate access to more systems and updates for user credentials or profile information.

Finally, when users leave the company, access to systems should also be removed as quickly as possible. Often, the various account statuses and changes must be audited for compliance or verification of appropriate access.

Okta Lifecycle Management: Benefits

30 Minutes
saved on every application provisioning request

30 Minutes
saved on determining and configuring groups and entitlements

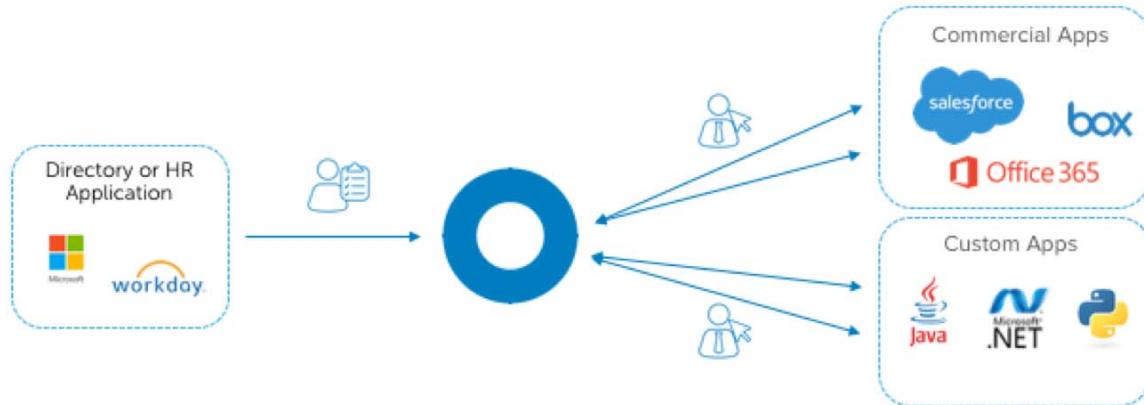
\$20
per user saved in preparing for audits each year

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Companies have several specialized systems that employees, contractors, partners, and customers use. Automating lifecycle management tasks can save a significant amount of time and money. Above are some statistics that Okta has collected from our customers and the benefits they have seen implementing Okta Lifecycle Management.

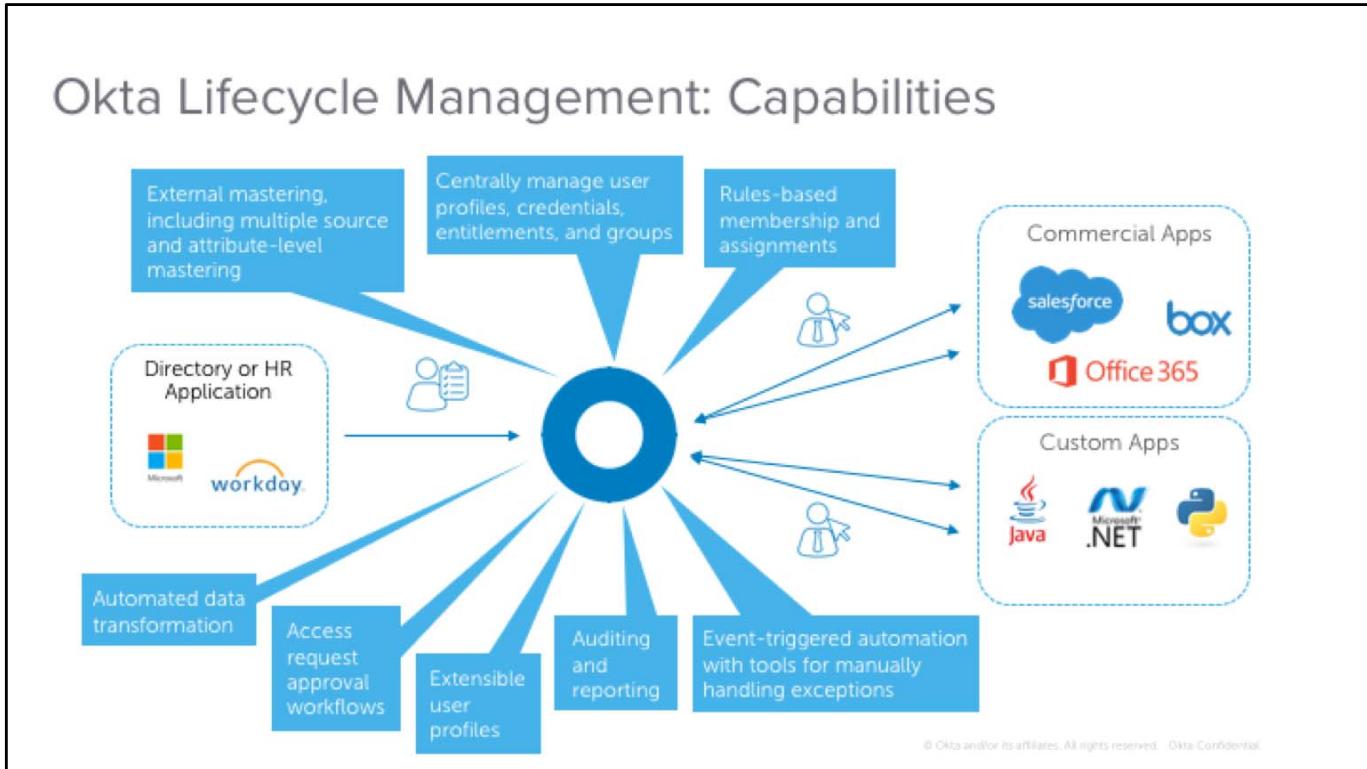
Okta Lifecycle Management: Architecture



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

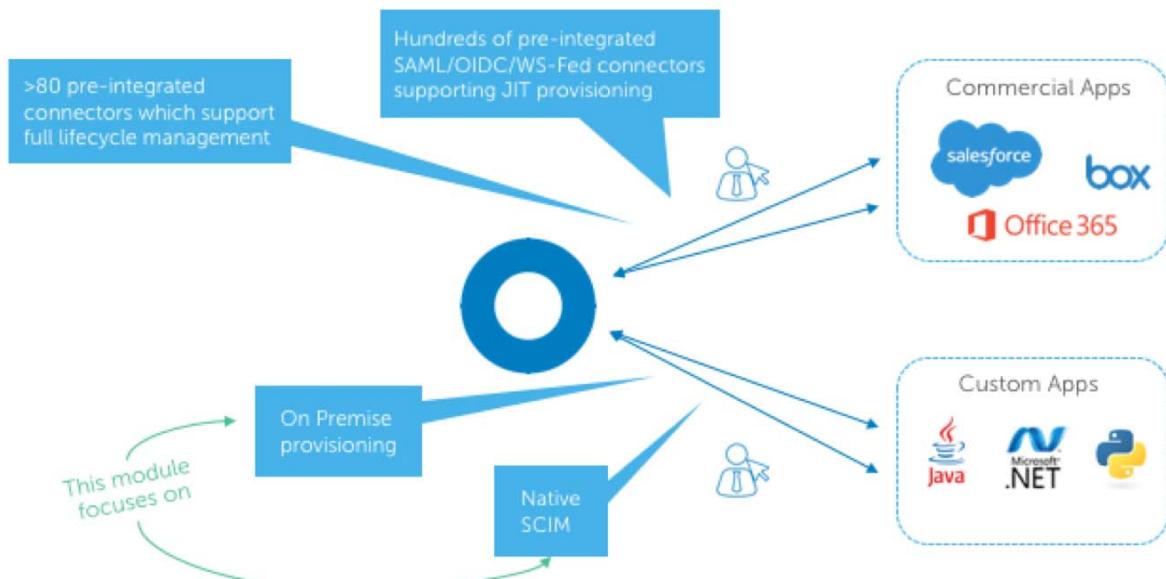
Directories and applications are a part of the Okta lifecycle management story. In addition, Okta can integrate with both commercial and custom applications used by enterprises.



Additional Information

- External mastering:** Okta supports deep integration capabilities to directories such as Active Directory or LDAP, and HR applications such as Workday and Bamboo. Not only can external directories and applications be used as masters, but multiple systems can be used as masters, down to the attribute level.
- Centralized management:** Okta is a single tool to manage all integrations and processes, including users, groups, entitlements, and credentials.
- Rules-based authorization:** Okta manages access to applications through group memberships, while also supporting direct resource assignment.
- Extensible user profiles:** Through UD, Okta supports custom user attributes to track important company information for any need, business, IT, application, or otherwise.
- Auditing and reporting tools:** Okta has robust reporting tools and API access to events in the system logs.
- Event-triggered automation plus tools for exceptions:** Okta automates these processes and notifies administrators when tasks require manual work because of rule exceptions.
- Access request workflow engine:** Within Okta, administrators can define processes enabling end users to request access to applications and decision makers to gate application access, eliminating the need for IT involvement.
- Data transformation:** Okta has the Expression Language, which can be used to perform automated transformation of user attributes where required on endpoint applications.

Okta Lifecycle Management: Connectivity



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

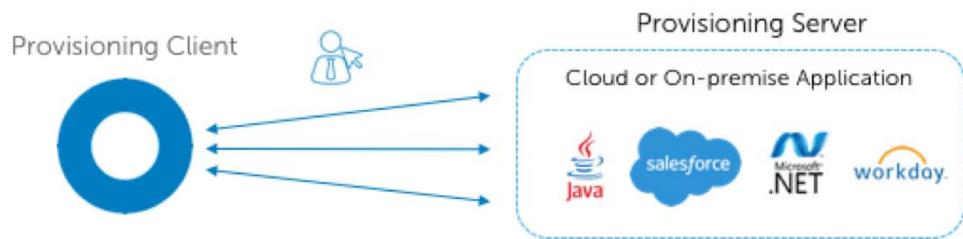
Additional Information

Okta lifecycle management can be connected with:

- **Commercial cloud apps:** Okta has many pre-integrated application connectors with several connectors supporting full lifecycle management for the most popular applications. Okta also has an extensive library of SAML, OIDC, and WS-Fed connectors for SSO, which can be used for JIT provisioning.
- **Custom apps:** For company-created applications requiring lifecycle management, Okta supports:
 - On Premise Provisioning (OPP)
 - Native SCIM

System for Cross-domain Identity Management (SCIM)

- IETF standard for provisioning
 - RESTful/JSON
 - Simple CRUD operations
- Current Version: 2.0
- Okta supports 1.1 and 2.0
- <http://www.simplecloud.info>

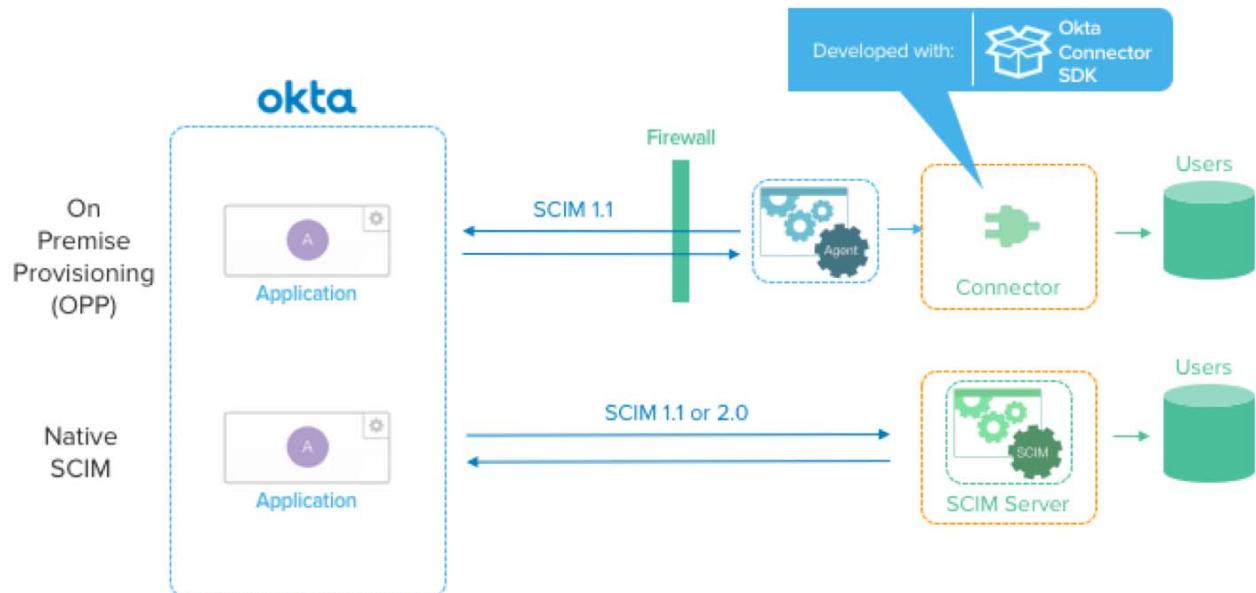


© Okta and/or its affiliates. All rights reserved. Okta Confidential

Additional Information

SCIM is an emerging standard for provisioning and de-provisioning. Okta acts as the provisioning client in this relationship. Application providers and customers can automate these provisioning operations in their endpoints by implementing SCIM servers. These integrations can be used by SCIM clients other than Okta.

Provisioning Options



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In the next few slides, you learn more about the On Premise Provisioning (OPP).

On Premise Provisioning vs. Native SCIM

	On Premise Provisioning	Native SCIM
Audience	Customers	ISVs/Customers
Advantages	<p>Is designed for applications behind a firewall</p> <p>Supports groups</p>	<p>Is re-usable across identity managers</p> <p>Is written in any language</p>
Disadvantages	<p>Requires Java</p> <p>Requires the installation and maintenance of an agent</p>	<p>Requires a public SCIM interface</p> <p>Does not support groups</p>

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

JIT Compared to Lifecycle Management

User Actions	SAML JIT	OPP/Native SCIM
Create on Access	X	
Create on Assignment		X
Update on Access	X	
Update on Change		X
Read		X
Deactivate		X
Password Sync		X

Additional Information

SAML JIT does overlap with SCIM, but SCIM has many additional features. One not mentioned here is that SCIM is a much more simple standard based on REST.



Group Discussion

Which applications are behind firewalls?

How are users created, managed, and removed from applications?

Which business drivers, such as departments or policies, determine how users are granted access to company applications?

Key SCIM Schemas

- **User:** A type of resource to be provisioned/de-provisioned
- **Group:** A type of resource to be provisioned/de-provisioned
 - Frame support required, but not yet implemented in Okta
- **ListResponse:** An array of resources
- **Error:** More detailed message for HTTP 400 Bad Request responses
- **PatchOp:** For deactivations/reactivations and password syncs
 - <http://jsonpatch.com>

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

While SCIM defines groups, Okta does not implement groups.

User schema: urn:ietf:params:scim:schemas:core:2.0:User

Group schema: urn:ietf:params:scim:schemas:core:2.0:Group

By "frame support" for groups, as you will see later, Okta does not yet read or write with Groups. But it does a Group query that can simply return an empty list of Groups. The SCIM Server must implement this empty responder.

ListResponse: urn:ietf:params:scim:api:messages:2.0>ListResponse

Error: urn:ietf:params:scim:api:messages:2.0>Error

PatchOp: urn:ietf:params:scim:api:messages:2.0:PatchOp



Key SCIM Schemas

User (resource): typically sent in write operations

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"],
  "id": "2819c223-7f76-453a-919d-4138619804646",
  "externalId": "00u6fbmuae800zhly0h7",
  "meta": {
    ...
  },
  "active": true,
  "name": { ... },
  "userName": "bjensen",
  "phoneNumbers": [ { ... } ],
  "emails": [ { ... } ]
}
```

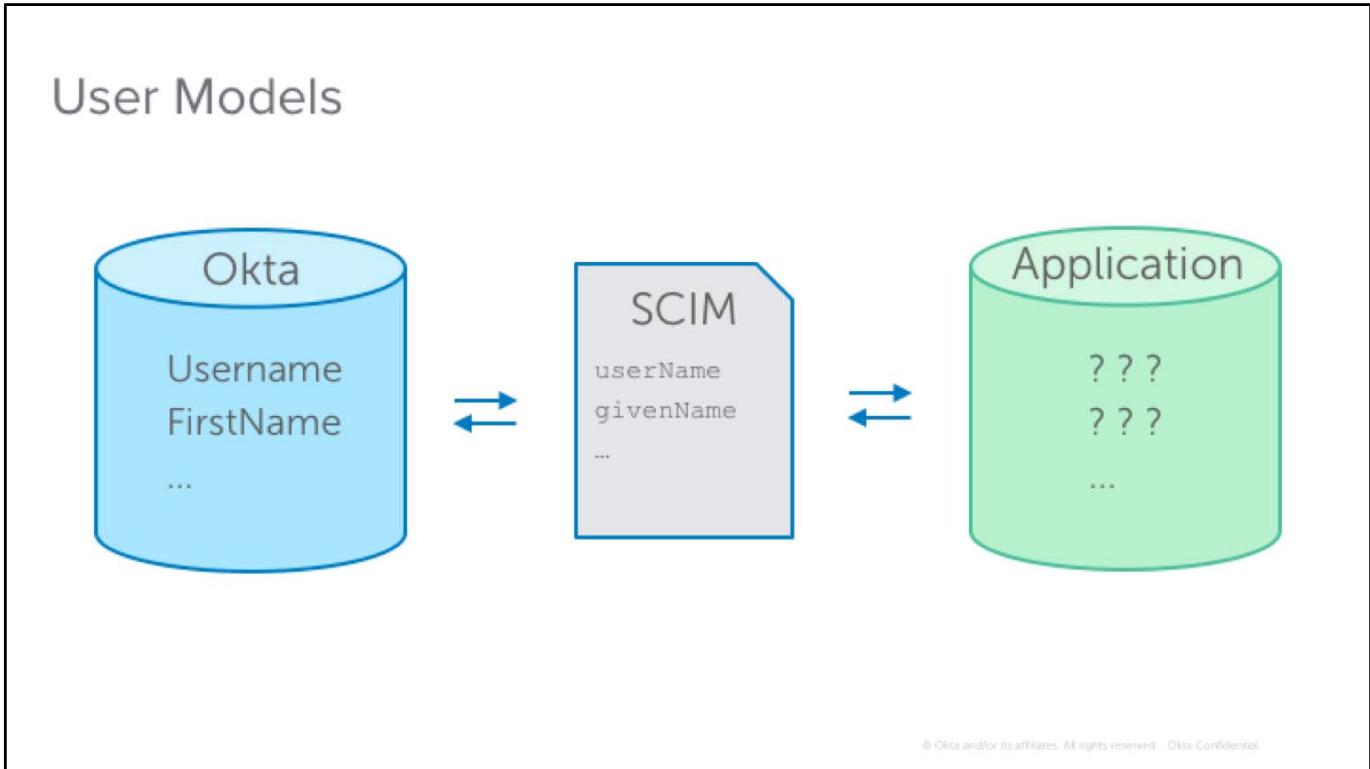
ListResponse: returned from read operations

```
{
  "Resources": [
    {
      "active": true,
      "id": "001e7386-86cb-4650-88f7-0f2b6e6ca5bf",
      "meta": {
        "location": "http://v2.08eb6bb44.ngrok.io/scim/v2/Users/001e7386-86cb-4650-88f7-0f2b6e6ca5bf",
        "resourceType": "User"
      },
      "name": {
        "familyName": "Gupta",
        "givenName": "Sehail",
        "middleName": null
      },
      "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:User"
      ],
      "userName": "sehail.gupta@example.com"
    }
  ],
  "itemsPerPage": 100,
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "startIndex": 0,
  "totalResults": 1
}
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

While there are two types of resources in SCIM, the slide shows a high level view of the user resource.



