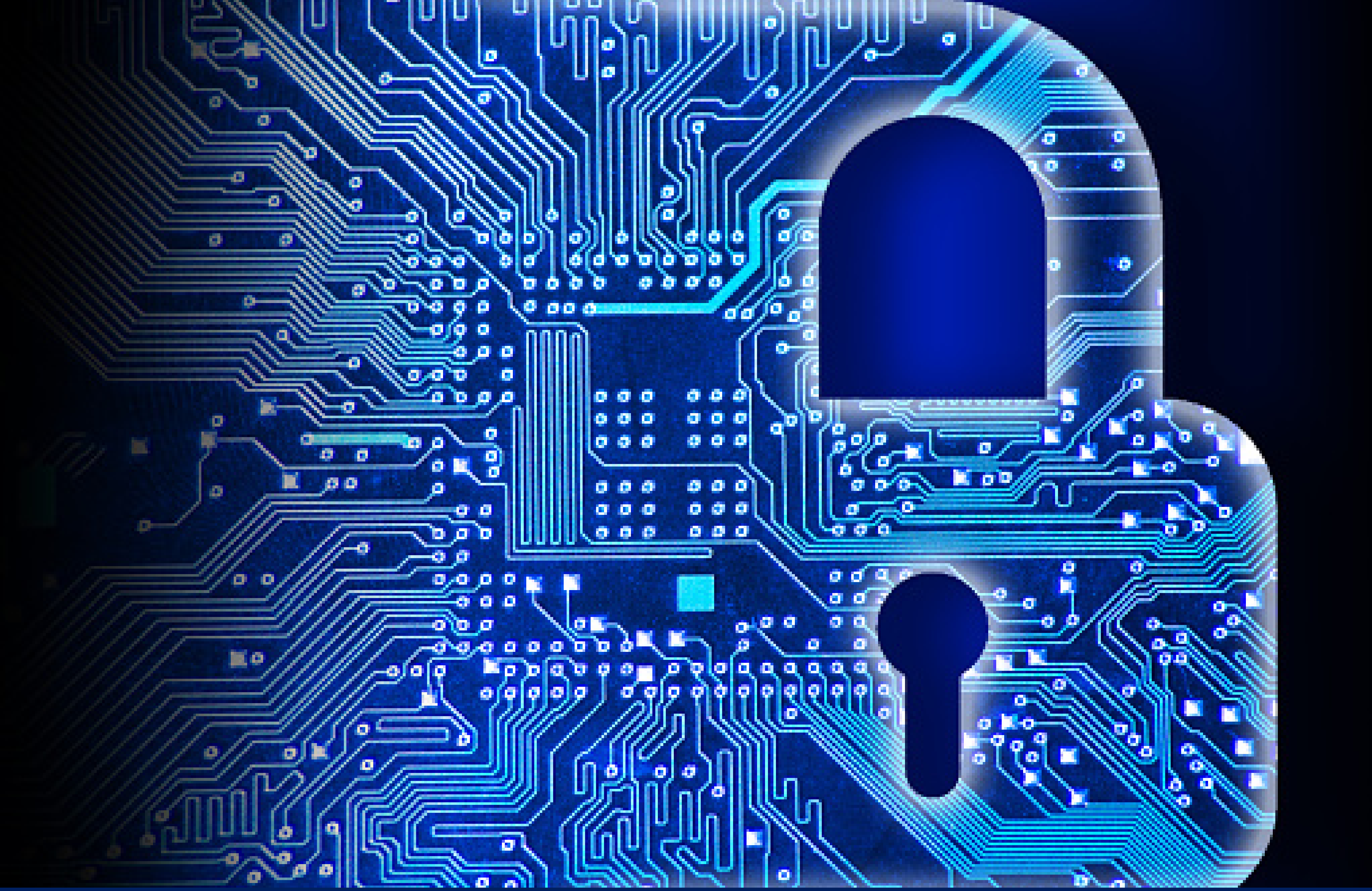


Identity, Credential, and Access Management (ICAM) Implementation Guidance

Science and Technology Directorate

Homeland Security

Science and Technology

Primary Authors

Christine Owen

Larry Kroll

Chris Price

Nyleena Roberts

Contributing Organizations

Oasys International Corporation

Department of Homeland Security (DHS) Science and Technology Directorate

Partner Engagement-Information Sharing Environment (PE-ISE)

DHS Office of Emergency Communications (OEC)

First Responder Network Authority (FirstNet)

Federal Communication Commission (FCC)

National Institute for Science and Technology Public Safety Communications Research Division (NIST PSCR)

PUBLIC SAFETY COMMUNICATIONS
IDENTITY, CREDENTIAL, AND ACCESS MANAGEMENT
WORKING GROUP

Identity, Credential, and Access Management (ICAM) Implementation Guidance

February 2019

Version 2

Dedicated to the memory of:

Tom Sorley
1965-2018

The Identity, Credential, and Access Management (ICAM) document series is dedicated to the memory of Tom Sorley. Tom was a member of the executive leadership of the Public Safety Communications ICAM Working Group, which sponsored this document. He was the Chief Information Officer and Deputy Director of the Information Technology Department for Public Safety for the City of Houston, Texas, and National Chair of the Public Safety Advisory Committee (PSAC). Tom was a thought leader in public safety communications and his vision is reflected in this ICAM Educational Series.

DISCLAIMER OF LIABILITY

The Identity, Credential, and Access Management (ICAM) Educational Series is provided by the Public Safety Communications ICAM Working Group (PSC ICAM WG) “as is” with no warranty of any kind, either expressed or implied, including, but not limited to, any warranty of merchantability or fitness for a particular purpose. This material is provided to support the efforts of public safety information sharing, situational awareness and key decision making. These documents are intended to guide users for making informed decisions on improving the security posture of their information systems by using ICAM principles.

The ICAM Educational Series is intended to provide guidance for implementing ICAM principles, and does not contain or infer any official requirements, policies, or procedures, nor does it supersede any existing official emergency operations planning guidance or requirements documents. As a condition of the use of the Series, the recipient agrees that in no event shall the United States Government or its contractors or subcontractors be liable for any damages, including but not limited to, direct, indirect, special or consequential damages, arising out of, resulting from, or in any way connected to the Series or the use of information from the Series for any purpose. It is recommended that organizations align their resources with tools that would best fit their infrastructure as well as their own standards and requirements.

The PSC ICAM WG, and DHS S&T does not endorse any commercial product or service referenced in the ICAM Educational Series, either explicitly or implicitly. Any reference herein to any specific commercial product, process, or service by service mark, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement, recommendation, or favoring by the PSC ICAM WG. The views and opinions of authors expressed herein do not necessarily state or reflect those of the PSC ICAM WG and shall not be used for advertising or product endorsement purposes.

EXECUTIVE SUMMARY

The Department of Homeland Security (DHS) Science and Technology Directorate (S&T) created systems by implementing Identity, Credential, and Access Management (ICAM) products within a sandbox for the Public Safety Communications (PSC) ICAM Working Group (WG). This document provides Implementation Guides for Public Safety Community (Community) solution architects to enhance existing ICAM efforts. These Implementation Guides provide instructions for building individual ICAM components within an organization's system.

The Community's goal is to have the ability to appropriately share critical information among its members, which can aid in saving lives and protecting property. This critical information is at times highly sensitive and can include law enforcement information, as well as personally identifiable information (PII) and protected health information, so each organization must have assurance the right person with the right credentials is accessing information at the right time. This is where ICAM comes in.

To support the Community with its mission-critical tasks, ICAM addresses the growing data management, interoperability and cybersecurity challenges facing public safety today. ICAM solutions, especially federated ones, align public safety communities around common identity and access management practices.

It is essential for important for Community members who are sharing information between different organizations to make sure the information does not fall into the wrong person's hands. ICAM provides identity proofing for an organization's employees and volunteers, providing strong credentials for system access and enabling the use of multifactor authentication, using attributes to provision resources, and creating strong access management all help an organization ensure that the right person is accessing an organization's information through a secure and seamless federation.

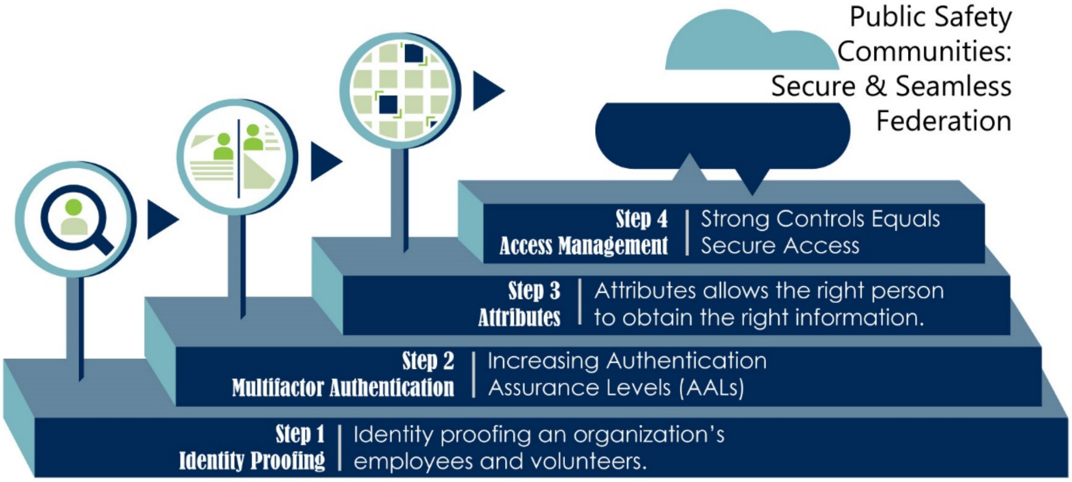

Public Safety Communities: Secure & Seamless Federation
Step 4 Access Management | Strong Controls Equals Secure Access
Step 3 Attributes | Attributes allows the right person to obtain the right information.
Step 2 Multifactor Authentication | Increasing Authentication Assurance Levels (AALs)
Step 1 Identity Proofing | Identity proofing an organization's employees and volunteers.

Figure 1 - Steps to Secure and Seamless Information Sharing (Federation)

The intent of this document is to provide implementation guides for ICAM-enabled systems. This document does not recommend specific ICAM products, but does strive to create useable aids for the implementation of ICAM products and encourage the adoption of multifactor authentication with a focus on the use of open source products. This document is paired with the ICAM Executive Primer, ICAM Acquisition Guidance and ICAM Federation System Checklist.

Intended Audience

This document provides guidance to implement ICAM products. The document teaches each of the following stakeholders[1] about ICAM solutions:

Stakeholder Group	Responsibilities	Documents to Read
Executive Leadership	…is the responsible authority for the department, state or agency's fiscal and human resources for ICAM investments. This stakeholder group will use the document to understand the importance of ICAM investments, and to translate the value proposition of ICAM solutions to their mission needs.	ICAM Executive Primer
Program Managers	…are responsible for the operational implementation and oversight of ICAM capabilities to ensure they meet the functional mission requirements defined by the intended users. They must communicate to both the executive leadership and solutions architects to ensure understanding and expectations of the requirements for interoperable ICAM investments. Managers are required to quantify the benefit and resource impacts, including cost and integration savings, to executive leadership to ensure continued support and resource sustainment. This document provides program managers with a description of the key capabilities, processes, services, infrastructure, standards and procurement language samples that are required of an interoperable ICAM architecture solution.	ICAM Executive Primer, ICAM Acquisition Guidance & ICAM Federation System Checklist
Solution Architects	…are responsible for acquisition requirements and the design/development/integration of ICAM solutions in accordance with their respective organization's enterprise architecture technical and management requirements. The solution architects will be required to compare and quantify the technical implementation options, alternatives and cost constraints to the program managers. This document provides structured technical guidance and reference artifacts to assist in achieving an ICAM-enabled system.	This document, ICAM Executive Primer, ICAM Acquisition Guidance & ICAM Federation System Checklist

[1] Each stakeholder's responsibilities were adopted from the Information Sharing Environment Geospatial Interoperability Reference Architecture (GIRA), available at https://www.dni.gov/files/ISE/documents/DocumentLibrary/GIRA.pdf.

CONTENTS

TABLE OF FIGURES

1 INTRODUCTION

This document serves as a source of Identity, Credential, and Access Management (ICAM) Implementation Guides resulting from the evaluation of publicly available documents and the implementation of ICAM-enabled systems in a sandbox. The ICAM landscape is complex and there are many elements to consider, particularly when determining organizational policies and system interoperability. ICAM policies are important to have in enabling technology to share data within a wide variety of applications, including an organization's existing legacy systems as well as emerging nationwide initiatives, such as the Nationwide Public Safety Broadband Network (NPSBN), Next Generation 911 services and the First Responder Network Authority (FirstNet).

While this document is focused on assisting state, local, tribal and territorial (SLTT) Public Safety Community (Community) entities in improving their system's security posture so they can safely and securely share information with each other, this document can be used by any organization implementing the products discussed within this document. Instead of considering both single and multifactor authentication, this document focuses on implementing multifactor authentication on an organization's systems. Adopting multifactor authentication when upgrading an ICAM-enabled system is highly recommended for any organization, but it is especially recommended based on a risk assessment of the typical information stored in the Community's systems. This document is the third of a series of ICAM educational tools, including the ICAM Executive Primer, ICAM Acquisition Guidance and ICAM Federation System Checklist.