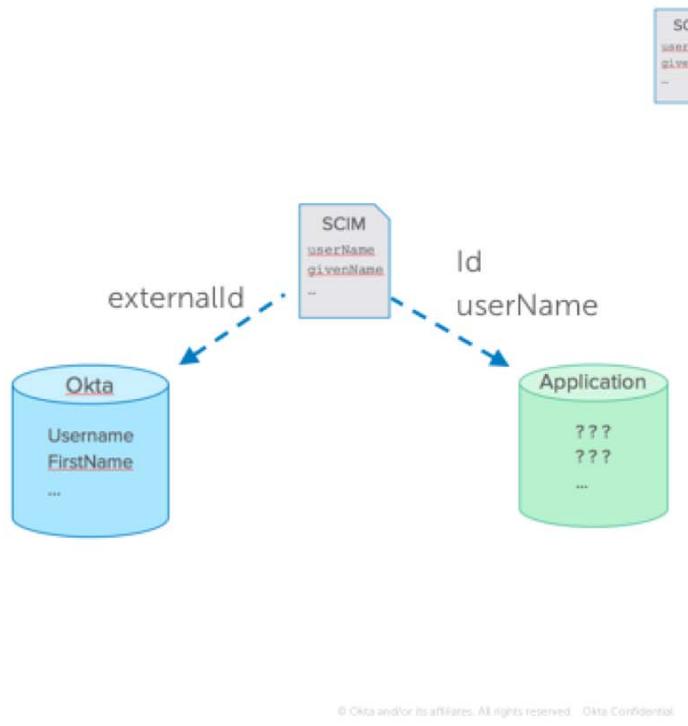
Additional Information

There are three different user models with SCIM.

1. The user as defined by Okta.
2. The user as defined by the SCIM standard.
3. The user as defined by the endpoint application.

Identifiers

- IDs
 - Immutable
 - Unique key per app
- Usernames
 - User facing
 - Unique per app



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In SCIM, IDs are reserved for system use to uniquely locate users and are immutable.

SCIM Keys



Key	Description	Example
<code>id</code>	<ul style="list-style-type: none"> A immutable key for identifying the user in the provisioning server Generated by the provisioning server Globally unique Used to generate a URI for each resource 	<code>"id": "2819c223-7f76-453a-919d-413861904646"</code>
<code>externalId</code>	<ul style="list-style-type: none"> An immutable internal identifier for the user in provisioning client (Okta) Sourced from provisioning client Unique in the provisioning client 	<code>"externalId": "00u6fbmuee800zh1y0h7"</code>
<code>userName</code>	<ul style="list-style-type: none"> Login credential of the user in the endpoint application May be set by provisioning client (Okta) May be sourced from provisioning server via Import Unique per tenant in endpoint application 	<code>"userName": "bjensen"</code>

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

ID refers to the identifier in the endpoint application. After provisioning on the endpoint, Okta stores this value internally for later updates and lookups.

The ID must be used by the SCIM server to create a location, which is a URI that fetches the details for a specific user.

The externalId refers to the identifier in Okta.

The `userName` is the login credential for the user, which can be changed. This can be created from the Okta username, or you can use Okta Expression Language to automate generating a username according to a formula.

Okta SCIM Client

- Configure
 - Connection settings
 - Authorization credentials
- Enable provisioning features
 - Import Users
 - Create Users
 - Update User Attributes
 - Deactivate Users
 - Sync Passwords
- Runtime
 - Detect Actionable Events
 - Send SCIM-compliant messages



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The slide contains a list of the integration points for Okta as a SCIM client.

SCIM Client Connection Settings

- Base URL:
 - Internet accessible URL for SCIM Server
 - Recommended root: /scim/v2
 - Must support HTTPS
- Authorization options:
 - OAuth 2.0 Bearer Token
 - Custom HTTP Headers
 - Basic Authentication



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

These are the three options for authorization supported by Okta. It is recommended that you use bearer tokens. Basic authentication is the least recommended option.

OAuth 2.0 Bearer Token

- Okta passes a bearer token in all HTTPS requests.

API CREDENTIALS

Enter your SCIM 2.0 Test App (OAuth Bearer Token) credentials to enable user Import and provisioning features.

SCIM 2.0 Base Url	<input type="text" value="https://b8a60b44.ngrok.io-Bf70n4ag19jw.runscope.net/scim/v2"/>
OAuth Bearer Token	<input type="password" value="*****"/>
<input type="button" value="Test API Credentials"/>	

```
GET /scim/v2/Users HTTP/1.1
Host: server.example.com
Authorization: Bearer token12345
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The OAuth 2.0 Bearer Token option passes a token in the HTTP header. Bearer tokens have the advantage of being revocable and not associated with a specific user.

Basic Authentication

- Username/Password concatenated with a colon and base64 encoded

API CREDENTIALS

Enter your SCIM 2.0 Test App (Basic Auth) credentials to enable user import and provisioning features.

SCIM 2.0 Base Url	<input type="text" value="https://abc625f1.ngrok.io/scim/v2"/>
Username	<input type="text" value="testuser"/>
Password	<input type="password" value="*****"/>
<input type="button" value="Test API Credentials"/>	

```
GET /scim/v2/Users HTTP/1.1
Host: server.example.com
Authorization: Basic dGVzdHVzZXI6VGVzdGluZzEyMw==
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Custom Header Authorization

- Simple token passed in the Authorization header

API CREDENTIALS

Enter your SCIM 2.0 Template (Header Auth) credentials to enable user import and provisioning features.

Base URL	<input type="text" value="https://d2579d6d-
ngrok-io-vmqbfqehekl.runscope.net/scim/v2"/>
API Token	<input type="text" value="token12345"/>
<input type="button" value="Test API Credentials"/>	

```
GET /scim/v2/Users HTTP/1.1
Host: server.example.com
Authorization: token12345
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Okta-supported SCIM Operations

- Create User (RFC 7644, Sec. 3.3)
- Update User Attributes (RFC 7644, Sec. 3.5.1)
- Deactivate User (RFC 7644, Sec. 3.5.2)
- Sync Password (RFC 7644, Sec. 3.5.2)
- Retrieve Known Resource (RFC 7644, Sec. 3.4.1)
- Import Users by Query (RFC 7644, Sec. 3.4.2)

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Links to the RFC specifications.

- Create User: <https://tools.ietf.org/html/rfc7644#section-3.3>
- Update User Attributes: <https://tools.ietf.org/html/rfc7644#section-3.5.1>
- Deactivate User: <https://tools.ietf.org/html/rfc7644#section-3.5.2>
- Sync Password: <https://tools.ietf.org/html/rfc7644#section-3.5.2>
- Retrieve Known Resource: <https://tools.ietf.org/html/rfc7644#section-3.4.1>
- Import Users: <https://tools.ietf.org/html/rfc7644#section-3.4.2>

Not Yet Supported by Okta

- Groups
- SCIM-as-a-master
- Delete Users
- Bulk write operations
- Schema discovery
- /Me
- /ServiceProviderConfig
- Query with POST
- Query Filter with meta.lastModified

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

SCIM operations not yet implemented by Okta.

http://developer.okta.com/docs/guides/scim_guidance.html#scim-features-not-implemented-by-okta

Query for Users Operation: Filters

Filters:

- `userName eq`
- `id eq`
- `emails eq`
- `externalId eq`

1.

Request from Okta

- `GET /scim/v2/Users?`
`filter=userName+eq+"sohail@example.com"`

Response from SCIM Server

- `HTTP 200`
- Returns **ListResponse** of User resources

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The example should be URL encoded, but is not for the purpose of readability. Although the query filter supports many operators, Okta only uses "eq" with the listed four fields. The SCIM server needs to support pages of results.

Query for Users Operation: Pagination

Request parameters:

- **startIndex**: 1 based – for pagination
- **count**: number of rows to retrieve per page.

Response attributes:

- **itemsPerPage**: Page size (matches "count" value from request)
- **totalResults**: Number of rows that match the filter criteria
- **startIndex**: The current index (matches "startIndex" value from request)

1. Request from Okta

- GET /scim/v2/Users?
startIndex=1&count=100

Response from SCIM Server

- HTTP 200
- Returns **ListResponse** of User resources

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

When configuring the SCIM application connector, the administrator must test the API connection and the authorization credentials. To get past this point, the SCIM server must implement listeners for the two calls listed. The Test API Credentials call only the first call to /scim/v2/Users.

Importing Users

- Import Wizard – tool available to Okta administrators
 - Configurable and automated user mapping rules
 - Manual exception handling (e.g. no or partial matches)

Import Results

 Import Now 0 imported users need review • 1 imported users confirmed

ALL NO EXACT PARTIAL IGNORED Search

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

An example of doing a SCIM query is the import wizard - a tool available to administrators. It is always enabled by default and has matching rules and a conflict resolution tool.

Test API Credentials

- Verifies authorization logic
 - Also executed when saving application connector changes
- Okay if array of resources is empty

1.	Request from Okta	Response from SCIM Server
	<ul style="list-style-type: none">• GET /scim/v2/Users? startIndex=1&count=2	<ul style="list-style-type: none">• HTTP 200• Returns ListResponse of User resources
2.	<ul style="list-style-type: none">• GET /scim/v2/Groups? startIndex=1&count=100	<ul style="list-style-type: none">• HTTP 200• Returns ListResponse of Group resources

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

When configuring the SCIM application connector, the administrator must test the API connection and the authorization credentials. To get past this point, the SCIM server must implement listeners for the two calls listed. The Test API Credentials call only the first call to /scim/v2/Users.

Retrieve Specific User Operation

1. Request from Okta

- GET /scim/v2/Users/{id}

Response from SCIM Server

- HTTP 200
- Returns **User** resource based on id

```
{  
  "active": true,  
  "id": "18feaf23-6d39-4968-85f4-187296697ed8",  
  "meta": {  
    "location": "http://ff8b76c7.ngrok.io/scim/v2/Users/18feaf23-6d39-4968-85f4-187296697ed8",  
    "resourceType": "User"  
  },  
  "name": {  
    "familyName": "Test",  
    "givenName": "Scim"  
  },  
  "schemas": [  
    "urn:ietf:params:scim:schemas:core:2.0:User"  
  ],  
  "userName": "scim@test.com"  
}
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The slide shows an example of a request for a single User based on the ID specified.

Create User Operation

Triggered by:

- User directly assigned to app
- User becomes member of group assigned to app

1.

Request from Okta

- GET /scim/v2/Users?
filter=userName+ eq+"{username}"

Response from SCIM Server

- HTTP 200
- Checks for existing person based on username and returns **ListResponse**

2.

Request from Okta

- POST /scim/v2/Users
- Sends new **User** resource

Response from SCIM Server

- HTTP 201
- Creates new user in data source, generates and returns unique Id in **User Resource**

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In the request sample on the slide, is the user creation operation and the schema the SCIM server receives.

In addition to the backslash-escaped location key in the meta section, the header response should also contain the same value for the Content-Location.

Update User Attributes Operation

Triggered by:

- Update in Okta to any mapped field for user assigned to application
- Update in directory or any application as a master to any mapped field for user assigned to application
- Addition of a new mapped field

1.	Request from Okta	Response from SCIM Server
	<ul style="list-style-type: none">• GET /scim/v2/Users/{id}	<ul style="list-style-type: none">• HTTP 200• Returns User resource based on id
2.	<ul style="list-style-type: none">• PUT /scim/v2/Users/{id}• Sends User resource	<ul style="list-style-type: none">• HTTP 200• Updates database and returns User resource

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The PUT call is mapped to a replace operation and represents a full copy of the modified user.

Deactivate User Operation

Triggered by:

- User unassigned from app.
- User removed from group assigned to app.
- User deactivated in Okta, directory or any application as a master
- User reactivated in Okta, directory or any application as a master

1.

Request from Okta

- PATCH /scim/v2/Users/{id}
- Sends **PatchOp** with replaced active field value

Response from SCIM Server

- HTTP 200
- Returns **User** resource

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The logic for a deactivation is to modify the value for the active field. The same message is sent on the re-activation of a user in Okta, except that active will be set to true.

Sync Password Operation

Triggered by:

- Push randomly generated password
- Push Okta password

1.

Request from Okta

- PATCH /scim/v2/Users/{id}
- Sends **PatchOp** with replaced password field value

```
"Operations": [
  {
    "op": "replace",
    "value": {
      "password": "Training123"
    }
  }
]
```

Response from SCIM Server

- HTTP 200
- Returns **User** resource

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Sync Password is often used in situations with mobile and thick client apps. When turning on Sync Password, the Create User and Update User Attributes include the password.

SCIM HTTP Errors

HTTP Error Code	Example
400 Bad Request	Failure to parse JSON.
401 Unauthorized	Password with Basic Auth failed.
403 Forbidden	Update User Attributes failed – configured OAuth user does not have permission to modify user.
404 Not Found	User lookup based on Id not found in user database.
409 Conflict	Create User failed - existing unique credentials already exist.
500 Internal Server Error	Endpoint user database returned error code.
501 Not Implemented	Deactivate User enabled in Okta, but not implemented in SCIM server.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Different possible HTTP errors that might be returned by the SCIM server.

JSON Error Messages

- For 400 responses, scimType enumerated values:
 - invalidFilter
 - tooMany
 - uniqueness
 - mutability
 - invalidSyntax
 - invalidPath
 - noTarget
 - invalidValue
 - invalidVers
 - sensitive

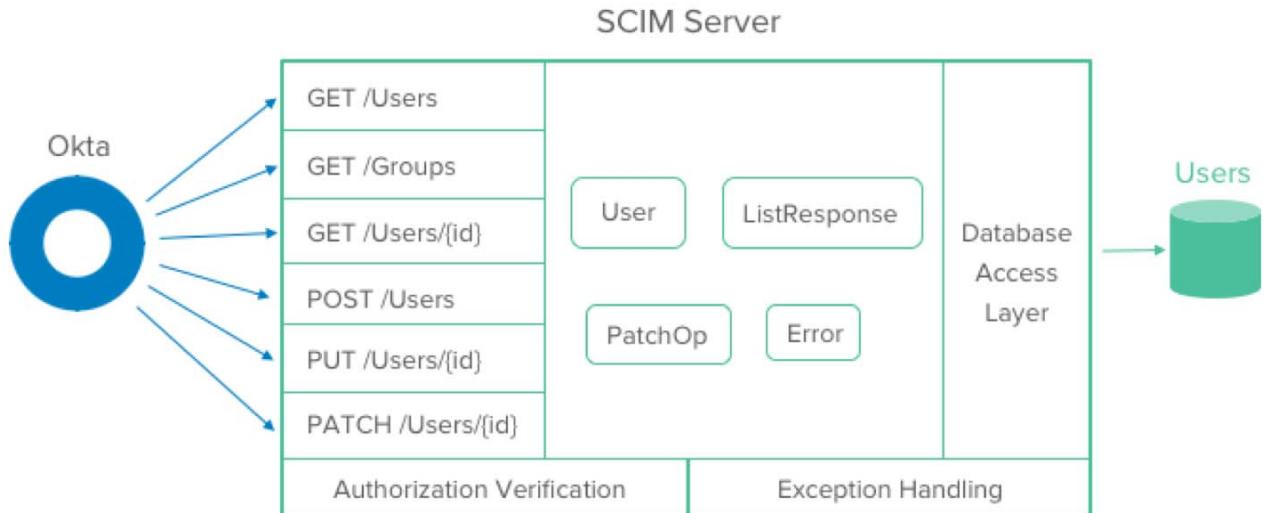
```
{  
  "schemas": ["urn:ietf:params:scim:api:messages:2.0/Error"],  
  "scimType": "mutability",  
  "detail": "Attribute 'id' is readOnly",  
  "status": "400"  
}
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

A Bad Request must return a specific error message and an associated scimType.
http://developer.okta.com/docs/guides/scim_guidance.html#support-for-scim-error-messages

Design



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The server will need to implement many endpoints, contain authorization verification logic, and handle exceptions. The four main structures that need to be implemented are shown in the middle, and the database access layer to whatever the user store is also needs to be implemented.

Testing with Runscope

- Execute scripted requests
- See and analyze test results



↑ [Ttadfile1.ngrok.io](#)

GET /scim/v2/Users?username=julian&filter=username%20eq%20%22julian.jensen@example.com%22&startIndex=1

Make sure random user doesn't exist.

Variables used

- ✓ Request 'url' set to "https://ttadfile1.ngrok.io/scim/v2/Users"
- ✓ Request 'param[0].value' set to 'username eq "julian.jensen@example.com"'

Assertions

- ✓ Status - "200" was a number equal to 200
- ✓ body.totalResults - "1" was a number equal to 1
- ✓ body.schemas -- "[\"urn:ietf:params:scim:api:messages:2.0>ListResponse\"]" did have the value 'urn:ietf:params:scim:api:messages:2.0>ListResponse'
- ✓ Response Time of 200.0ms was less than 600

↑ [Ttadfile1.ngrok.io](#)

GET /scim/v2/Users/732a0533d8c60779619d8d8c3bfcc703/

Check error schema

Variables used

- ✓ Request 'url' set to "https://ttadfile1.ngrok.io/scim/v2/Users/732a0533d8c60779619d8d8c3bfcc703/"

Assertions

- ✓ Status - "404" was a number equal to 404
- ✗ body.detail -- was empty -- Unable to parse "body" value as JSON
- ✗ body.schemas -- "None" did not have the value 'urn:ietf:params:scim:api:messages:2.0>Error' -- Unable to parse "body" value as JSON
- ✓ Response Time of 245.0ms was less than 600

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Runscope is a web application that is very useful for testing and debugging. Okta has created some pre-built tests for verifying SCIM services. (beta) <https://github.com/joelfranusic-okta/runscope-scim-test>

The testing functionality allows you to record and define requests, then make assertions. It also has the ability to script setting and reading variables.

Capturing Traffic with Runscope

- HTTP Proxy
- View Request and Response details
- Useful for debugging



```

HEADERS
Content-Length: 381
Content-Type: application/json
Date: Mon, 27 Jun 2016 01:12:34 GMT
Location: http://1a416910.ngrok.io/scim/v2/Users/scim%40test.com
Server: Werkzeug/0.11.9 Python/2.7.11

BODY
{
  "active": true,
  "id": "18feaf23-6d39-4968-85f4-187296697ed8",
  "meta": {
    "location": "http://1a416910.ngrok.io/scim/v2/Users/18feaf23-6d39-4968-85f4-187296697ed8",
    "resourceType": "User"
  },
  "name": {
    "familyName": "Test",
    "givenName": "Scim"
  },
  "schemas": [
    "urn:ietf:params:scim:schemas.core:2.0:User"
  ],
  "userName": "scim@test.com"
}
  
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

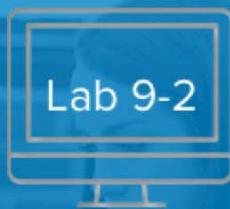
Additional Information

Runscope can also act as an HTTP proxy to capture and forward messages.