1.1 PURPOSE

The goal of this document is to enable any organization, including the SLTT Community, to spend its resources wisely on thoughtful and well-specified ICAM products to support information sharing and fortified authentication capabilities. This document provides ICAM implementation guides for the Community and seeks to leverage existing efforts to avoid duplication and maximize the value of existing ICAM products and solutions. While it does use specific vendors to complete the builds, this document does not endorse any specific vendor, technology or software. The implementation guidance within this document aids the Community in creating ICAM-enabled systems and adopting multifactor authentication. Further, each organization is responsible for its own cybersecurity measures, and they should make decisions based on their own requirements, laws and expertise on the subject.

1.2 BACKGROUND

The Public Safety Communications Identity, Credential, and Access Management Working Group (PSC ICAM WG) is a subsidiary group of the Information Sharing Council (ISC) with a Federal Advisory Committee Act (FACA) exempt status under Section 1016(g)(4) of the Intelligence Reform and Terrorism Prevention Act of 2004 (as amended). Its members include the Department of Homeland Security (DHS) Science and Technology Directorate (S&T), Partner Engagement Information Sharing Environment (PE-ISE), DHS Office of Emergency Communications (OEC), FirstNet, as well as National Institute of Standards and Technology (NIST) Public Safety Communications Research Division (PSCR). The PSC ICAM WG supports the ISC in fulfilling the ISC's duties (with a focus on public safety) pertaining to the interchange of information between public safety agencies by addressing policy, governance, standards, technology and acquisition guidance on ICAM capabilities for the Community.

1.3 APPROACH

Through outreach to various communities (including FirstNet stakeholders, the Federal ICAM Subcommittee, DHS Cybersecurity and the Office of the Director of National Intelligence's Sensitive But Unclassified [SBU] Technical Advisory Committee [STAC]) and utilization of the capabilities at the DHS S&T, the PSC ICAM WG gathered existing ICAM-related documents, performed research and collected information. This information provided the background for building ICAM-enabled systems in a test environment (i.e., sandbox), the result of which was distilled into this implementation guide.

1.4 ENABLING MULTIFACTOR AUTHENTICATION

ICAM concepts appear in everyday life. When leaving our homes or cars, most of us choose to lock our doors and restrict access only to those with a key.[2] Likewise, anyone who accesses the internet creates a username and password to access a computer system, email or social network, which are examples of *single factor authentication*.

Most people only use a username and password when accessing computer systems, likely due to a perceived notion of convenience. Single factor authentication is a weaker form of protection for computer systems and applications. A username and password combination or pin number are examples of single factor authentication when they are not used on their own. While single factor authentication is simple and easy to implement, the practice is akin to leaving a door unlocked for hackers. Since passwords are easily obtained via email phishing, single factor credentials played a factor in 81 percent of the systems that were hacked in 2016.[3]

Multifactor authentication uses a combination of credentials to provide higher assurance that the individual attempting to access a protected resource is that individual. To create this higher assurance, multifactor authentication requires the use of two of three "factors." The factors include something you have (a credential), something you are (fingerprint or other biometric) and something you know (password or personal identification number ([PIN]). Information breaches can be prevented by using multifactor authentication to access your data. Implementing simple multifactor authentication methods can help organizations prevent costly, time-consuming attacks.

When organizations connect their systems with others within their community of interest, the organizations will need to know who is accessing which pieces of information at what times to ensure only authorized users are within the system. Multifactor authentication becomes useful in information sharing scenarios. After performing a risk assessment based on the type of information held in the SLTT Community's systems, it was determined that multifactor authentication would reduce the risk of a system intrusion. As a result, this document, as well as the Executive Primer and Acquisition Guidance, are all tailored towards ICAM systems with multifactor authentication.

1.5 HOW TO USE

The content within this document provides a deeper view for solutions architects into an ICAM-enabled system configured with multifactor authentication through implementation guides. This document is part of a progressive series that begins with the ICAM Executive Primer, a high-level, educational dive into ICAM for executive leadership and others through ICAM concepts and real-world scenarios. The ICAM

[2] Most public safety communications systems, radio sites, public safety facilities, data centers and radios are physically secured against internal and external threats.

[3] Verizon Enterprise Solutions, 2017 Data Breach Investigations Report, which can be found at http://www.verizonenterprise.com/verizon-insights-lab/dbir/2017/.

Acquisition Guidance is the second in the series. It gives an overview to program managers of what to look for while acquiring ICAM products and advises solutions architects about lessons learned from sandbox builds of ICAM-enabled systems with multifactor authentication. The series ends with an ICAM Federation System Checklist, which includes five topics that should be addressed by a system owner before federating a system. This series also contains several helpful sections and appendices, including reference architecture, implementation guidance and procurement language.

This document was developed by the DHS-established PSC ICAM WG; questions and comments can be sent to DHS S&T at: SandTFRG@hq.dhs.gov.

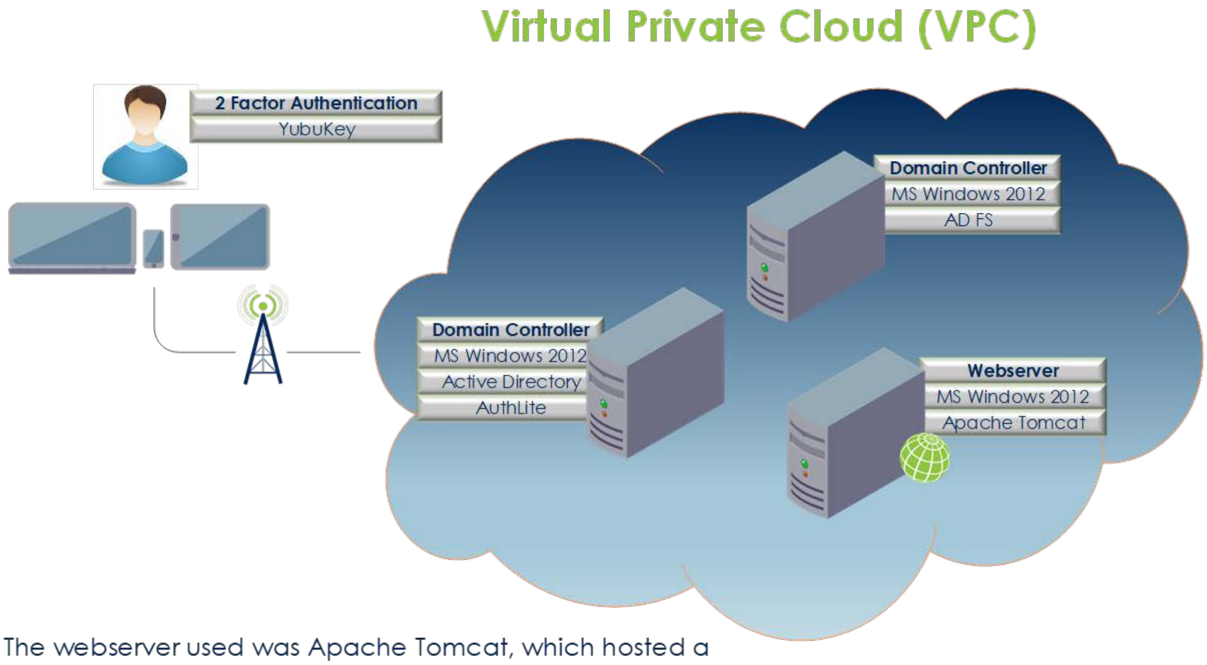
1.6 WHAT IS IN THIS DOCUMENT

To collect data for these Implementation Guides, DHS S&T created ICAM-enabled systems configured with multifactor authentication in a sandbox, producing ample data to create them. Each build used different software and credentials Community members could use in a build. The

Active Directory Federated Services (ADFS) Implementation Guide used low-cost products. The Shibboleth IdP 3 & SP 2 used a free, open source product that was mentioned on the Global Federation Identity and Privilege Management (GFIPM) website. The Duo Access Gateway (DAG) and Okta IdP Implementation Guides are both cloud-based commercial products.

Based on the installation procedures and associated outcomes of these builds, PSC ICAM WG created the below reference architecture and Implementation Guides. In no way does the PSC ICAM WG endorse or favor the vendors and software implemented below. These guides are for educational purposes only. And as such, each organization is responsible for its own cybersecurity measures, and should make decisions based on the requirements, laws and expertise of the organization.

2 Active Directory Federated Services (ADFS) Implementation Guide



- The webserver used was Apache Tomcat, which hosted a resource (webpage) only accessible when on the VPC.
- Microsoft Active Directory was the user directory.
- AuthLite was installed to add a second authentication factor to MS Active Directory

Figure 2 - System Configured for Multifactor Authentication Using Yubikey HMAC-based one-time password, AuthLite & MS Windows Active Directory.

AuthLite is a two-factor authentication tool that can operate on Windows and Active Directory (AD). This build was conducted in an Amazon Web Services (AWS) environment and, as such, its guide contains implementation instructions specific to AWS. This build was configured in conjunction with Active Directory Federated Services (AD FS) to extend AuthLite’s federation capabilities. AD FS is a Microsoft component that can be installed to provide Simplified Sign-on (SSO) to applications across organizational boundaries. AD does not natively support multifactor authentication, so AuthLite was used as a middleware to provide multifactor authentication.

For the purposes of this guide, we used Google Authenticator as a one-time password (OTP) token for the second factor. AuthLite can also be configured to use a Yubikey as a credential. The Yubikey, a FIDO Alliance-approved credential, is a hard token that can be plugged into a universal serial bus (USB) port or used with another near field communication (NFC) device and touched to release a certificate. Its cost varies on the type of Yubikey chosen and the device can be configured several different ways – it could be used as a single-factor authenticator, as an OTP device, or even as a public key infrastructure token. In this build, the Yubikey is configured for two-factor authentication as a keyed-hash message authentication code (HMAC)-based one-time password.

To use the Yubikey in an AWS environment, USB hardware needs to be accessible by the Amazon Elastic Compute Cloud (EC2) instances. One way to accomplish this is by setting up a publicly accessible virtual private network (VPN) tunnel into AWS. Because USB credentials cannot be passed through directly to EC2 instances, this allows the user to provide two authentication factors to gain access to the VPN (or virtual private cloud (VPC) in this case). In this build, OpenVPN was chosen for this purpose and setup to use a Microsoft domain controller as its user directory. AuthLite is relatively inexpensive and interacts seamlessly with the Yubikey and AD. In this build, Yubikey and AD username/password provide the two factors needed to access the VPC via the OpenVPN (and AuthLite) software.

2.1 PREREQUISITES

- Register your Windows Server 2012 server as a member server on a preexisting domain.
- The AD deployment engineer should have domain administrator privileges.
- The ADFS configuration requires a publicly trusted certificate for secure sockets layer (SSL) server authentication according to the following guidelines located here on the Microsoft website.
- Follow Microsoft's installation guide to deploy ADFS and configure it for your domain.

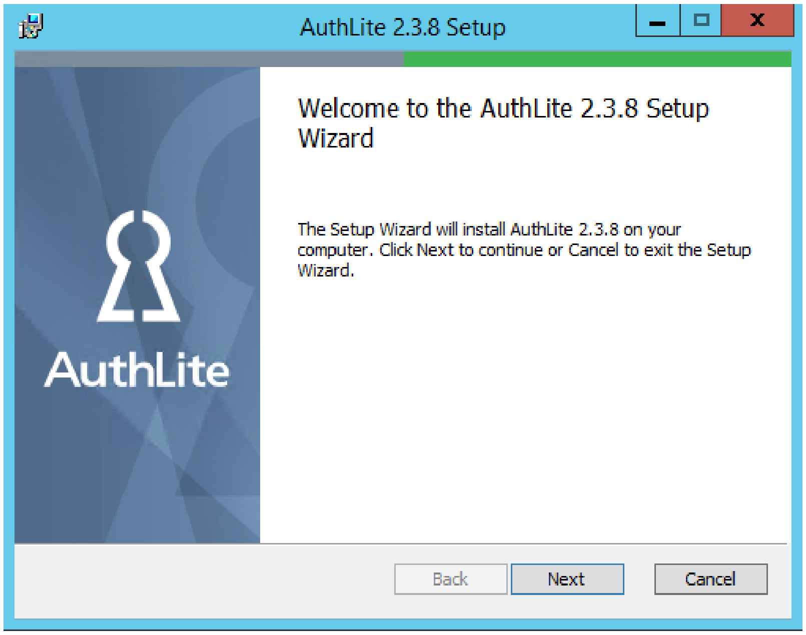
2.2 INSTALLING AUTHLITE ON THE AD FS SYSTEM

Note: The deployment engineer must have the **Schema Admin** role in AD. AuthLite must be installed on a Domain Controller.

1. Download and/or copy the most recent version of AuthLite to a folder located on the local server.

2. Open **Command Prompt** as an administrator and run the installer through the command line.

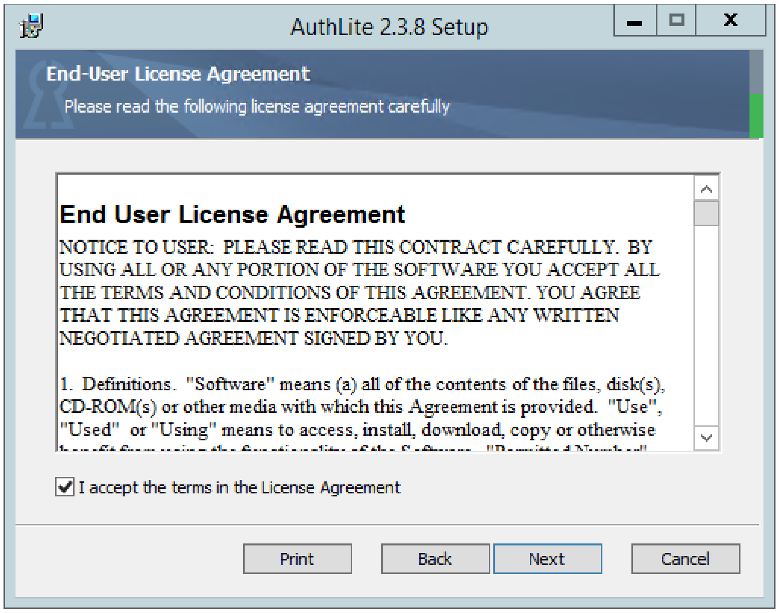
3. Click **Next** on the AuthLite **Welcome** window.