Launch and Test the SCIM Server

- Test and Verify the SCIM Server with Runscope



Define a Native SCIM Application in Okta

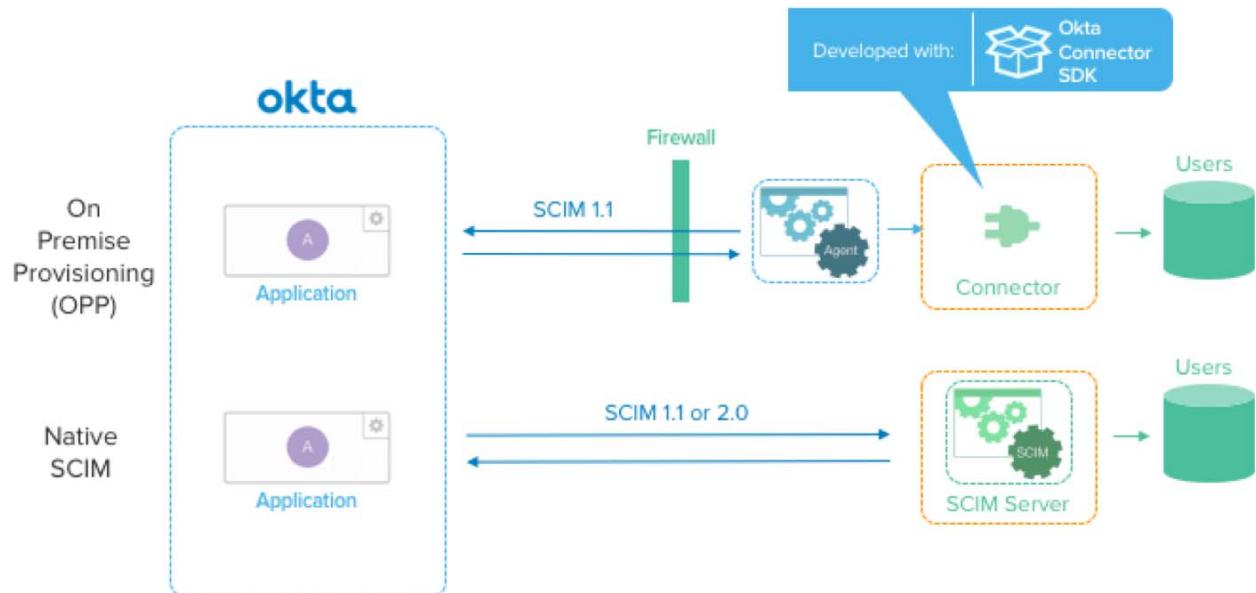
- Test the API Connection from Okta
- Enable Provisioning in Okta and Capture Traffic in Runscope



Extend Native SCIM with Custom Attributes

- Setup Custom Attributes in Okta
- Test the Custom Attribute Mapping
- (Optional) Review the SCIM Server Code

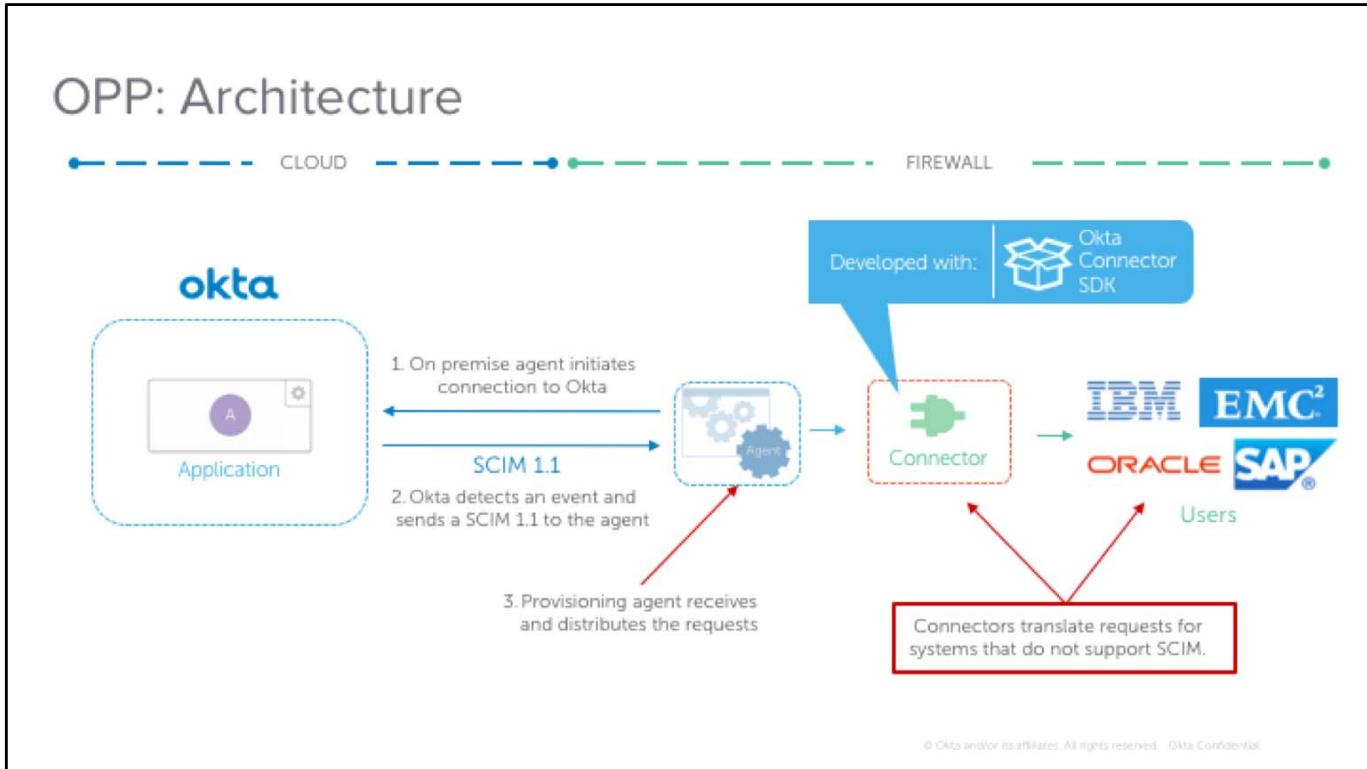
Provisioning Options: Review



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In the next few slides, you learn more about the On Premise Provisioning (OPP).



Additional Information

OPP enables lifecycle management in systems located on your network and is delivered through two main components:

- The **Provisioning Agent** act as a broker. It receives and distributes provisioning requests from Okta behind the firewall.
- The **On Premise Connector** is a customized code that translates provisioning requests from Okta (SCIM protocol) for systems that do not support SCIM.

Okta Provisioning Agent



- Available for Windows (exe) and Linux (rpm)
 - Runs on premise
 - Supports Linux Distros: CentOS and Red Hat Enterprise (RHEL)
- Initiates connection to Okta cloud through an HTTPS and does not require special firewall rules.
- Can handle connections to multiple backend connectors
- Sends messages according to SCIM 1.1 protocol

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

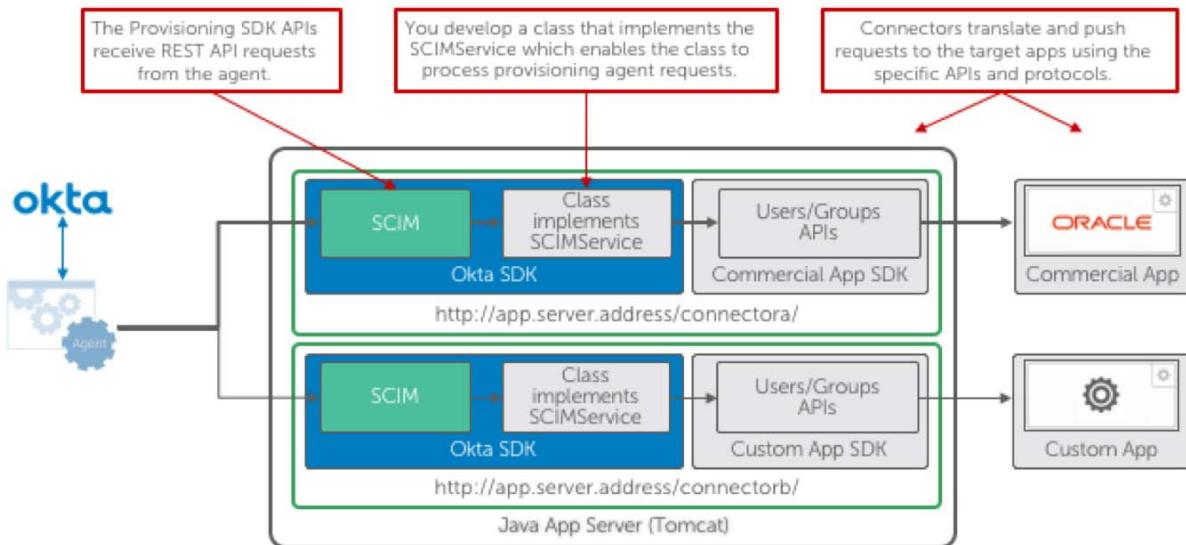
Connectors



- Act in the Application Integration Layer
- Integrate Okta Provisioning with internal systems that do not support SCIM.
- Are developed in Java and hosted in a J2EE application server.
 - Supported App Server: Tomcat.
- Listen the Provisioning Agent calls in a HTTPS endpoint.
 - Example: <https://tomcat.host:port/connectorx>
- Use Okta's Provisioning Connector SDK to understand the agent requests.
 - The SDK includes APIs, a tester utility, and two code samples.
- Translate requests to systems using the specific Java APIs or protocols.
 - For example: SQL, People Code, SAP Doc, EJB, RPC.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Connectors: Architecture



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

You learn to use the Provisioning SDK once. Implementation for other connectors will be similar.

In addition to using the provisioning agent and the connector, you can integrate applications without using connectors:

- When the you can modify the custom application with direct support to SCIM
- Without using the provisioning agent and when your application is accessible through the internet.

The main difference in new connectors are the different APIs and protocols to contact the target.

Connectors: MySQL Code Sample Interface

```

public class MySqlSCIMServiceImpl implements SCIMService {
    ...
    @Override
    public void createUser(SCIMUser user) throws SCIMException {
        String query = "INSERT INTO users (first_name, last_name, gender, hire_date) VALUES (?, ?, ?, ?)";

        try (Connection conn = dataSource.getConnection();
             PreparedStatement stat = conn.prepareStatement(query, Statement.RETURN_GENERATED_KEYS)) {
            Map<String, JsonNode> customPropertiesMap = user.getCustomPropertiesMap();
            JsonNode customNode = customPropertiesMap.get("USER_CUSTOM_ATTRIBUTES");
            stat.setString(1, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_BIRTH_DATE).asText());
            stat.setString(2, user.getFirstName());
            stat.setString(3, user.getLastName());
            stat.setString(4, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_GENDER).asText());
            stat.setString(5, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_HIRE_DATE).asText());

            //get the new user id from the DB
            String newUserId;
            ResultSet rs = stat.getGeneratedKeys();
            if (rs.next()) {
                newUserId = Integer.toString(rs.getInt(1));
                user.setId(newUserId);
                LOGGER.info("Created a new user with the id " + newUserId);
            } else {
                ...
            }
        }
    }
}

```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The connector code implements the SCIMService. This interface sets requirements for your implementation code, such as having the createUser method.

Connectors: MySQL Code Sample Methods

Each method required by the SCIMService interface perform a lifecycle management task in the target system with user or group information provided by Okta.

The createUser method receives information about the user in the user object.

```
public class MySqlSCIMServiceDept implements SCIMService {
    // This method creates a user ...28 lines ...
    @Override
    public SCIMUser createUser(SCIMUser user) throws OnPremUserManagementException {
        // Create the new user
        String query = "INSERT INTO users (first_name, last_name, email, password, birth_date) VALUES (?, ?, ?, ?, ?)";
        PreparedStatement stat = conn.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
        Map<String, JsonNode> customPropertiesMap = user.getCustomPropertiesMap();
        JsonNode customNode = customPropertiesMap.get("CUSTOM_NODE");
        stat.setString(1, user.getFirstName());
        stat.setString(2, user.getLastName());
        stat.setString(3, user.getEmail());
        stat.setString(4, user.getPassword());
        stat.setString(5, user.getBirthDate());
        stat.executeUpdate();
        ResultSet rs = stat.getGeneratedKeys();
        if (rs.next()) {
            newUserId = Integer.parseInt(rs.getInt(1));
            user.setUserId(newUserId);
            LOGGER.info("Created a new user with the id " + newUserId);
        } else {
            throw new OnPremUserManagementException("User creation failed");
        }
    }
}
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

- The connector code contains methods for each lifecycle management operation.
- These methods perform lifecycle management task in the target system with user or group information provided by Okta.
- If the method execution is successful, the method returns the user object back to Okta.
- The user object may retrieve additional data that's created by the connector or target system automatically during the provisioning.
- If the method execution is unsuccessful, the method returns error information that's parsed at Okta and displayed in logs.

Connectors: MySQL Code Sample Target Specific

```

public class MySQLSCIMServiceImpl implements SCIMService {
    /** This method creates a user ...20 lines */
    @Override
    public SCIMUser createUser(SCIMUser user) throws OnPremUserManagementException {
        PreparedStatement stmt = null;
        ResultSet rs = null;
        Connection conn = null;
        try {
            //Start the INSERT query
            String query = "INSERT INTO employees (birth_date, first_name, last_name, gender, hire_date) VALUES (?, ?, ?, ?, ?)";
            //get a new connection and start a new transaction
            conn = getDatabaseConnection(); conn.setAutoCommit(false);
            //create the statement and make sure our auto-incremented ID is returned
            stmt = conn.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
        }
    }
}

//Start the INSERT query
String query = "INSERT INTO employees (birth_date, first_name, last_name, gender, hire_date) VALUES (?, ?, ?, ?, ?, ?)";

//get a new connection and start a new transaction
conn = getDatabaseConnection(); conn.setAutoCommit(false);

//create the statement and make sure our auto-incremented ID is returned
stmt = conn.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);

```

Depending on the system, you find different APIs and protocols.

Examples include:

- Mainframe: RPC calls
- Linux OS: SSH commands.
- SOA: SOAP/XML call.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Inside each method, you will find code that uses the target system APIs or protocols.

For MySQL, you have code that handles SQL queries, database connections, commit operations, and statement configuration.

Depending on the target system implementation, you may find different code.

Connectors: MySQL Code Sample Utils

```
//populate our prepared statement with all the parameters
Map<String, JsonNode> customPropertiesMap = user.getCustomPropertiesMap();
JsonNode customNode = customPropertiesMap.get(USER_CUSTOM_URN);
stmt.setString(1, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_BIRTH_DATE).asText());
stmt.setString(2, user.getName().getFirstName());
stmt.setString(3, user.getName().getLastName());
stmt.setString(4, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_GENDER).asText());
stmt.setString(5, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_HIRE_DATE).asText());
```

VALUES (1, 1, 1, 1, ?)

```
//populate our prepared statement with all the parameters
Map<String, JsonNode> customPropertiesMap = user.getCustomPropertiesMap();
JsonNode customNode = customPropertiesMap.get(USER_CUSTOM_URN);
stmt.setString(1, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_BIRTH_DATE).asText());
stmt.setString(2, user.getName().getFirstName());
stmt.setString(3, user.getName().getLastName());
stmt.setString(4, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_GENDER).asText());
stmt.setString(5, customNode.get(CUSTOM_SCHEMA_PROPERTY_NAME_HIRE_DATE).asText());