Welcome to the AuthLite 2.3.8 Setup Wizard

The Setup Wizard will install AuthLite 2.3.8 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

4. Accept the license agreement then click **Next**.



End-User License Agreement

Please read the following license agreement carefully

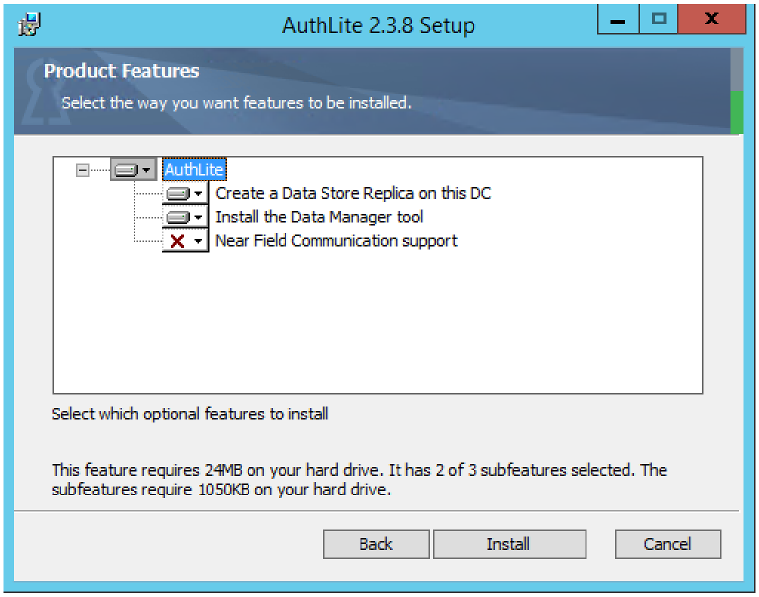
End User License Agreement

NOTICE TO USER: PLEASE READ THIS CONTRACT CAREFULLY. BY USING ALL OR ANY PORTION OF THE SOFTWARE YOU ACCEPT ALL THE TERMS AND CONDITIONS OF THIS AGREEMENT. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

1. Definitions. "Software" means (a) all of the contents of the files, disk(s), CD-ROM(s) or other media with which this Agreement is provided. "Use", "Used" or "Using" means to access, install, download, copy or otherwise benefit from using the functionality of the Software. "Permitted Number"

☑ I accept the terms in the License Agreement

5. Click **Install** on the **Product features** page.



AuthLite 2.3.8 Setup

Product Features

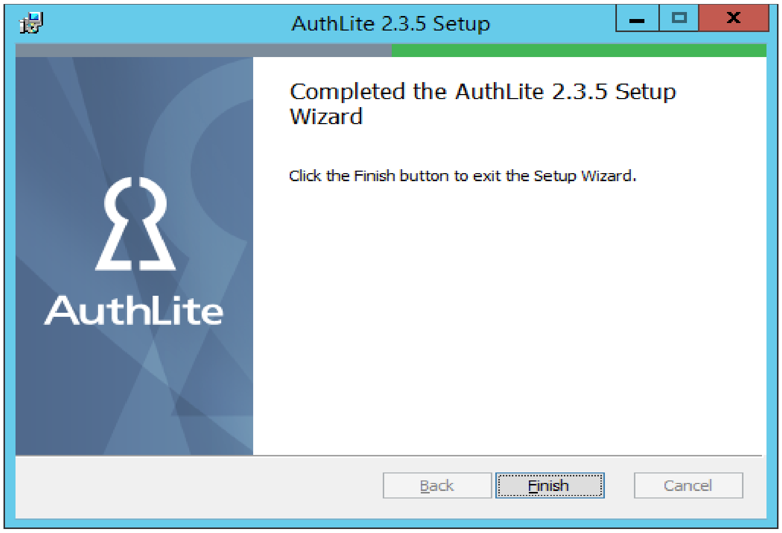
Select the way you want features to be installed.

- AuthLite
  - Create a Data Store Replica on this DC
  - Install the Data Manager tool
  - ✗ Near Field Communication support

Select which optional features to install

This feature requires 24MB on your hard drive. It has 2 of 3 subfeatures selected. The subfeatures require 1050KB on your hard drive.

6. Click **Finish** once the install has completed.

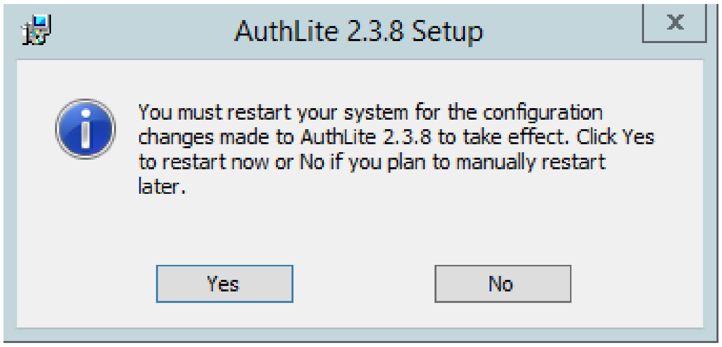


AuthLite 2.3.5 Setup

Completed the AuthLite 2.3.5 Setup Wizard

Click the Finish button to exit the Setup Wizard.