//get the new user id from the DB
String newUserId;
rs = stmt.getGeneratedKeys();
if (rs.next()) {
    newUserId = Integer.toString(rs.getInt(1));
    user.setId(newUserId);
    LOGGER.info("Created a new user with the id " + newUserId);
} else {
```

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

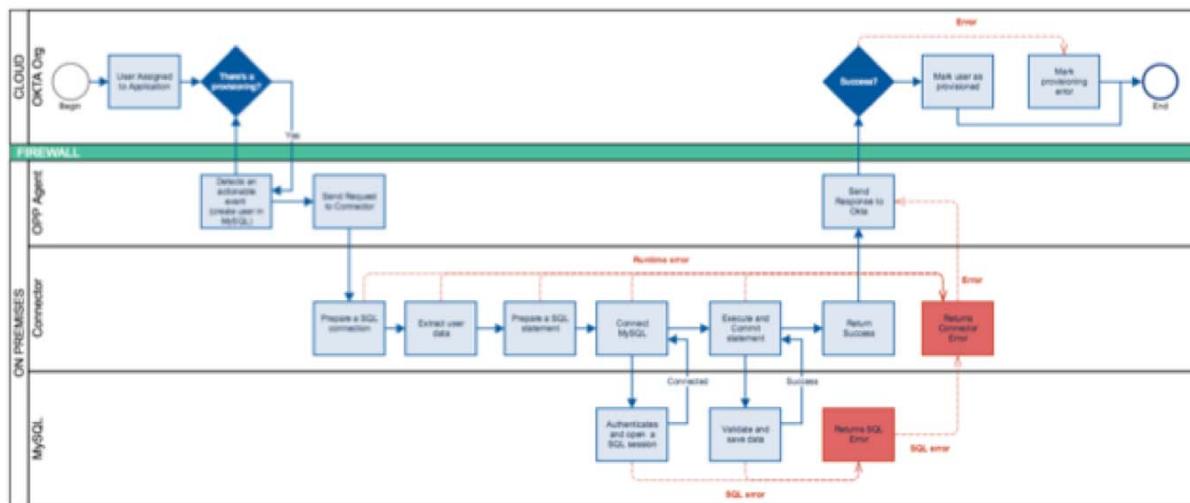
Additional Information

The Okta Connector SDK comes with util APIs that simplify the way you code. For example, you can use:

- `getName().getFirstName()`: Getters in the user object to obtain a user data without parsing the SCIM payload.
- `getCustomPropertiesMap()`: Extract custom user attributes from the SCIM payload.

To learn more about the on premise provisioning APIs and utility objects refer to the provisioning SDK documentation.

OPP: In Action

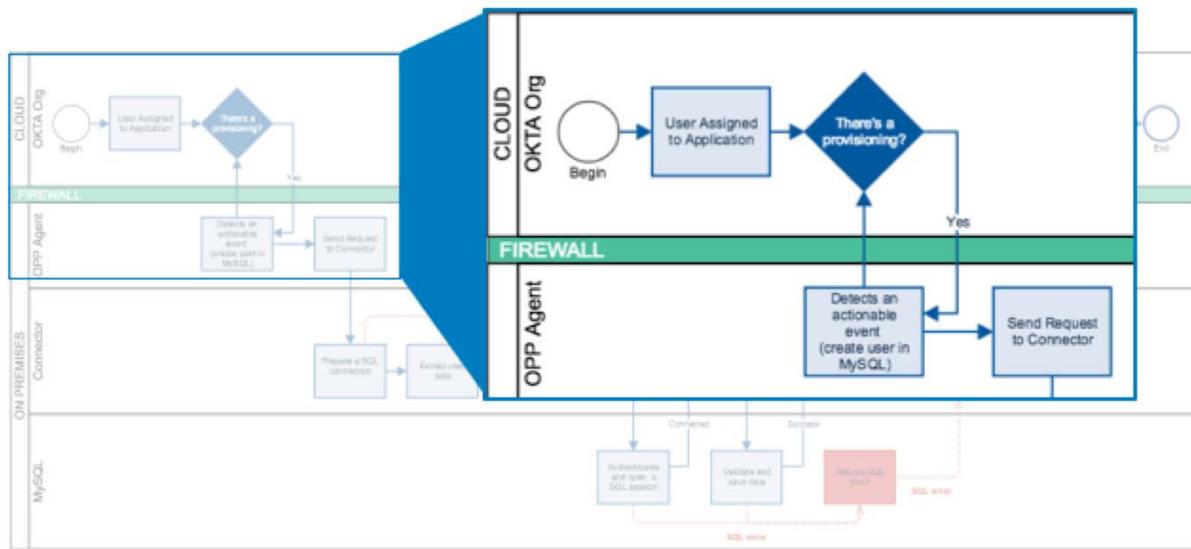


© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

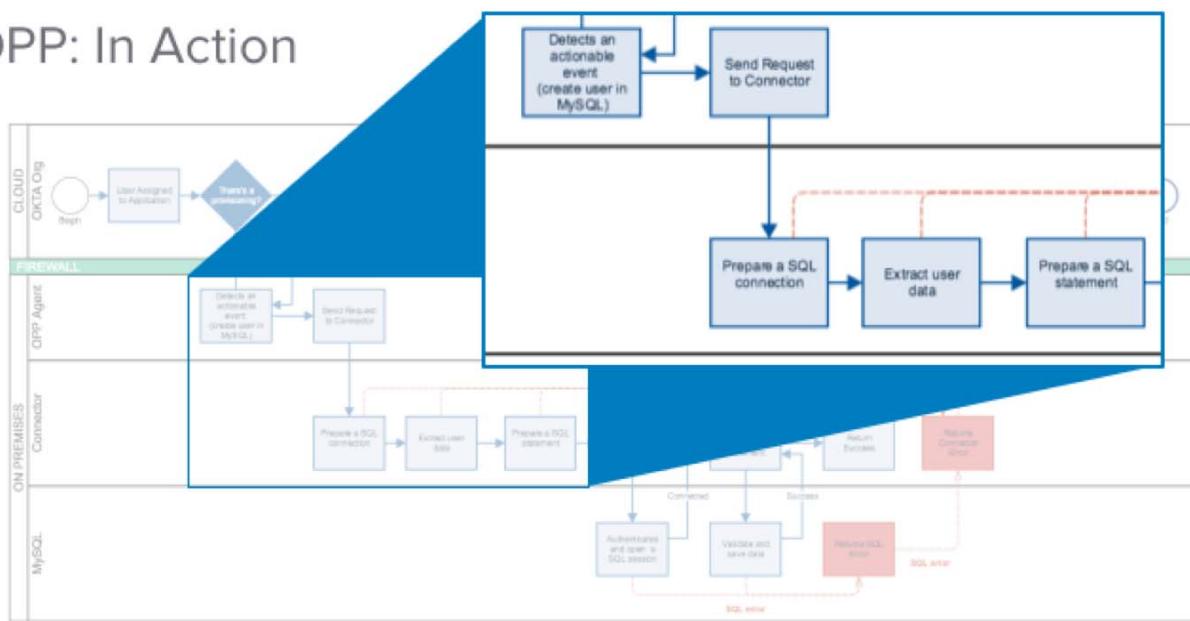
This slide shows how Okta lifecycle management tasks are processed through OPP.

OPP: In Action



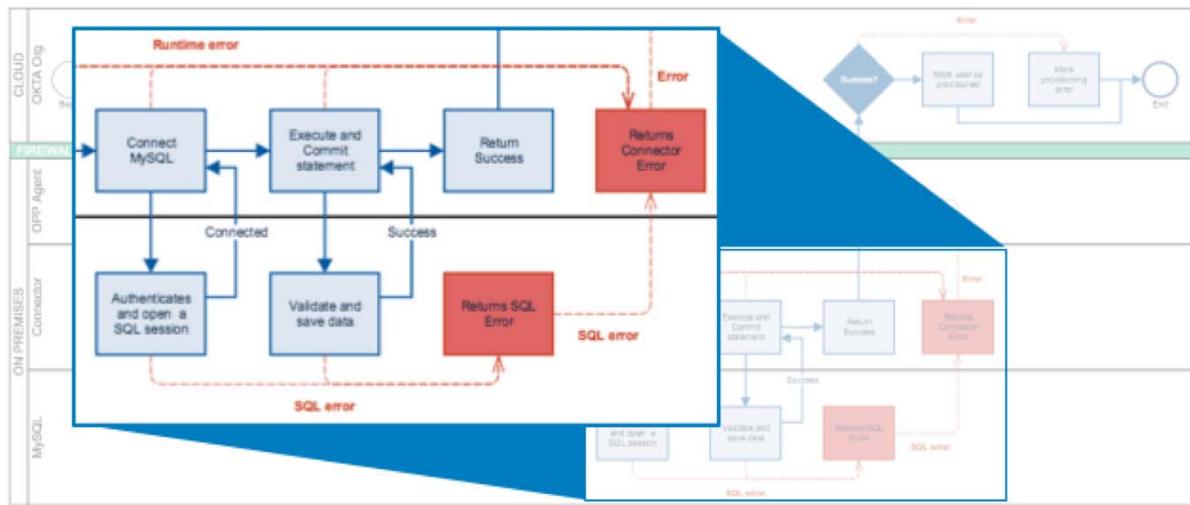
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

OPP: In Action



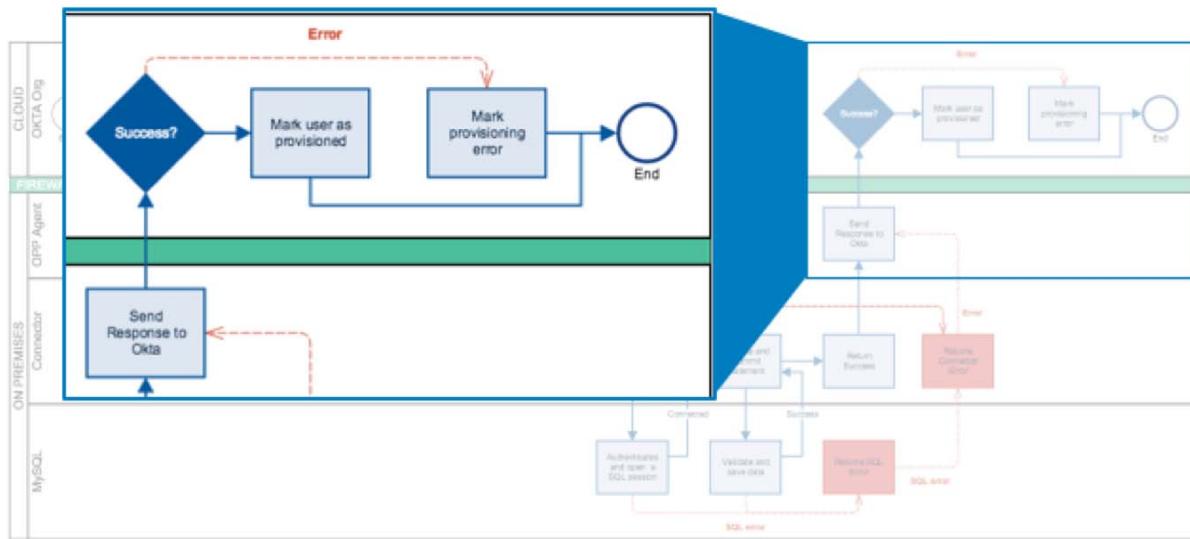
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

OPP: In Action



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

OPP: In Action



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

OPP: When to use the agent ?

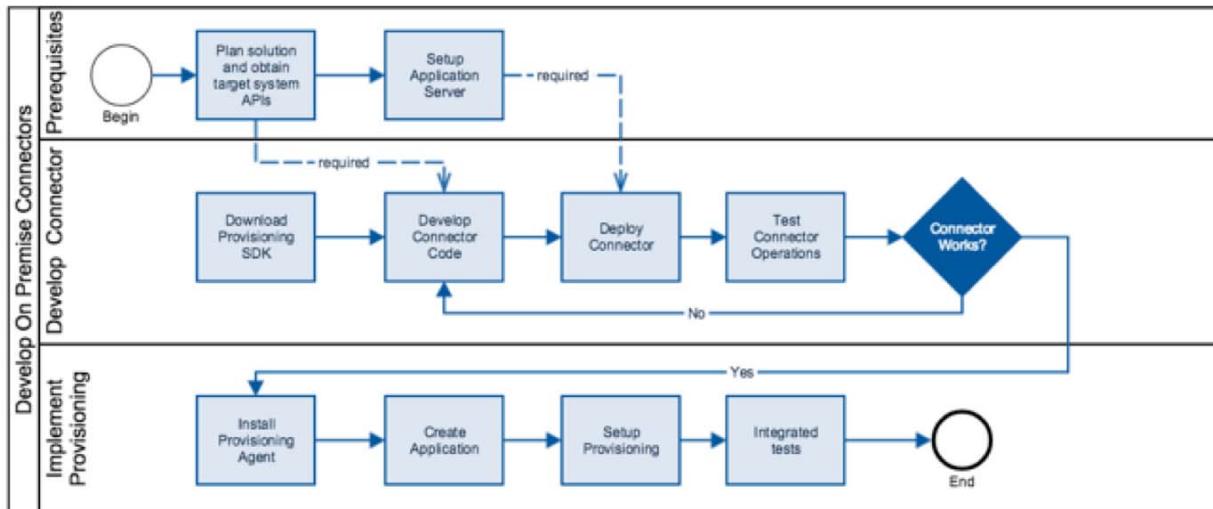
Scenarios	OPP
Provision to a custom app hosted in a PaaS provider, such as AWS or Heroku.	
Provision to a custom app hosted on-premises without internet access.	
Provision to a proprietary app without support to SCIM and hosted on-premises.	
Provision to a cloud provider.	

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

- **Scenario 1 solution:** You can modify your custom app to understand Native SCIM provisioning from Okta.
- **Scenario 2 solution:** You can use the provisioning agent to reach your on-premises application without internet access.
- **Scenario 3 solution:** You can use the provisioning agent to reach your on-premises proprietary app. If the app does not support Native SCIM, you can either develop a connector using the proprietary app APIs or ask the proprietary app owner to add a feature to support Okta provisioning.
- **Scenario 4 solution:** If you have a cloud app, you can request Okta and your cloud provider to provide the provisioning feature in OAN.

OPP: Development/Configuration Lifecycle



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

This slide shows the main steps to implement Okta OPP.

Connectors: Development Tips

- During development, use Docker to get an application server up to speed:

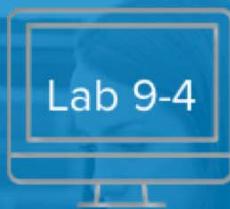
```
docker run --name tomcat --link mysql --rm -p 8080:8080 -d tomcat:6-jre7
```

- When developing code, use the SDK assets:


© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

- Connectors are required to run in a certified Java application server. To save time and start developing immediately, you can pull the Tomcat Docker image.
- The Okta Connector SDK comes with APIs, complete code samples, and a tester utility to simplify your connector development.



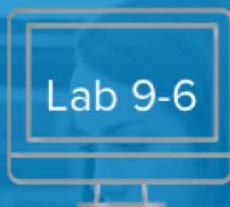
Deploy an On Premise Provisioning Connector

- Launch MySQL and Tomcat Servers
- Download the Connector SDK
- Deploy, Launch, and Test the Connector



Install and Configure the Okta Provisioning Agent

- Download, Install, and Configure the Provisioning Agent
- Verify the Agent Status



Integrate the Custom Application Provisioning

- Launch ngrok
- Register the MySQL Application
- Enable and Configure Provisioning
- Work with Users



- Scope applications that support OPP with attributes for provisioning and lifecycle management events.
- Make sure that the target application supports external account management through APIs or protocols.
- If the target application does not support provisioning through open protocols such as REST or SOAP, use the Java API.
- Consider additional provisioning agents for High-Availability.
- Consider Load Balancer, HTTPS, and Authentication for connector endpoints.
- Use a dedicated service account to connect the connector to the target application.

Continued

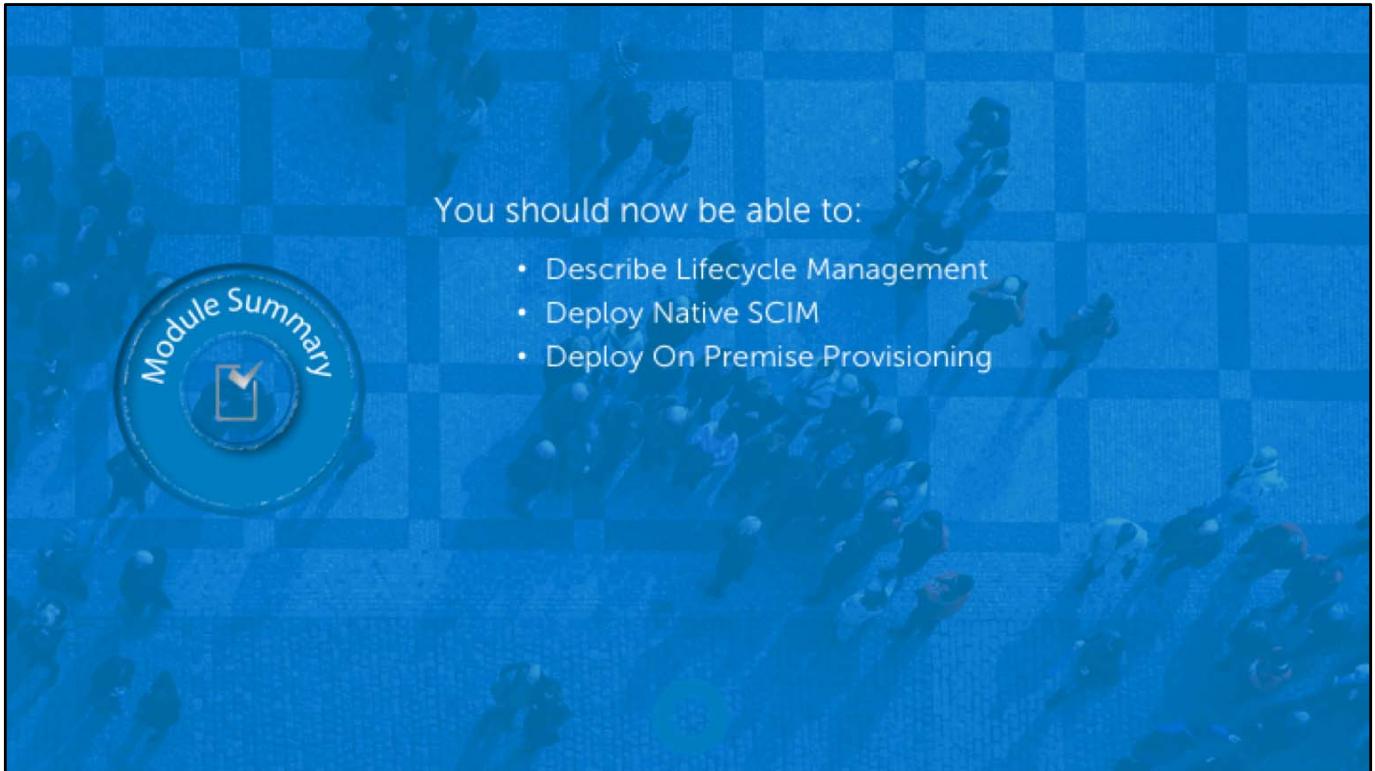


- Use UD and Workflow features to improve your provisioning capabilities.
- Do not port to connector code data transformations that you can perform through UD attribute mapping.
- When developing error handling (Java Exceptions), throw complete error messages back to Okta and avoid generic error messages.
- During connector testing, validate all lifecycle management events such as create, activate, deactivate, update, and disable.
- Have an onboarding plan to match accounts with users correctly and avoid disabling service accounts in the target application.
- To increase value after integration, suggest plans to improve account security in the target application, such as disabling rogue accounts and privileges and applying least privilege in existing accounts.



When troubleshooting OPP issues:

- Look at the Dashboard, Reports, and System Logs.
- Verify the agent, connector, and target app status.
- Run the connector tester.



End of module review question:

1. Which action is a supported SCIM operation?

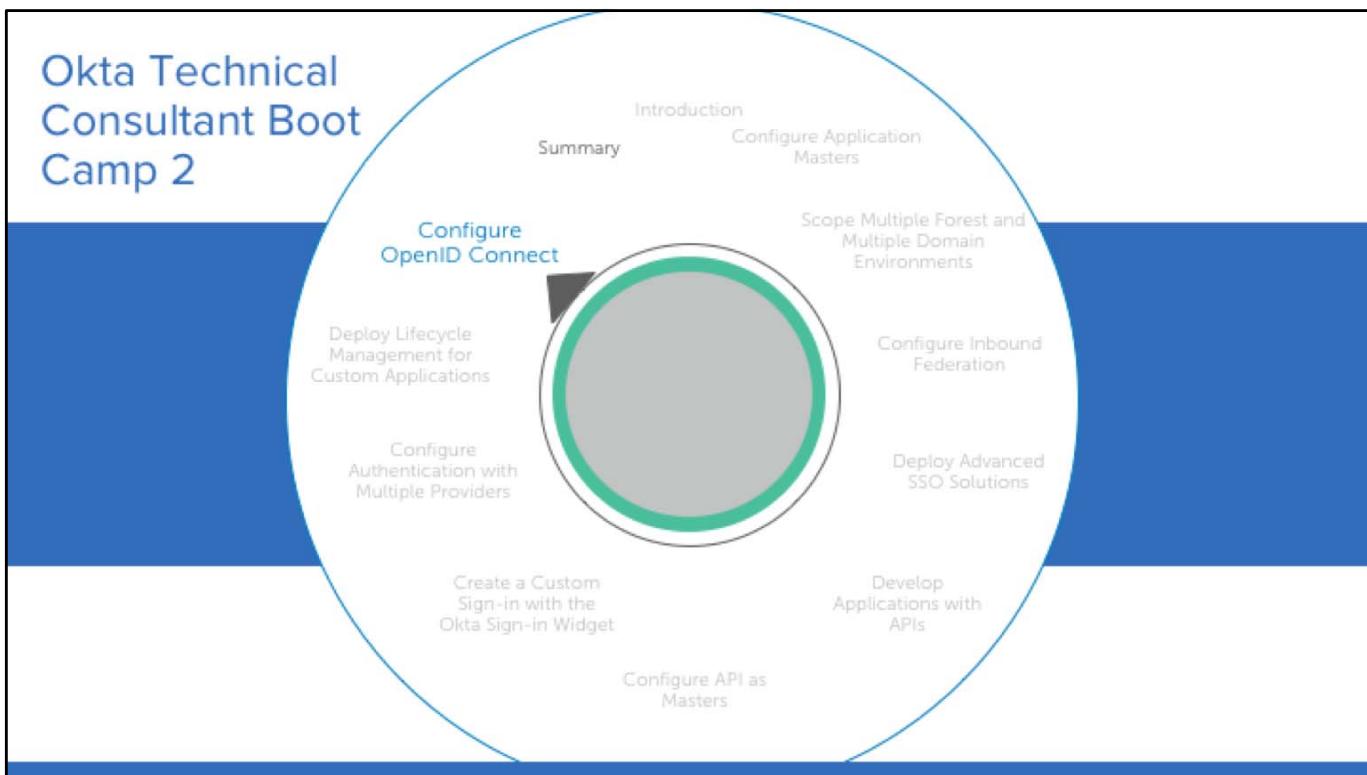
- a. Create
- b. Delete
- c. Query with Post
- d. Query Filter with meta.lastModified

Source: Page 222

2. What is an advantage of an On-Premise Provisioning Connector?

- a. Requires Java
- b. Supports groups
- c. Requires an on-premise agent
- d. Can be written in any language

Source: Page 243



Configure OIDC

OpenID Connect (OIDC) is an open standard published in early 2014 that defines an interoperable way to use OAuth 2.0 to perform user authentication. Instead of building a different protocol to each potential identity provider, an application can speak one protocol to as many providers as they want to work with.

Configure OIDC



Describe Authentication Basics OAuth 2 and OIDC
Describe Claims and Scopes
Describe the OIDC Implicit Flow
Deploy SSO with OIDC and the Auth SDK

Configure OIDC Overview

In this module, you will see how you can help customers use OIDC to connect applications with Okta.

This module consists of several labs and review questions.

OIDC Overview

Enables you to...

- Use Okta as an OAuth authorization server for social authentication, SSO, and API access management using OIDC.

Is important because...

- Many current authentication, authorization, and identity solutions use OIDC.

A Simple Solution

The Stored Password Anti-Pattern

Import your Fantasy Football leagues from NFL.com

Please enter your NFL.com account information so that we can access your leagues:

Username this is the username that you use to log in to <http://www.nfl.com/fantasyfootball>

Password

Public Leagues [Enter your league URL directly instead.](#)

[Next >](#)



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

To import data from system A into system B, it is not suggested that system A hold user credentials to system B. This creates a weakness where if system A is hacked, then they now have access to system A and system B.

An example integration between applications: Employees have to manually re-type client and sales data into the order management system. That process would be faster and less prone to errors if details of the closed opportunities in the sales CRM system could be copied into the order management system. The goal would be to allow the order management system to read data from the sales system.

OAuth 2.0

Modern delegation authorization framework

- Enables independent apps to securely share
 - ✓ Modern: Supports multiple app architectures
 - ✓ Token-based: avoids stored password anti-pattern
 - ✓ User-specific access
 - ✓ Scoped access
 - ✓ Revocable
 - ✓ Standardized
 - ✓ Secure



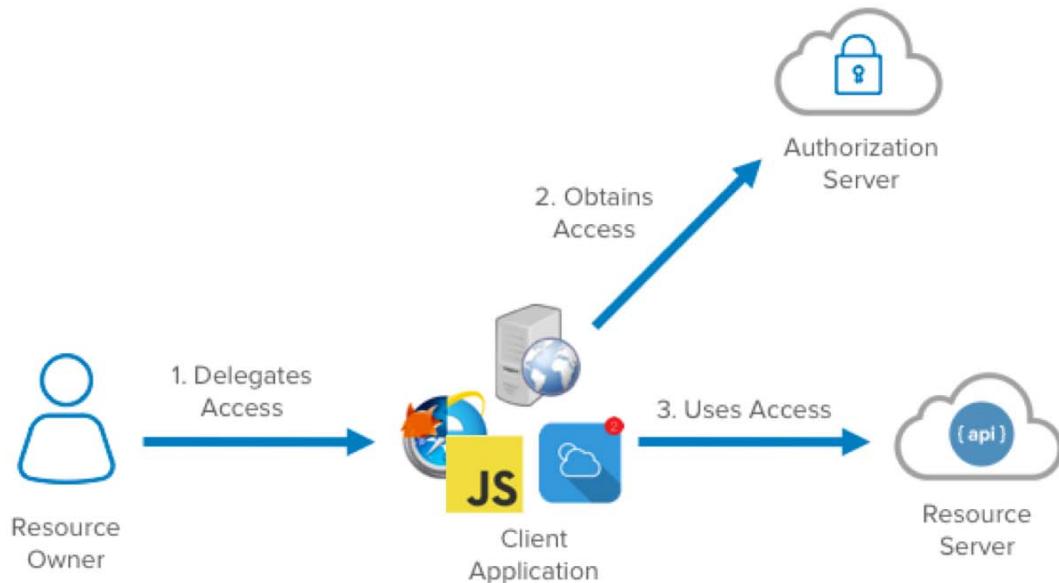
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

OAuth simplified:

1. Application requests authorization from user.
2. User authorizes application and delivers proof.
3. Application presents proof of authorization to server to get a token.
4. Token is restricted to only access what the user authorizes for the specific application.

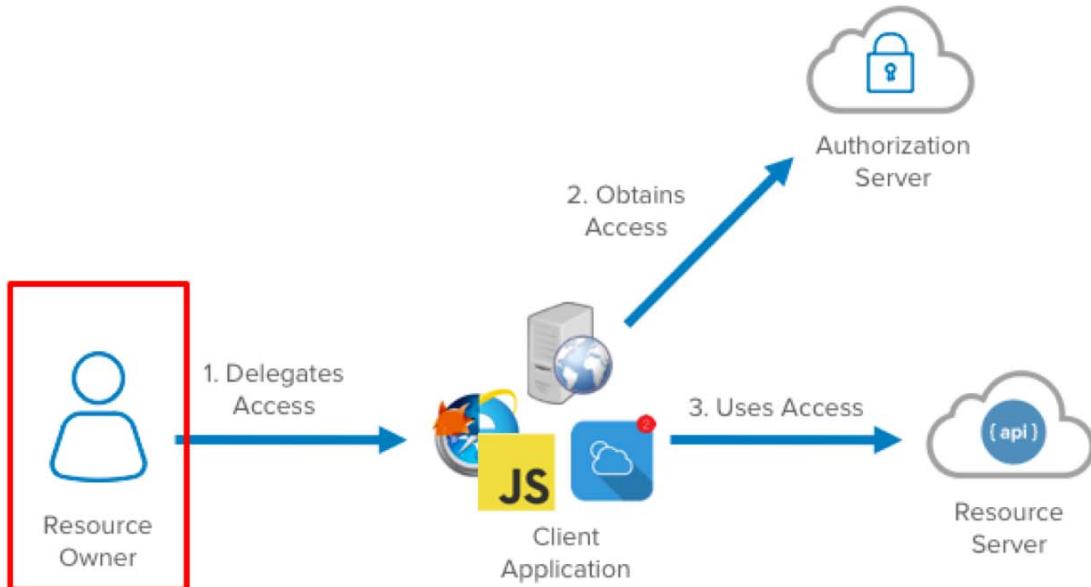
OAuth 2.0 Actors



Additional Information

At a very high level, there are four main actors in the OAuth relationships. The resource owner delegates access to the client application. The client application then obtains access from the resource server, using one of the OAuth flows to be discussed shortly. The client application will then use the granted access to utilize the protected resources.

OAuth 2.0 Actors



Additional Information

First we'll take a look at the Resource Owner in these relationships.

The Resource Owner

The person who authorizes access to the resource.



In a consumer situation, the [end user](#).

Consent is given the first time the service is used.

In an enterprise situation, typically the [administrator](#).

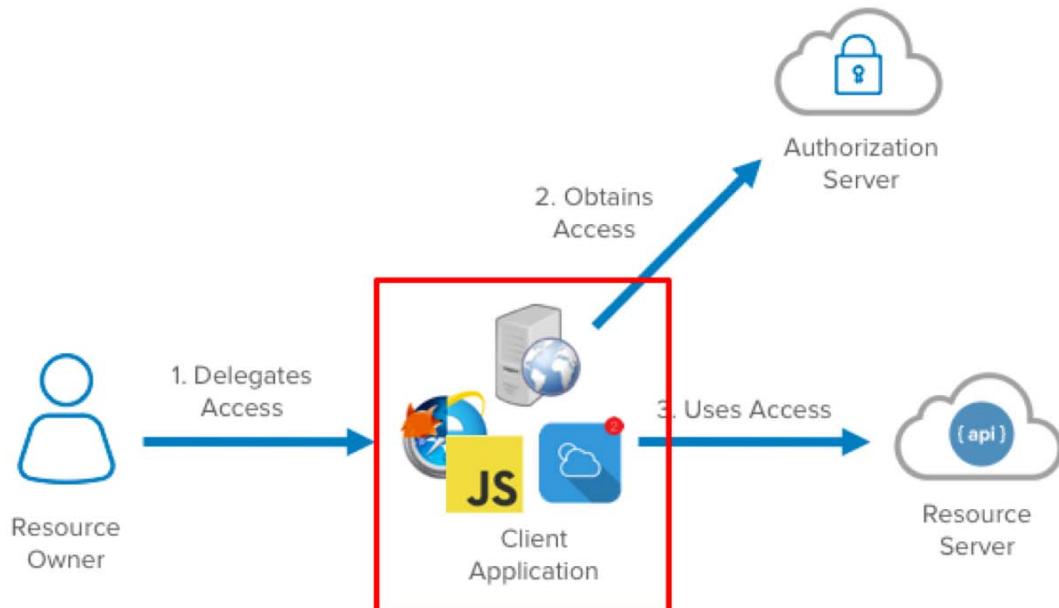
Consent is given during the assignment of access to the resource.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

In most discussions of OAuth the resource owner is the end user. This is most applicable to consumer situations. In enterprise situations though, it is typically the administrator who grants access to the client application by assigning it to the end user.

OAuth 2.0 Actors



Additional Information

The client application is the consumer of the resource. There are many types of applications which can participate in the OAuth flows, including JavaScript browser-based clients, mobile apps, and server-side web applications.

The Client Application



The application which is the consumer of the data or services.



A **learning management system** would be more automated if it could schedule virtual training sessions in a web conferencing system.



A **travel and expense application** would be more streamlined if client data could be read from a professional services automation system.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Samples of client applications for real-world scenarios.

Two Types of Client Application Flows

Trusted Clients – can secure secrets

- Web Server Apps



- Long-term access
- Secured via Redirect URI and Client Secret

Untrusted Clients – cannot secure secrets

- Native Apps
- JavaScript Apps



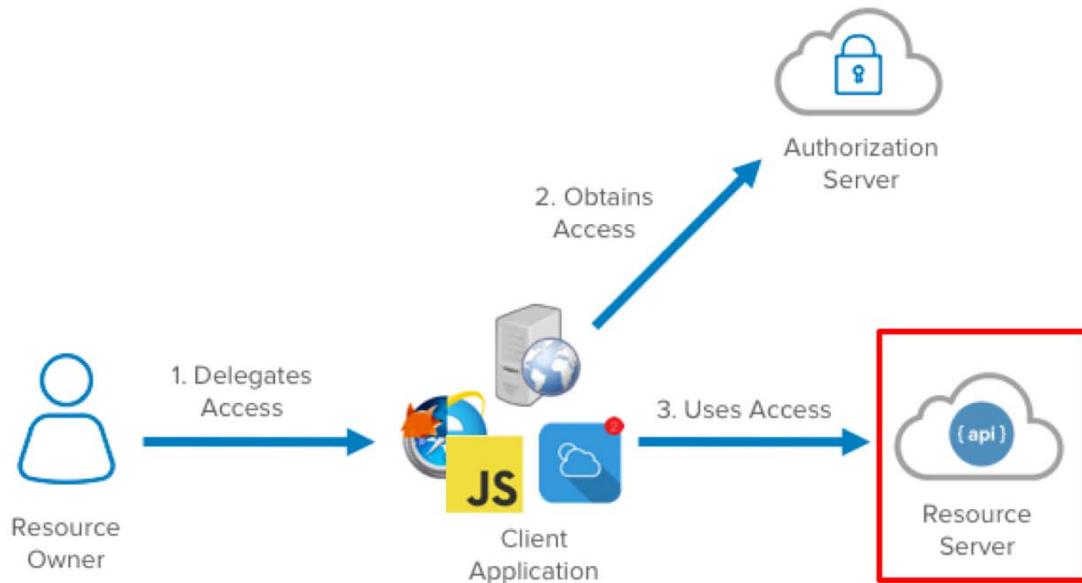
- Short-term access
- Secured via Redirect URI

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Broadly, in OAuth, there are two types of clients, trusted and untrusted. Because the attack surface area of untrusted apps are greater compared to trusted apps, trusted apps are considered more secure. This isn't to say that one should be used and the other should not. In certain scenarios, depending on the data and the system being accessed, the trusted model might be required, but in other scenarios the untrusted model might be well within the security requirements. Typically in trusted apps, the data access can be greater (e.g. read/write) and longer term, and in untrusted apps the access would be less (e.g. read-only) and shorter term.

OAuth 2.0 Actors



Additional Information

The resource server is the system to be used or the data to be consumed.

The Resource Server



The application which is the provider of the data or services.



A learning management system would be more automated if it could schedule virtual training sessions in a [web conferencing system](#).



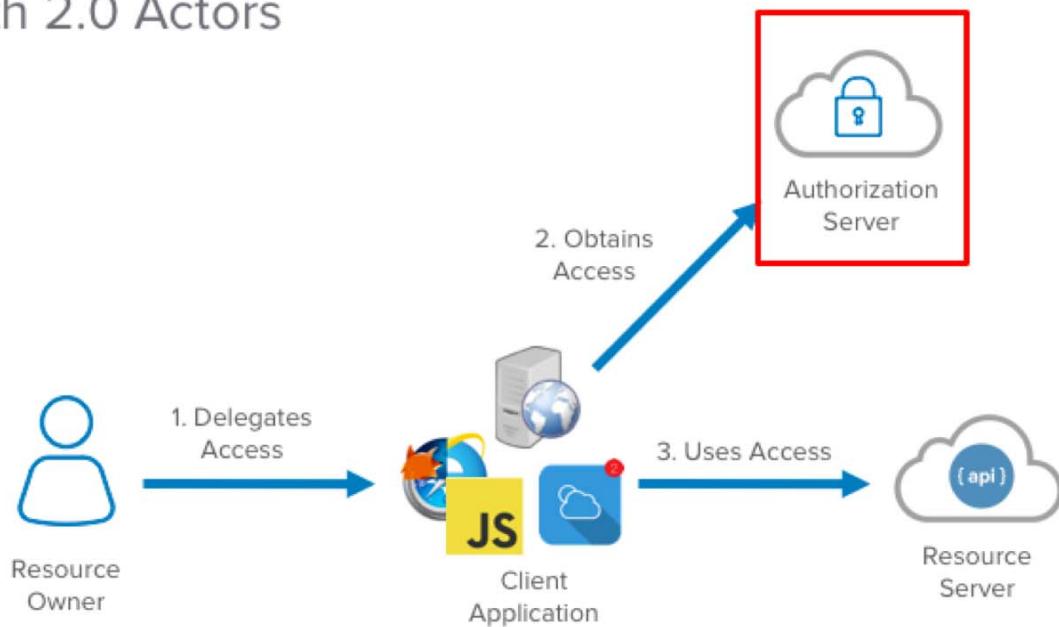
A travel and expense application would be more streamlined if client data could be read from a [professional services automation system](#).

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Samples of resource servers for real-world scenarios.

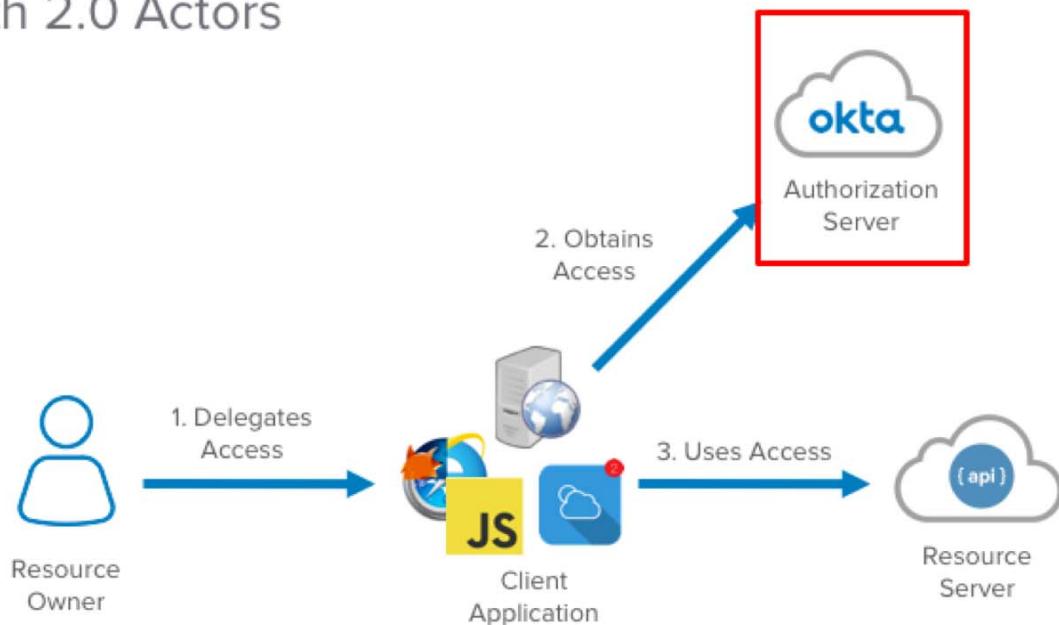
OAuth 2.0 Actors



Additional Information

The authorization server is responsible for providing access for the clients to the resources.

OAuth 2.0 Actors



Additional Information

In our architecture, Okta is the authorization server.

The Authorization Server



The central point for secure access control.

- Registers the Client Application.
- Authenticates the End User.
- Provides administrative control for the Resource Owner.
- Generates tokens for the Resource Server.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

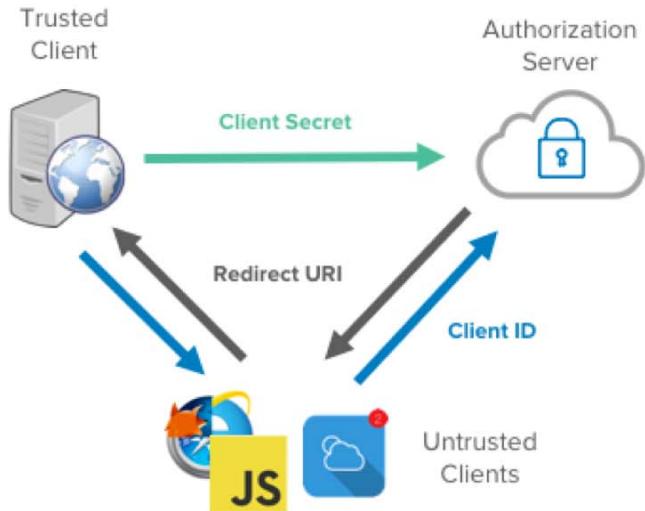
Additional Information

The authorization server provides many services. At a high level, it is the central system to control access to resources. In the OAuth architecture, Okta fulfills the role of the authorization server. In the next few slides, we'll take a look at how client applications are registered, interfaces for the end user to use to authenticate and the administrator to control, and ultimately produce tokens which are used by the resource servers.

Client Registration

Establishes a trust relationship between the client application and the authorization server

- Redirect URI: when capturing user consent, where the authorization code or access token can be sent
- Client ID: public identifier used in all situations
- Client Secret: private passcode only used by trusted clients



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

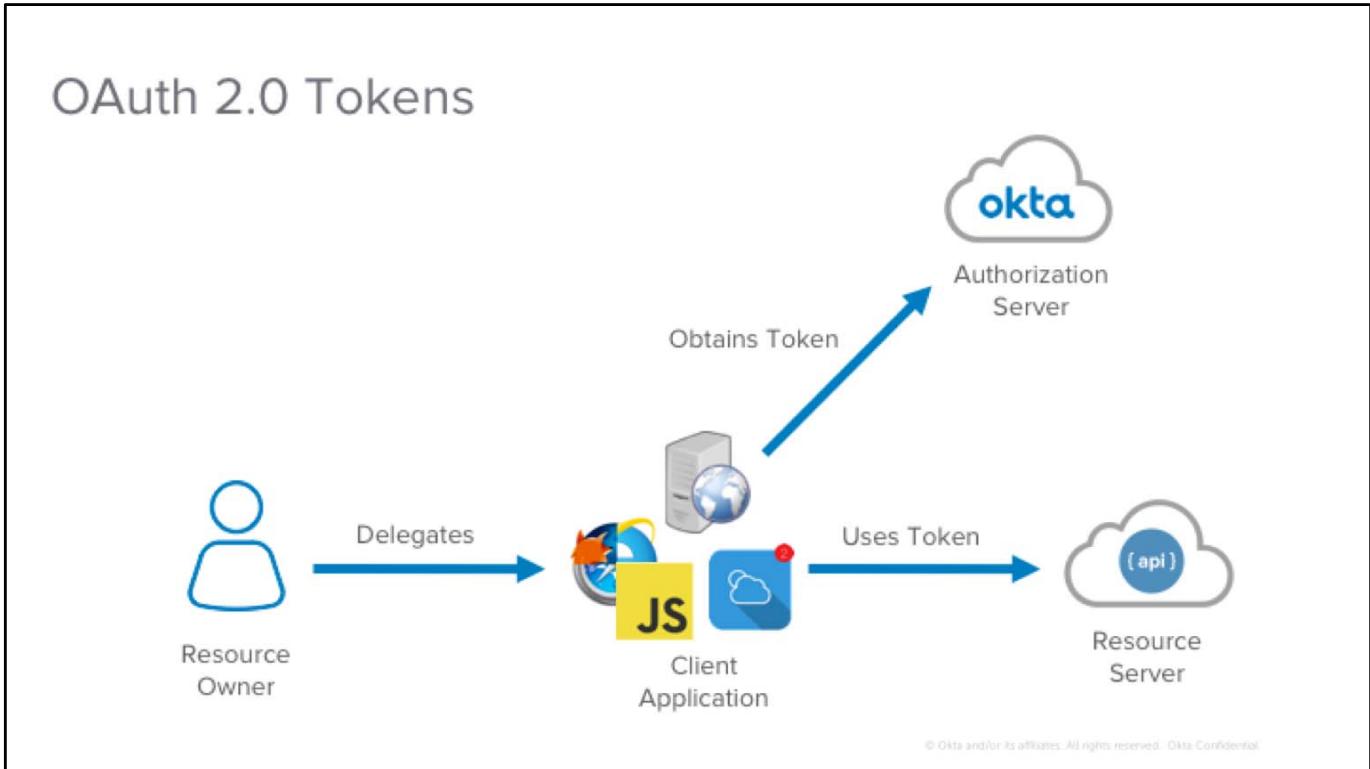
Additional Information

The client registration process produces a lot of elements which are used in the flows to provide authorized access.

The redirect URI is a security feature which is very important to limit where either the access token or the authorization code are returned to. This is used to prevent those artifacts from being sent somewhere else. This is entered in by the Resource Owner.

The Client ID is similar to a username for the client application. It is public and often passed as an HTTP parameter. This is generated by the Authorization Server.

The Client Secret is used with Trusted Clients who are able to secure it. It is similar to a password. It should never be stored in clear text, for instance a source control system. This is generated by the Authorization Server.



Additional Information

Tokens are the key to OAuth. Instead of credentials, tokens are used to provide access. The authorization server produces the tokens for the Client Applications. The Client Applications then pass the tokens as a part of requests when accessing Resource Servers.

OAuth 2.0 Tokens



Access Token

Grants temporary access to a resource to perform specific operations on behalf of a user.
Contents are not defined by OAuth.



Refresh Token

Used to obtain a new Access Token after the old one has expired.