7. Click **Yes** in the AuthLite modal window to restart the server.

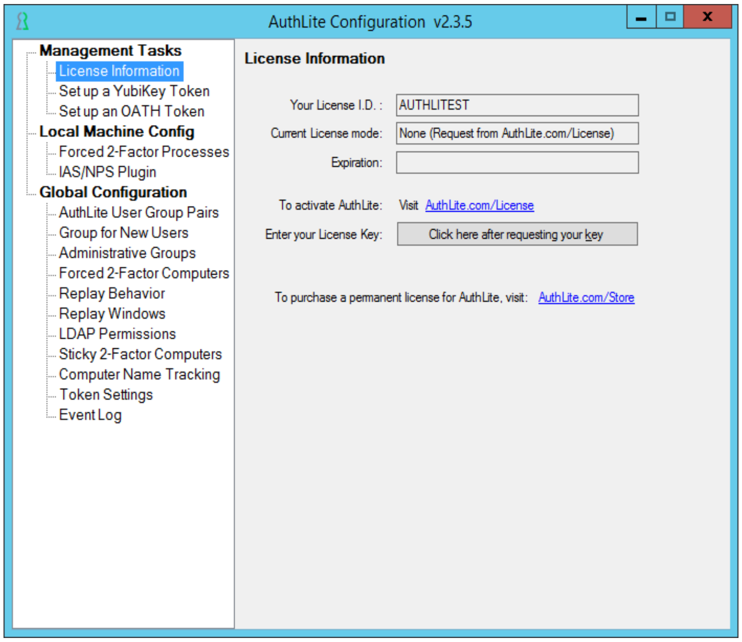


AuthLite 2.3.8 Setup

You must restart your system for the configuration changes made to AuthLite 2.3.8 to take effect. Click Yes to restart now or No if you plan to manually restart later.

2.3 ENABLE AUTHLITE ON AD FS

1. Login to the server once the restart is completed. The AuthLite configuration tool should open



License Information

Your License I.D. : AUTHLITEST

Current License mode: None (Request from AuthLite.com/License)

Expiration:

To activate AuthLite: Visit AuthLite.com/License

Enter your License Key:

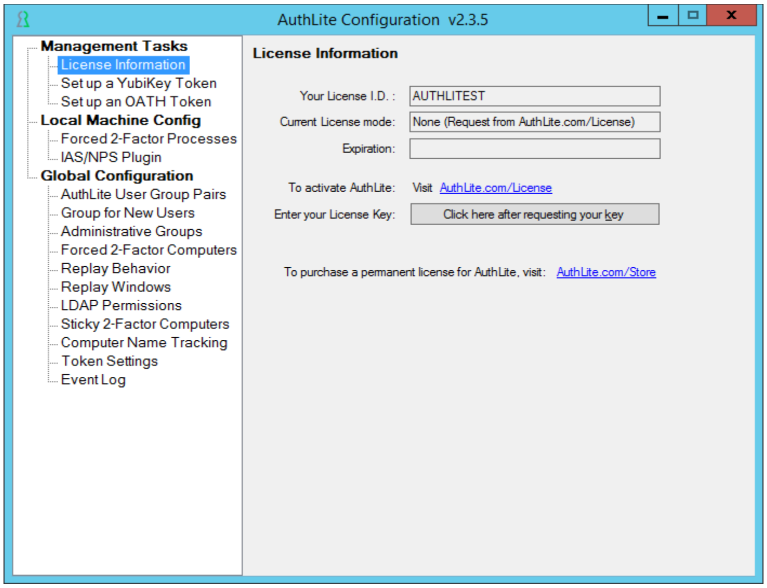
To purchase a permanent license for AuthLite, visit: AuthLite.com/Store

2. Click on **License Information** from the Left Navigation. Request an **AuthLite License Key**.

   a. Go to authlite.com/license to request a temporary evaluation license key

   OR

   b. Go to authlite.com/store to purchase a permanent license for your deployment.



License Information

Your License I.D. : AUTHLITEST

Current License mode: None (Request from AuthLite.com/License)

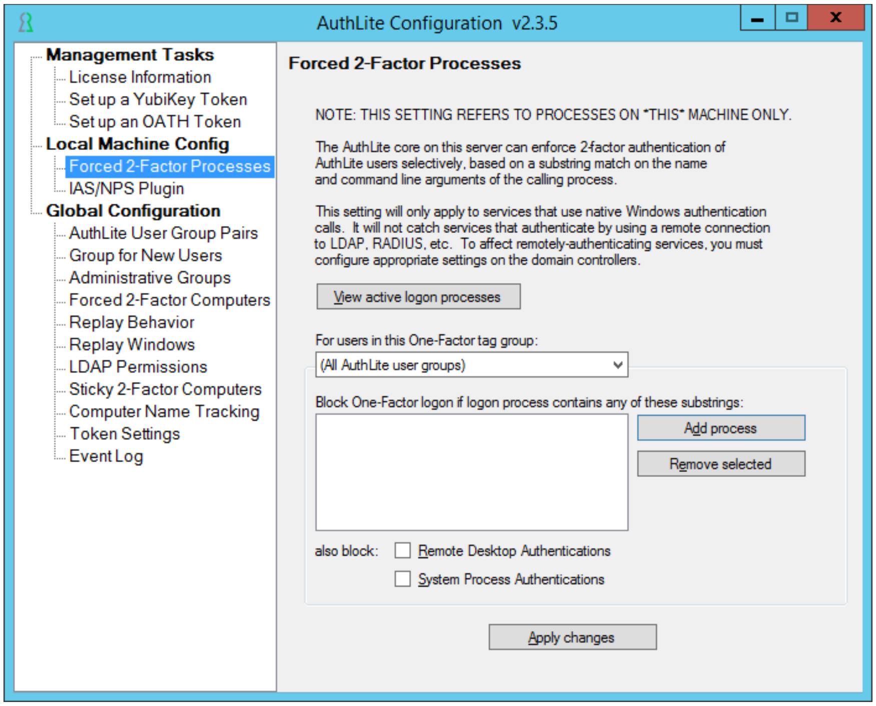
Expiration:

To activate AuthLite: Visit AuthLite.com/License

Enter your License Key:

To purchase a permanent license for AuthLite, visit: AuthLite.com/Store

3. Click on **Forced 2-Factor Processes** under **Local Machine Config**.



Forced 2-Factor Processes

NOTE: THIS SETTING REFERS TO PROCESSES ON *THIS* MACHINE ONLY.

The AuthLite core on this server can enforce 2-factor authentication of AuthLite users selectively, based on a substring match on the name and command line arguments of the calling process.

This setting will only apply to services that use native Windows authentication calls. It will not catch services that authenticate by using a remote connection to LDAP, RADIUS, etc. To affect remotely-authenticating services, you must configure appropriate settings on the domain controllers.