Only available to trusted clients.



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

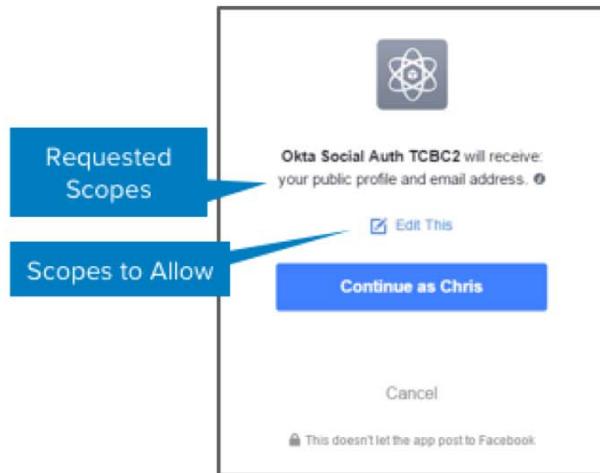
Additional Information

In OAuth 2.0, there are two types of tokens, Access Tokens and Refresh Tokens. The access token is a user-specific, temporary token passed to the Resource Server to authorize the requests. The Refresh Token is used to create new Access Tokens, from the Authorization Server, when the current Access Token is about to expire or has already expired. It is only used with Trusted Clients.

Scopes

Permissions requested by and granted to the client application to access data or perform actions at the resource server – defines the capabilities of the token

- E.g. view user profile data, modify user profile data, view contacts, make posts, create orders, access specific APIs



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

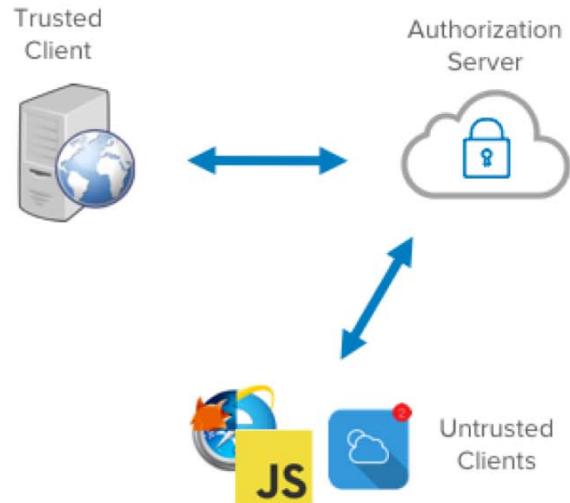
Additional Information

Scopes determine the capability of the Tokens at the Resource Server. In other words, what can the Client Application do or what data can it access in the Resource Server. The available scopes are defined by the Resource Server. In a situation where the end user is the Resource Owner, the Client Application specifies which scopes it would like to use in the request to grant access. During that process the end user, acting as the Resource Owner, decides which scopes they are willing to grant to the Client Application. In an enterprise situation like Okta, the administrator sets the granted permissions during the configuration and assignment stages.

Flows

OAuth supports many flows for different application architectures

- Server-side web flow (trusted clients): involves a secure application server
- Implicit flow for native apps (untrusted clients): mobile or desktop apps
- Implicit flow for single page apps (untrusted clients): JavaScript client apps



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

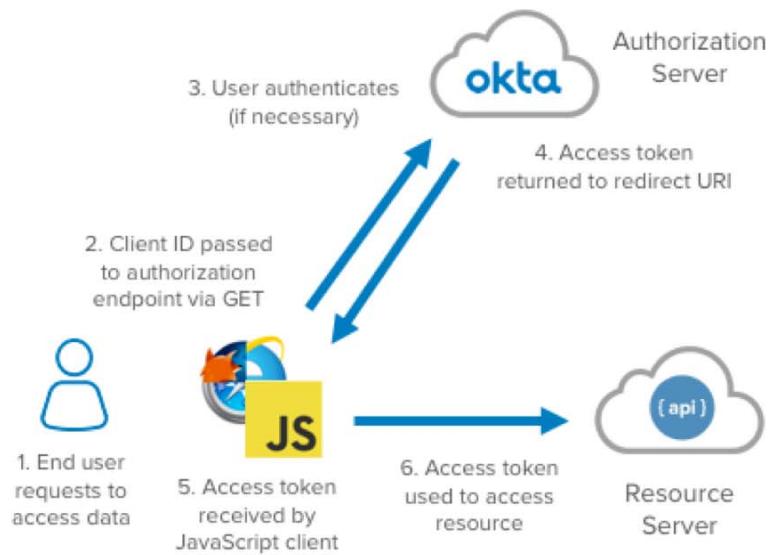
Additional Information

Flows determine the process for how tokens are requested from the Authorization Server and then obtained by the Client Application. It optionally includes interactions by the end user, who may or may not be the Resource Owner. There are many types of flows for different types of applications, over the next few slides these will be reviewed in detail. In general though, there are two types based on whether the Client Application is trusted or not.

OAuth 2.0 Implicit Flow for Single Page App

Implemented in JavaScript run from a browser context, considered an untrusted client because secrets cannot be secured

- Shorter-term access
- Lesser access (e.g. read-only)



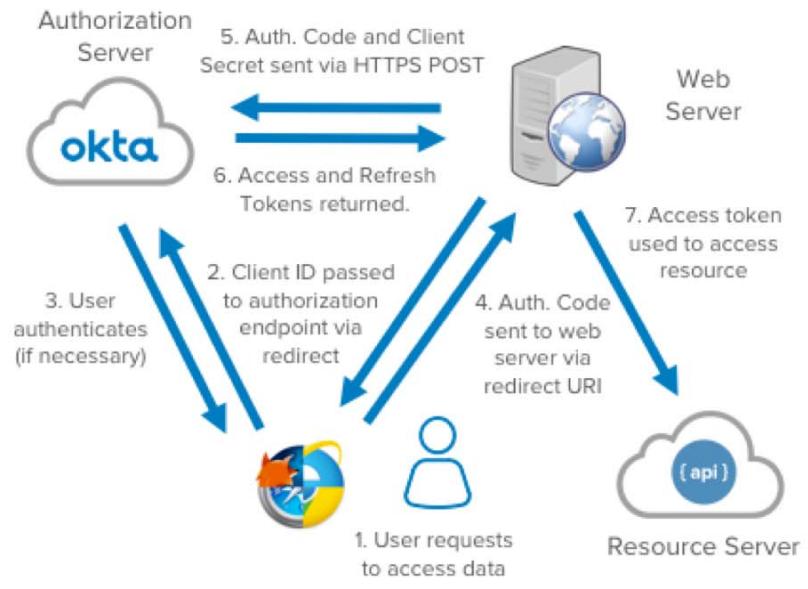
Additional Information

The OAuth 2.0 Implicit Flow for Single Page Apps allows for pure JavaScript client applications to obtain tokens. These applications are typically built with a JavaScript framework such as JQuery, Angular, or React. Because they cannot secure a Client Secret, they follow a model that is a shorter-term grant with typically a lower level of privileges issued.

OAuth 2.0 Server-side Web Flow

Considered a trusted client because secrets can be secured on the server

- Longer-term access (uses refresh tokens)
- Greater access (e.g. read/write)



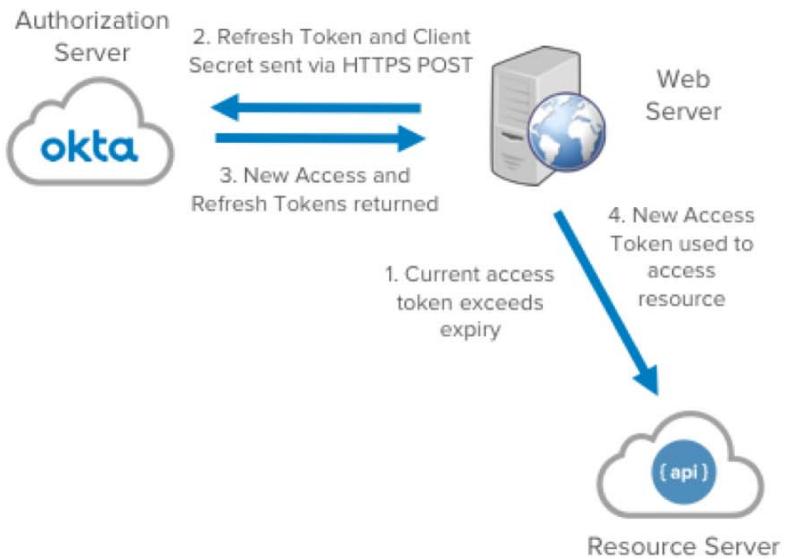
Additional Information

The OAuth 2.0 Server-side Web Flow is the most secure flow. In this case, a secured server is involved in the process. For that reason the grants are typically longer-term and with greater access privileges. On top of the elements involved in the Implicit Flow, it uses temporary one-time Authorization Codes issued by the Authorization Server. These codes are then paired with the Client Secret to request the Access Token over secured HTTPS connections. In addition, this flow can use Refresh Tokens.

OAuth 2.0 Refresh Flow

Used to request a new access token after it expires

- Returned only to trusted clients



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

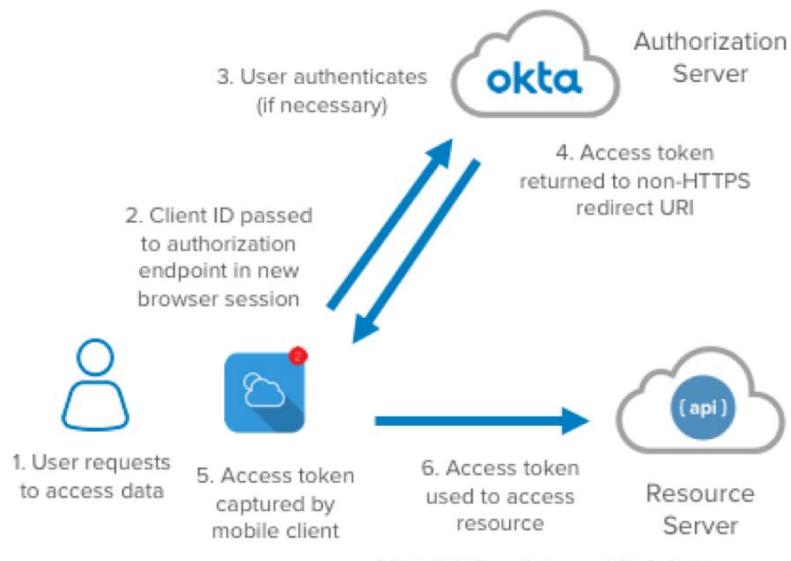
Additional Information

The Refresh Flow is used to obtain a new Access Token and Refresh Token either before or after the current Access Token has expired. This is all done over a secured (HTTPS) backend channel. This allows for much longer term grants.

OAuth 2.0 Implicit Flow for Mobile App

Example: native client needs to access data for mobile use cases

- Cannot secure a client secret but could reasonably securely store an access token
- Could be architected with auth. code/client secret for a hybrid flow



Additional Information

A native application such as a desktop app or a mobile app can use the Implicit Flow, as shown here. In this case, the mobile app would launch a browser session to obtain the Access Token. The redirect URI would not actually be a location on a web server. Instead, when the mobile app launched the browser session it would listen for the event of the redirect to the specified URI. This URL could be something like myapp://oauth. This is just one way to do OAuth with a mobile app. Sometimes a server is incorporated into a mobile app flow so that it may utilize Authorization Codes and Client Secrets.

Additional Information

Access tokens as proof of authentication: Because an authentication usually occurs ahead of the issuance of an access token, it is tempting to consider reception of an access token of any type proof that such an authentication has occurred. However, mere possession of an access token doesn't tell the client anything on its own. In OAuth, the token is designed to be opaque to the client, but in the context of a user authentication, the client needs to be able to derive some information from the token. This problem stems from the fact that the client is not the intended *audience* of the OAuth access token. Instead, it is the *authorized presenter* of that token, and the *audience* is in fact the protected resource. The protected resource is not generally going to be in a position to tell if the user is still present by the token alone, because by the very nature and design of the OAuth protocol the user will not be available on the connection between the client and protected resource. To counter this, there needs to be an artifact that is directed at the client itself. This could be done by dual-purposing the access token, defining a format that the client could parse and understand. However, because general OAuth does not define a specific format or structure for the access token itself, protocols such as the OIDC ID Token and the Facebook Connect Signed Response provide a secondary token along side the access token that communicates the authentication information directly to the client. This allows the primary access token to remain opaque to the client, just like in regular OAuth.

Different protocols for every potential identity provider

One of the biggest problems with OAuth-based identity APIs is that even when using a fully standards-compliant OAuth mechanism, different providers will inevitably implement the details of the actual identity API differently. For example, a user's identifier might be found in a `user_id` field in one provider but in the `subject` field in another provider. Even though these are semantically equivalent, they would require two separate code paths to process. In other words, while the authorization may happen the same way at each provider, the conveyance of the authentication information could be different. This problem can be mitigated by providers using a standard *authentication protocol* built on top of OAuth so that no matter where the identity information is coming from, it is transmitted in the same way.

Lack of audience restriction

Another problem with trading the access token for a set of attributes to get the current user is that most OAuth APIs do not provide any mechanism of audience restriction for the returned information. In other words, it is very possible to take a naive client, hand it the (valid) token from another client, and have the naive client treat this as a "log in" event. After all, the token is valid and the call to the API will return valid user information. The problem is of course that the user hasn't done anything to prove that they're present, and in this case they haven't even authorized the naive client.

This problem can be mitigated by communicating the authentication information to a client along with an identifier that the client can recognize and validate, allowing the client to differentiate between an authentication for itself versus an authentication for another application. It is also mitigated by passing the set of authentication information directly to the client during the OAuth process instead of through a secondary mechanism such as an OAuth protected API, preventing a client from having an unknown and untrusted set of information injected later in the process.

OIDC

Conceptually...

- An assertion of user authentication
- Authorized identity information for that user

Okta OIDC Use Cases...

- Social Authentication
- Single Sign-on
- API Access Management



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

OIDC Capabilities

- Extends OAuth 2.0
- Defines signed id_token for the client
- Defines UserInfo endpoint to fetch user attributes
- Provides a standard set of scopes and claims for identity profiles
- Built-in registration, discovery and metadata for dynamic federations (BYOI)
- Supports high assurance levels and key SAML scenarios (enterprise)



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Because it's an open standard, OIDC can be implemented by anyone without restriction or intellectual property concerns. OIDC is built directly on OAuth 2.0 and in most cases is deployed right along with (or on top of) an OAuth infrastructure. OIDC also uses the JSON Object Signing And Encryption (JOSE) suite of specifications for carrying signed and encrypted information around in different places. In fact, an OAuth 2.0 deployment with JOSE capabilities is already a long way to defining a fully compliant OIDC system, and the delta between the two is relatively small. But that delta makes a big difference, and OIDC manages to avoid many of the pitfalls discussed above by adding several key components to the OAuth base:

ID Tokens

Identity Token

- An implementation of JSON Web Tokens (JWTs)
- The following, base64 encoded, separated by periods
 - Header
 - Identifies algorithm
 - Payload
 - Contains claims – information about the user
 - Signature
 - For authentication and integrity

hhh.ppp.sss

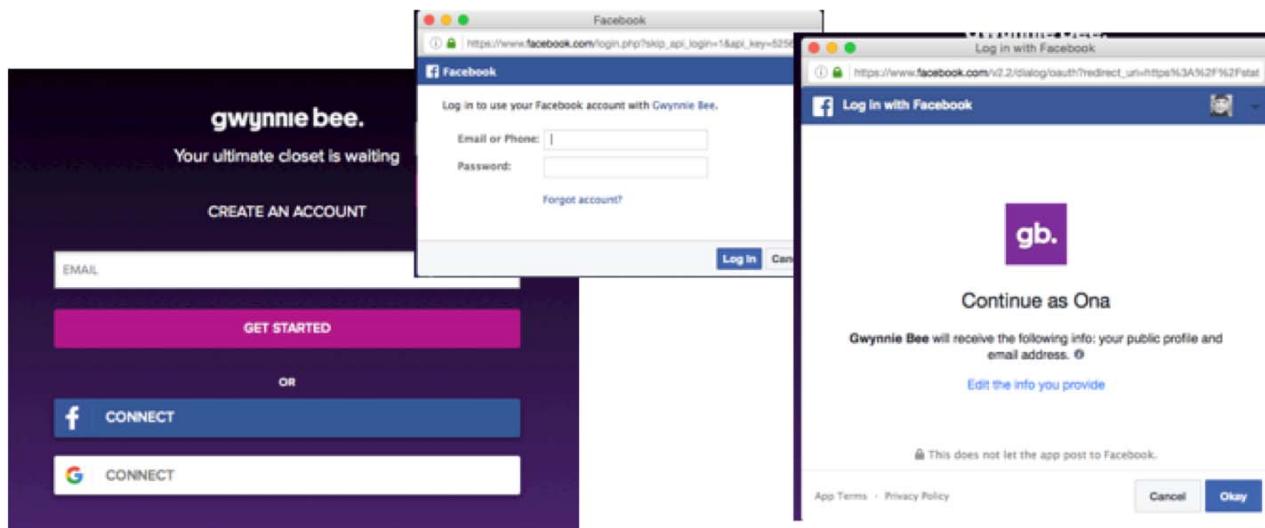
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

ID Token Claims

- Always present:
 - **iss**: the issuer, e.g. <https://oktaice000.okta.com>
 - **sub**: the Okta user id, e.g. 00uid20ijfiowje
 - **aud**: the audience, the app's Client Id
 - **exp**: expiration timestamp, in epoch time, 1 hour lifetime
 - **nonce**: one-time token to prevent replay attacks
- Scope-dependent:
 - **name**: the user's full name
 - **preferred_username**: Okta username
 - **email**: primary email address
- Customizable

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Social Authentication



Additional Information

Lets take a step back and look at the consumer world first. We all have probably seen sign in screens like this – where you go to a web page and it says "Sign up with Facebook" or "Connect with Facebook" – If we think about this – Facebook is a great Identity Provider for the consumer world – Lots of identity information is stored in Facebook and the consumer site can gain access to this via an authentication request but it is more than just an authentication request – the next step is authorizing access to the user information on Facebook. At first glance this seems like a dream come to true for an Enterprise solution where you need consumers to login into your application – secure, easy to use access – Everyone knows their Facebook credentials. Reducing friction from your awesome application to consumers logging in – that sounds like a win win to me.

Facebook, Twitter and LinkedIn have made this framework famous in the consumer world, how can we take this model and apply it to the Enterprise world? To do that we need to look at the underlying framework.

Social Authentication Scenarios

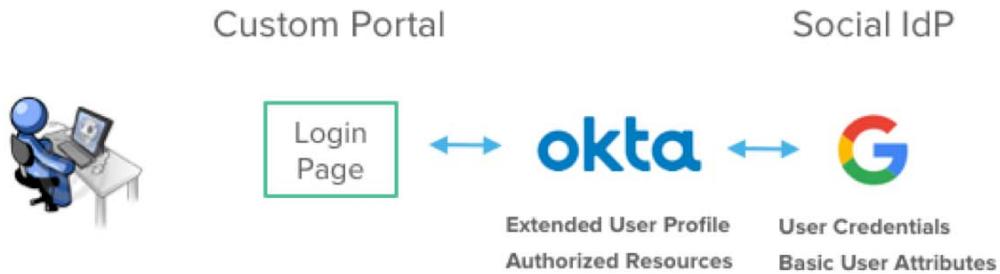


© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

Okta currently works with 4 social authentication providers: Google, Microsoft, Facebook and LinkedIn. The connection to integrating with Okta can be as simple as generating a URL.

Social Authentication: Responsibilities



Additional Information

The role of the Social Identity Provider is to store the user credentials for authentication, and basic user attributes to initialize a profile in Okta.

Some of the key roles of Okta are to extend the user profile by tracking additional attributes relevant to the business, and provide access to authorized resources.

Social Auth: User Registration

1. Matching user in Okta?
↓ No.
2. Redirect User to Social IdP via Okta.
↓
3. User authenticates at Social IdP.
↓
4. User authorizes social IdP to share selected information.
↓
5. ID Token returned to Okta, user provisioned and linked in Okta.



Additional Information

This is the flow of a user getting provisioned from the social identity provider. The first step is optional, but typically a custom portal provides the ability to lookup based on email or phone to see if the user already has an account.

Social Auth: Authentication

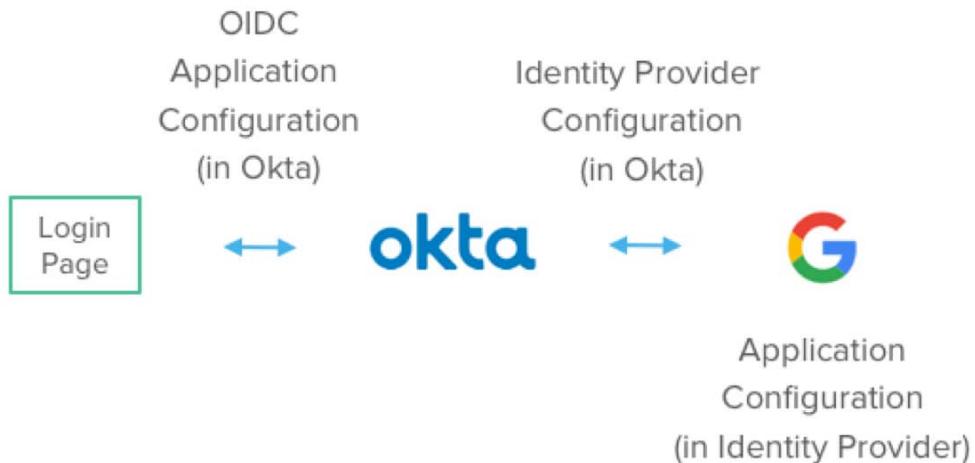
1. User indicates they wish to sign in with social IdP.
2. User redirected to social IdP.
2. User authenticates at social IdP.
3. ID Token returned to Okta
4. User redirected back to app with Okta session established.



Additional Information

A social authentication scenario.

SSO with OIDC and the Auth SDK



Additional Information

There are three different areas to configure: at the IdP, defining the OIDC client in Okta, and defining the connection to the IdP from Okta.

Identity Provider Configuration (in Okta): User Matching

- **IdP Username:** Which claim to use to generate the username in Okta
 - Defaults to email.
 - Supports Okta Expression Language
- **Match against:** Which field(s) in Okta to use to locate an existing user
 - Username
 - Email
 - Username or email

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Identity Provider Configuration (in Okta): User Linking

- **Account Link Policy:** Rules for binding IdP users to [existing users](#) in Okta
 - Automatic
 - Call-out
 - Disabled
- **Auto-Link Restrictions:** Limits linking to existing users based on their Group membership

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Identity Provider Configuration (in Okta): JIT User Provisioning

- **Provisioning Policy:** determines whether to create [new users](#) in Okta
 - Automatic
 - Call-out
 - Disabled
- **Profile Master:** whether to use the IdP as-the-master (apply updates).
- **Group Assignments:** initial Group memberships during user create operation

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Link to Okta from Custom Login Page

- Base URL: <https://myorg.oktapreview.com/oauth2/v1/authorize?...>
- Parameters:
 - **idp**: unique identifier for the IdP configuration in Okta
 - **client_id**: unique identifier for the client application configuration in Okta
 - **scope**: OAuth scopes passed to IdP which determine which claims are requested. "openid" is required to return the ID Token.
 - **response_type**: indicates which OAuth flow to use and whether the OAuth Access Token is returned.
 - **redirect_uri**: redirect URI for client app. Must match value in Okta.
 - **state**: one-time value to prevent replay attacks. Used by Okta.
 - **nonce**: one-time value to prevent replay attacks. Used by IdP.

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

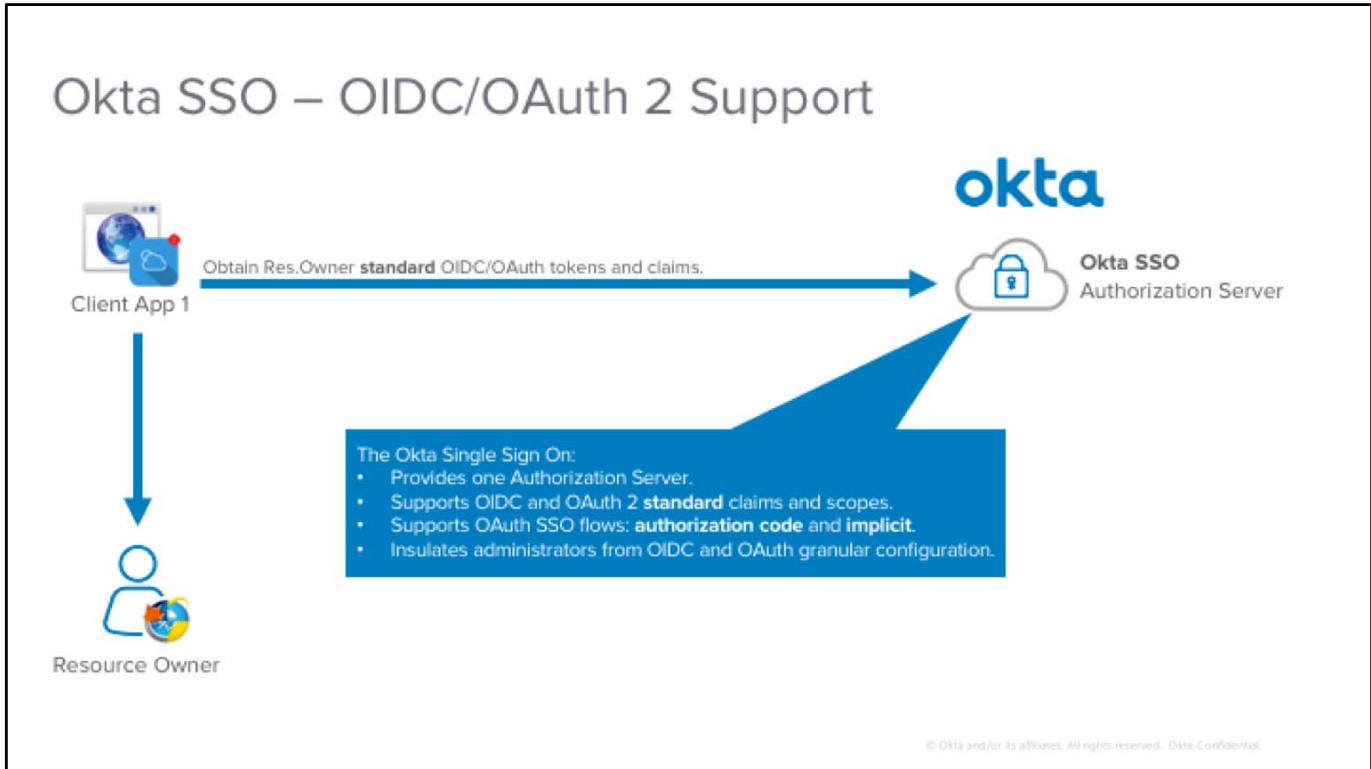
Additional Information

The link in the endpoint app initiating the social authentication process must have multiple parts.



Implement Social Authentication with Facebook

- Configure the Okta Sign-in Widget
- Enable CORS
- Configure and Test Social Auth



Additional Information

The Okta SSO:

- Provides a single Authorization Server under the Okta org url:
<https://oktaorg.okta.com/oauth2/v1/>
- Supports the OIDC and OAuth 2 **standard** claims and scopes.
- Supports OAuth SSO flows:
authorization code: For authenticating trusted applications.
implicit: For authenticating SPA and Mobile Apps.
- Insulates administrators from OIDC and OAuth granular configuration.

OIDC vs. SAML

- OIDC
 - Lightweight
 - Easy to implement
 - Works with single page apps (SPAs), native apps, and form-based web apps
- SAML
 - Heavyweight
 - Hard to implement
 - For form-based web apps

© Okta and/or its affiliates. All rights reserved. Okta Confidential

SSO with OIDC

1. Does the user have a local (app) session or valid ID Token?

↓ No.

2. Does the user have a session with Okta?

↓ Yes.

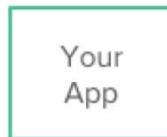
3. App requests ID Token for user.

↓

4. App validates ID Token.

↓

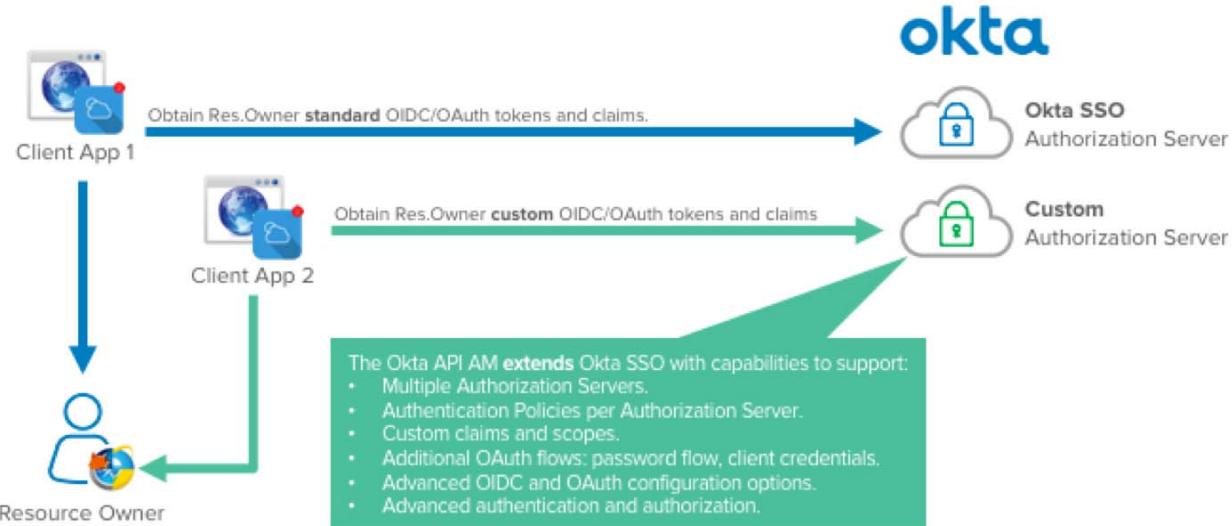
5. Establish local (app) session or store ID Token in local storage.



User Credentials
Extended User Profile
Authorized Resources



Okta API AM – OIDC/OAuth 2 Support



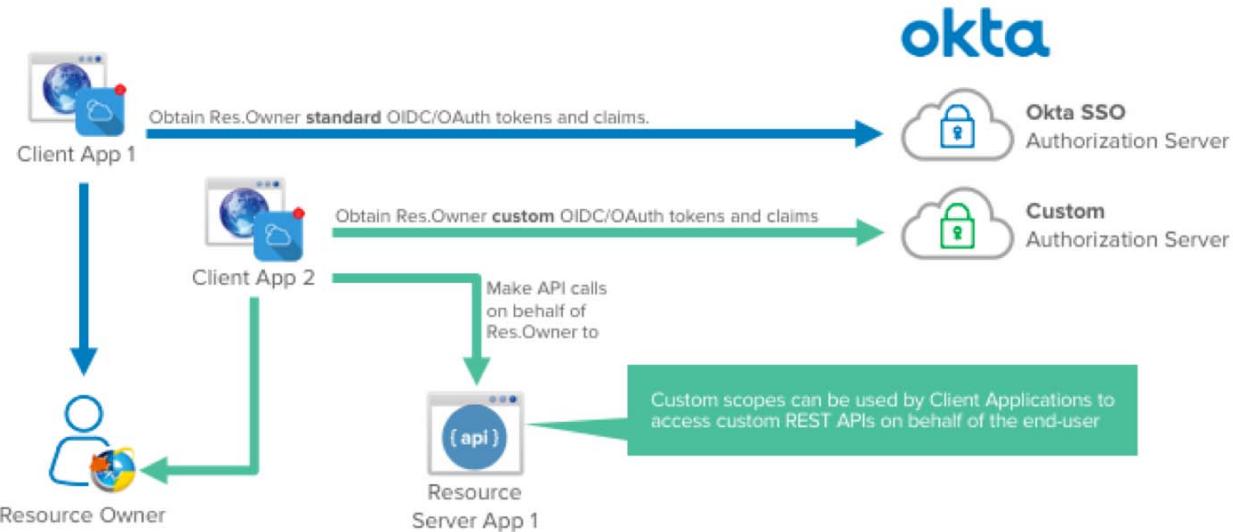
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Additional Information

The Okta API AM **extends** Okta SSO with additional capabilities to support:

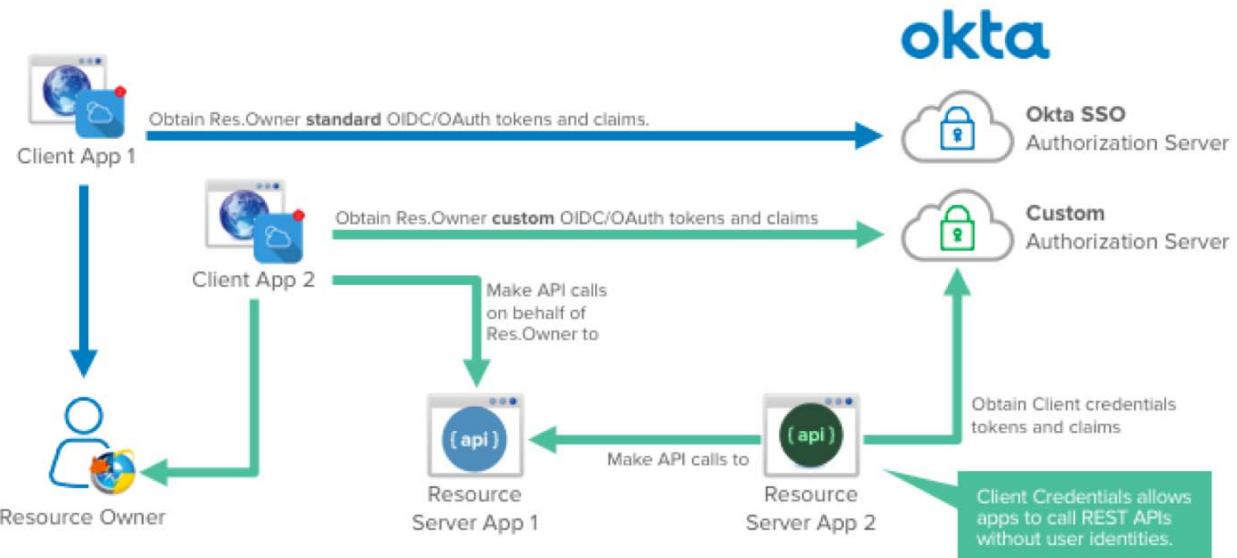
- **Multiple Authorization Servers:** Administrators can configure multiple OAuth/OIDC authorization servers. Each authorization server can have its own authentication and authorization configurations and policies.
- **Authentication Policies:** Administrators can setup policies to control the access per authorization server.
- **Custom claims and scopes:** Administrators can create custom claims and scopes.
 - **Custom claims** provide information about the end-user accessing the application (resource owner). The information comes from the Universal Directory and can be customized using the Okta UD expression language and regex.
 - **Custom scopes** are used by the Client Application to authorize the access to APIs in other applications (Resource Servers) on behalf of the Resource Owner.
- **Additional OAuth flows:**
 - The **password flow** is an OAuth 1.0 flow, used for deprecated implementations. This flow requires the client application to collect the resource owner credentials, which may lead to security concerns.
 - The **client credentials** is an OAuth 1.0 flow used for app-to-app communications. It's used when one client app needs to perform requests in a resource server app without having a user identity. Examples: B2B, batch processes, monitoring requests, and webhooks.
- **Advanced OIDC and OAuth configuration options and Authentication and authorization for additional scenarios:** the API AM supports advanced configuration that's not supported by Okta SSO. These configurations are used in new applications and architectures, such as microservices.

Okta API AM – OIDC/OAuth 2 Support



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Okta API AM – OIDC/OAuth 2 Support



Okta SSO vs Okta API AM

	Feature	SSO	API AM
OIDC	SSO	✓	✓
	ID Token	✓	✓
	Standard Claims	✓	✓
	Custom Claims		✓

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Okta SSO vs Okta API AM

	Feature	SSO	API AM
OAuth flows	Resource Owner	✓	✓
	Implicit	✓	✓
	Client Credentials		✓
	Password		✓
OAuth Entities	Resource Owner	✓	✓
	Client App	✓	✓
	Default Authorization Server	✓	✓
	Multiple Authorization Servers		✓
	Resource Servers		✓

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Okta SSO vs Okta API AM

	Feature	SSO	API AM
OAuth features	Access Token (JWT)	✓	✓
	Refresh Token	✓	✓
	Custom Scopes		✓
	Policies per Authorization Server		✓
	App-to-App Security		✓

© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Things you can do with API AM that you cannot do with Okta SSO

Define custom claims using regex and Okta Expression Language.

Support password flow (OAuth 1.0 apps)

Define custom tokens

Protect REST APIs

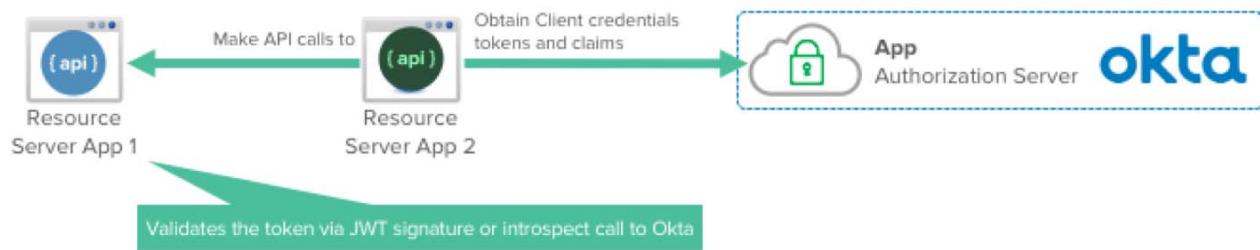
Secure APP-to-APP connections

Secure IoT connections

Secure Microservice connections

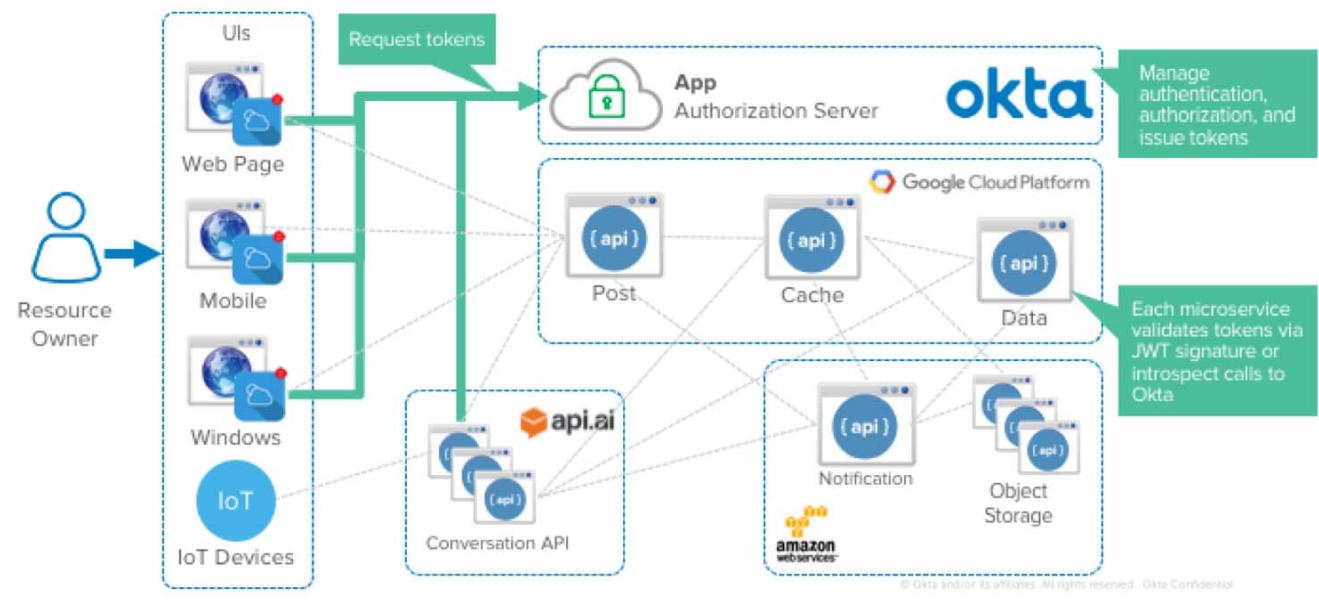
© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Okta API AM – Securing APP-to-APP



© Okta and/or its affiliates. All rights reserved. Okta Confidential.

Okta API AM – Securing IoT and Microservices





The background of the slide features a photograph of a person sitting at a desk in an office environment, working on a laptop. A computer monitor in front of them displays the text "Lab 10-2". The entire image is overlaid with a solid blue tint.

Create an OIDC Application Using the AIW



Test the OIDC SSO

- Launch and Configure Postman
- Get OIDC
- Obtain and Test the Access Token



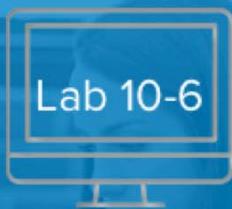
Configure API AM

- Register the Service Application, Auth Server, Scopes, and Claims



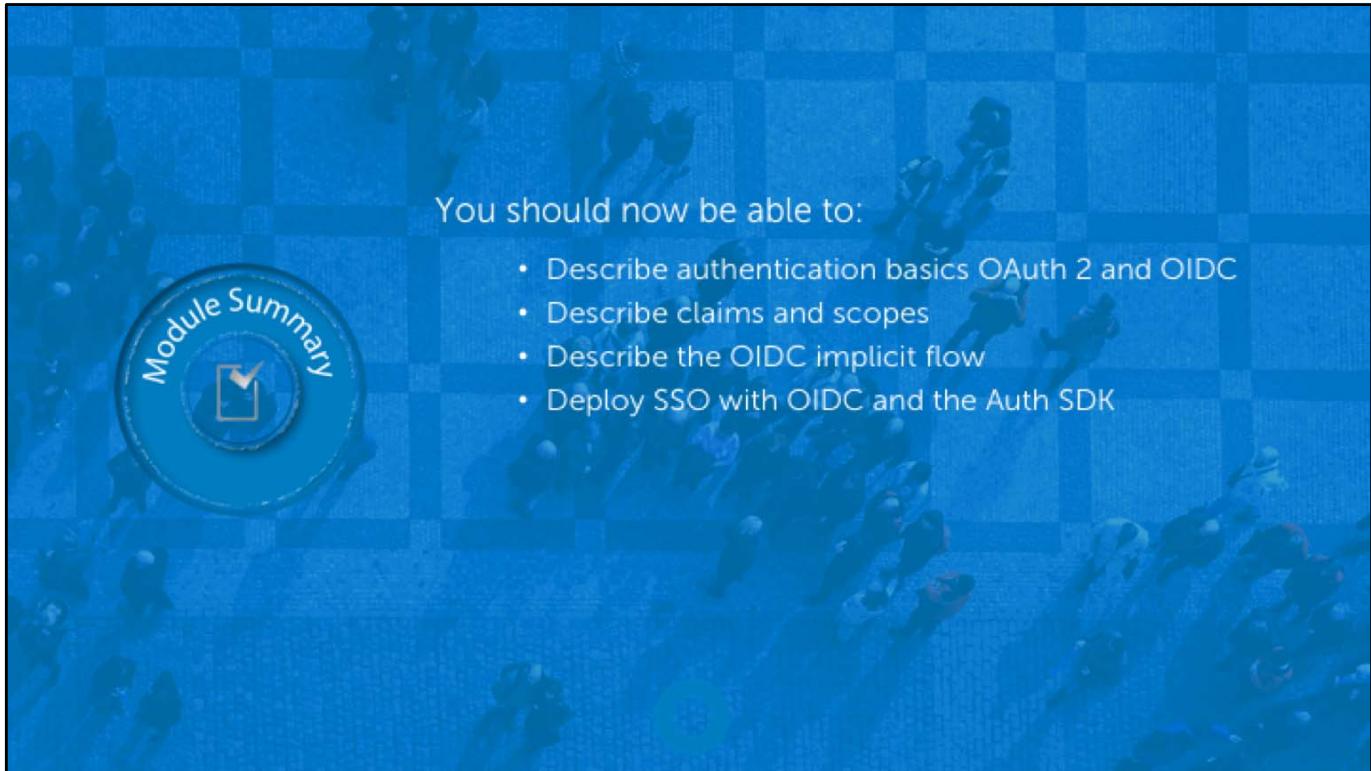
Test API AM Requests

- Update Environment Variables
- Get OIDC and Test



Enable the Development Team

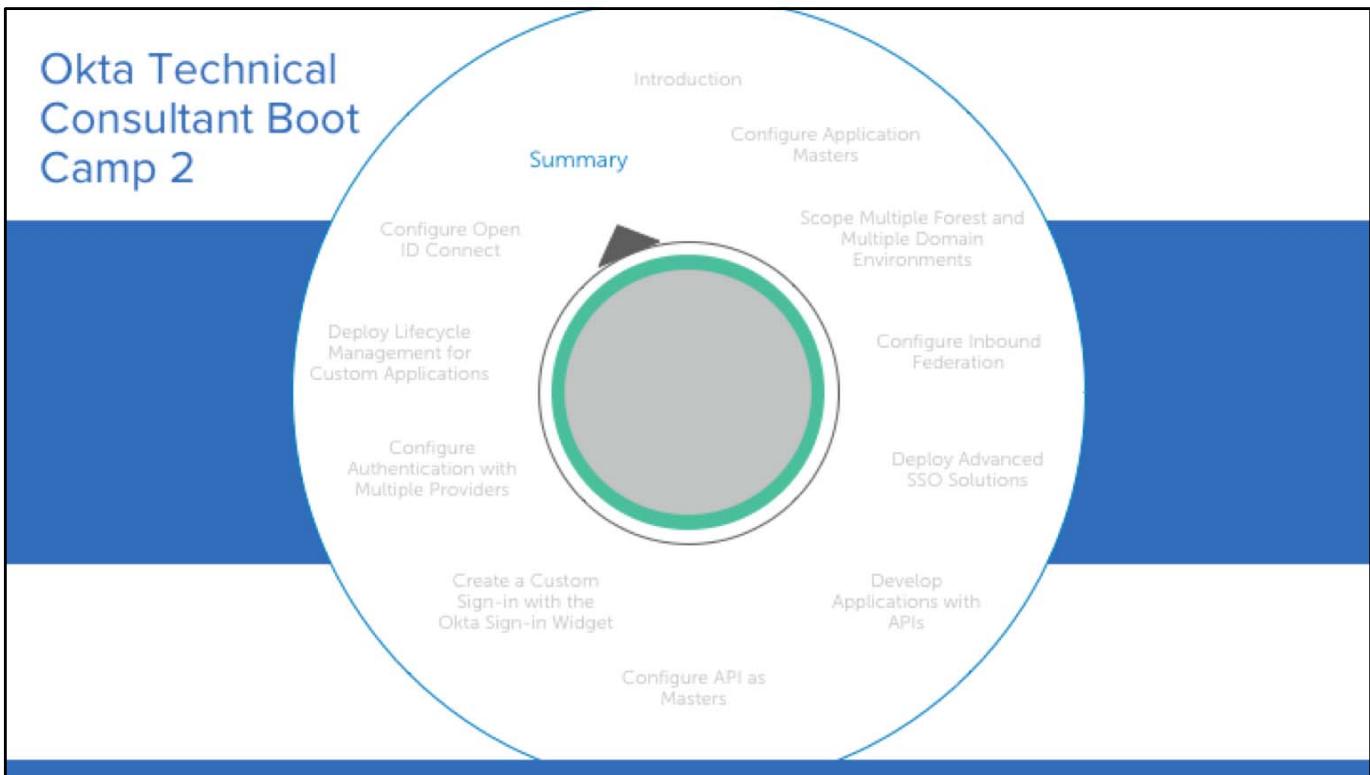
- Work with Postman
- Bookmark Assets
- Explore a Code Sample



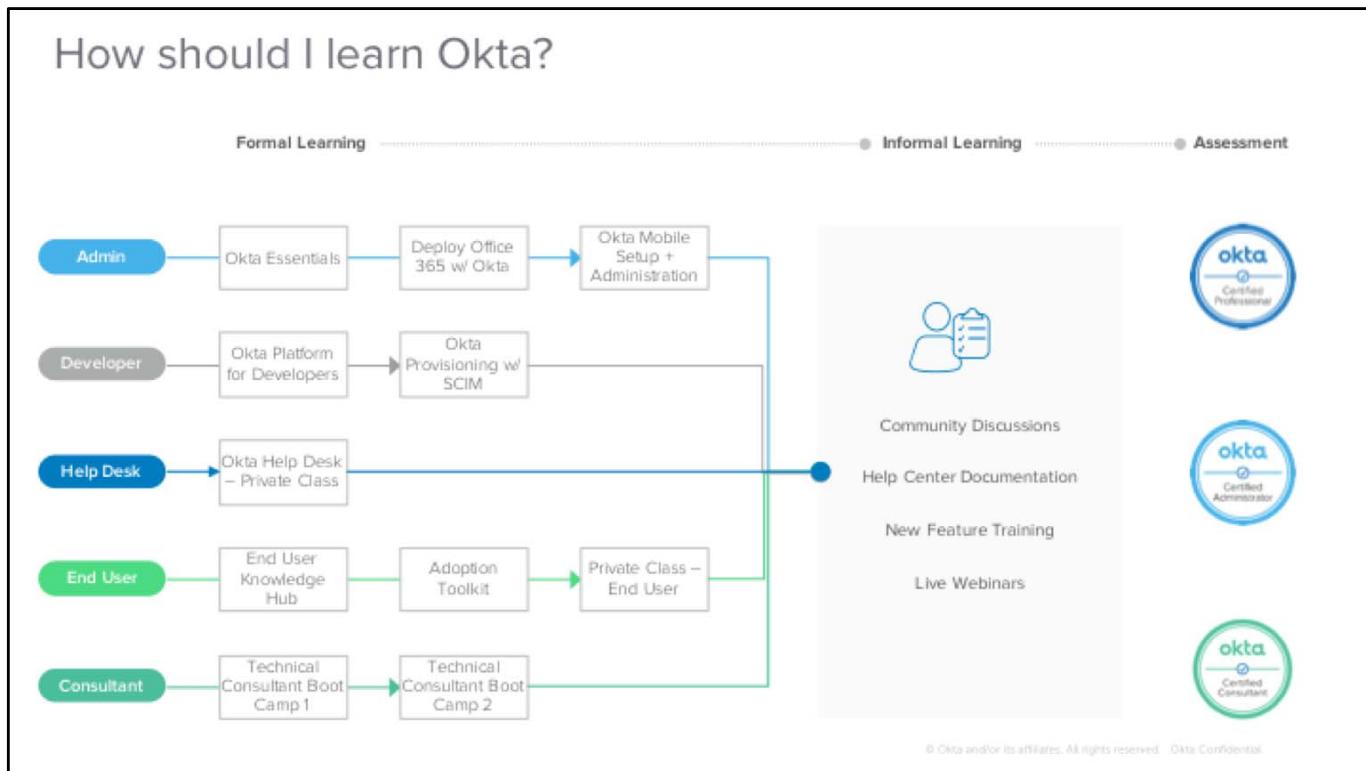
You should now be able to:

- Describe authentication basics OAuth 2 and OIDC
- Describe claims and scopes
- Describe the OIDC implicit flow
- Deploy SSO with OIDC and the Auth SDK

Module Summary



How should I learn Okta?





okta Certification

Promotion:

Partners who attend the Okta Technical Consultant Boot Camp course beginning February 15 and until further notice, can take the Okta Professional Exam for 50% off the standard market price. Eligible partners must register for and sit the Okta Professional Exam within 30 days of completing the boot camp course.

© Okta and/or its affiliates. All rights reserved. - Okta Confidential