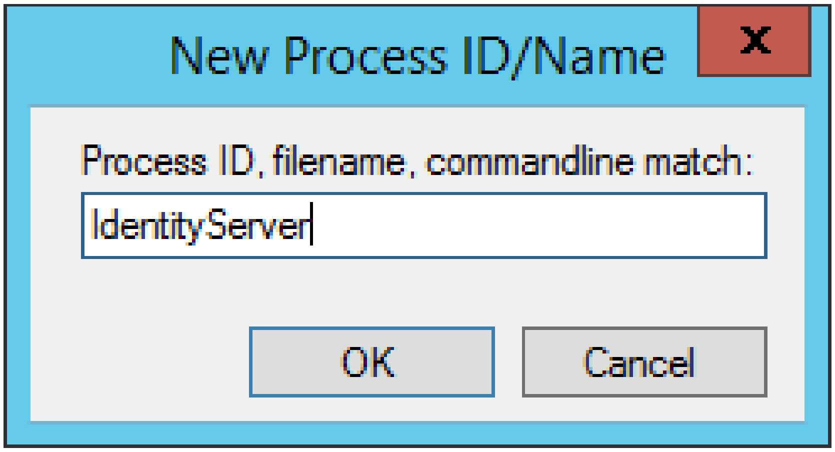
For users in this One-Factor tag group: (All AuthLite user groups)

Block One-Factor logon if logon process contains any of these substrings:

also block: □ Remote Desktop Authentications

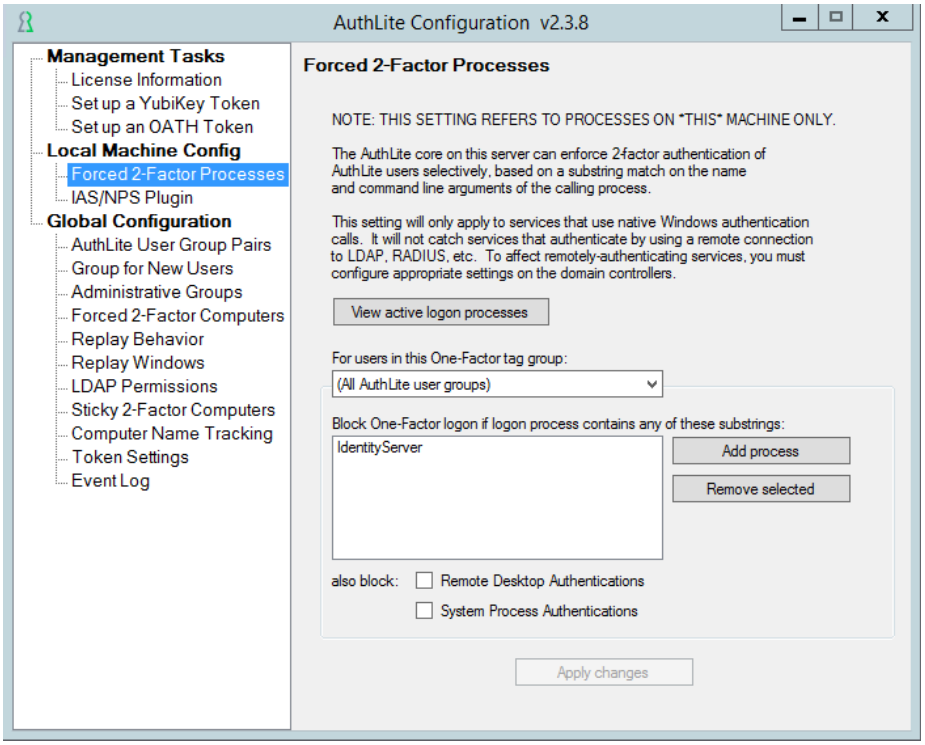
□ System Process Authentications

4. Click the **Add process** button. Enter **IdentityServer** in the text box. Click **OK**.



Process ID, filename, commandline match: IdentityServer

5. Click **Apply Changes**.



Forced 2-Factor Processes

NOTE: THIS SETTING REFERS TO PROCESSES ON *THIS* MACHINE ONLY.

The AuthLite core on this server can enforce 2-factor authentication of AuthLite users selectively, based on a substring match on the name and command line arguments of the calling process.

This setting will only apply to services that use native Windows authentication calls. It will not catch services that authenticate by using a remote connection to LDAP, RADIUS, etc. To affect remotely-authenticating services, you must configure appropriate settings on the domain controllers.

For users in this One-Factor tag group: (All AuthLite user groups)

Block One-Factor logon if logon process contains any of these substrings: IdentityServer

also block: □ Remote Desktop Authentications

□ System Process Authentications

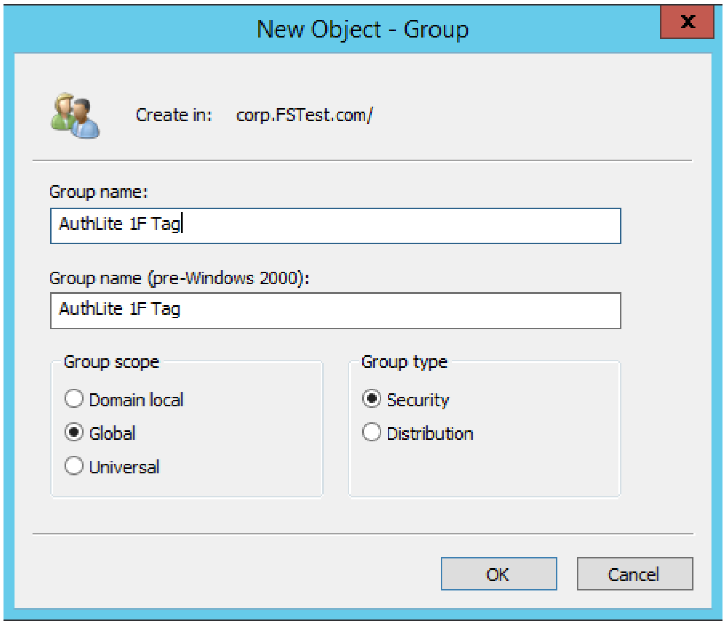
2.4 CREATE AUTHLITE GROUPS IN AD FS

The creation of AuthLite groups in AD is required for AuthLite to enforce multifactor authentication in

1. Open **Active Directory Users and Computers**. This can be opened from the **Tools** menu in the **Server Manager**. Select **Active Directory Users and Computers** from the dropdown.

2. Create the following three Global Security Groups in your Domain. The Security Group names are specific and case sensitive:

   a. AuthLite Users
   b. AuthLite 1F Tag
   c. AuthLite 2F Tag



New Object - Group

Create in: corp.FSTest.com/

Group name: AuthLite 1F Tag

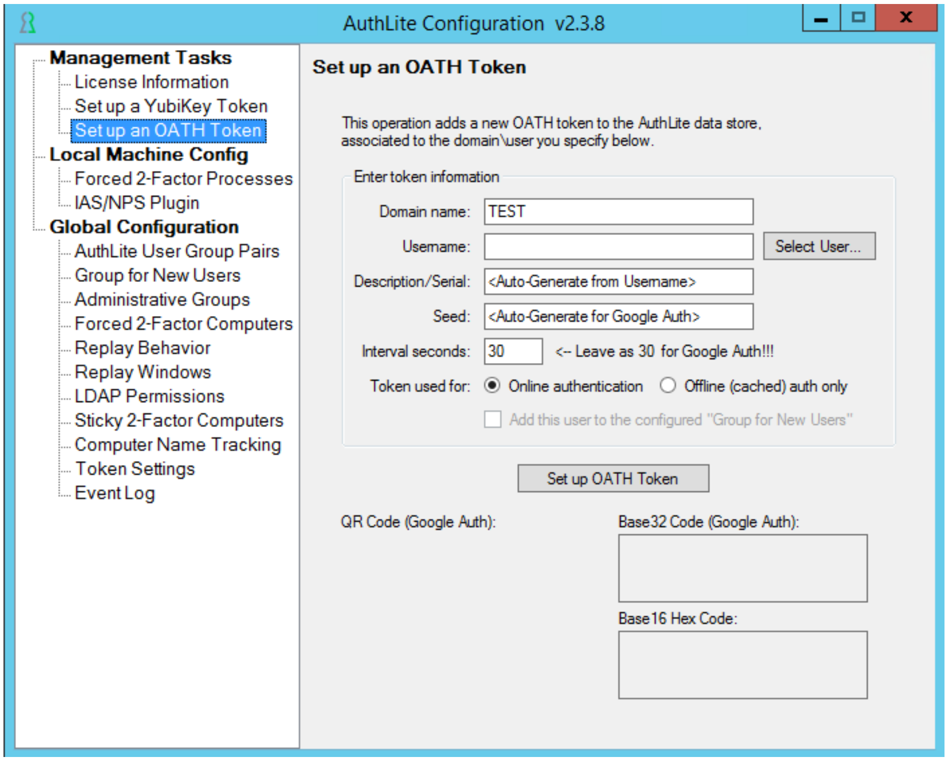
Group name (pre-Windows 2000): AuthLite 1F Tag

Group scope
○ Domain local
● Global
○ Universal

Group type
● Security
○ Distribution

3. Add any Users you wish to be managed by AuthLite to the AuthLite Users group.

4. Users in the AuthLite Users group must be provided with a multifactor token. Multifactor tokens can be created from **Setup an OATH Token** under **Management Tasks** in the left navigation pane. Note: Users can be managed within the AuthLite Token Manager tool.



AuthLite Configuration v2.3.8

- Management Tasks
  - License Information
  - Set up a YubiKey Token
  - Set up an OATH Token
- Local Machine Config
  - Forced 2-Factor Processes
  - IAS/NPS Plugin
- Global Configuration
  - AuthLite User Group Pairs
  - Group for New Users
  - Administrative Groups
  - Forced 2-Factor Computers
  - Replay Behavior
  - Replay Windows
  - LDAP Permissions
  - Sticky 2-Factor Computers
  - Computer Name Tracking
  - Token Settings
  - Event Log

Set up an OATH Token

This operation adds a new OATH token to the AuthLite data store, associated to the domain\user you specify below.

Enter token information

Domain name: TEST

Username:

Select User...

Description/Serial: <Auto-Generate from Username>

Seed: <Auto-Generate for Google Auth>

Interval seconds: 30 <-- Leave as 30 for Google Auth!!!

Token used for: ● Online authentication ○ Offline (cached) auth only

□ Add this user to the configured "Group for New Users"

Set up OATH Token

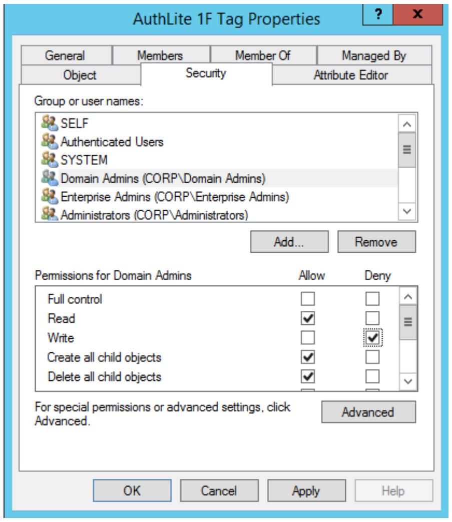
QR Code (Google Auth):

Base32 Code (Google Auth):

Base16 Hex Code:

5. Add the **AuthLite Users** group as a **Member** of the AuthLite 1F Tag group. This ensures that all members of the AuthLite users group are also members of the AuthLite 1F Tag group.

6. Set the domain administrators group to **Deny on Write** for both the AuthLite 1F Tag and AuthLite 2F Tag groups. This is preventative but not required. AuthLite should be allowed to assign these permissions dynamically.



AuthLite 1F Tag Properties

General | Members | Member Of | Managed By | Object | Security | Attribute Editor

Group or user names:

- SELF
- Authenticated Users
- SYSTEM
- Domain Admins (CORP\Domain Admins)
- Enterprise Admins (CORP\Enterprise Admins)
- Administrators (CORP\Administrators)

Add... Remove

Permissions for Domain Admins	Allow	Deny
Full control	□	□
Read	☑	□
Write	□	☑
Create all child objects	☑	□
Delete all child objects	☑	□

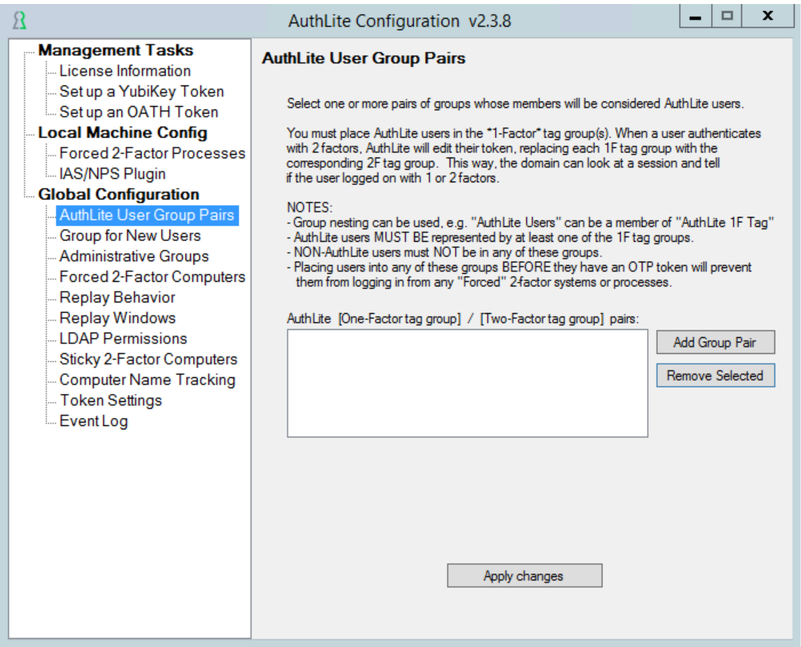
For special permissions or advanced settings, click Advanced.

Advanced

OK Cancel Apply Help

7. Configure the 1F and 2F Group Pair in the AuthLite Configuration Tool.

   a. Click the **AuthLite User Group Pairs** item on the left navigation panel



AuthLite User Group Pairs

Select one or more pairs of groups whose members will be considered AuthLite users.

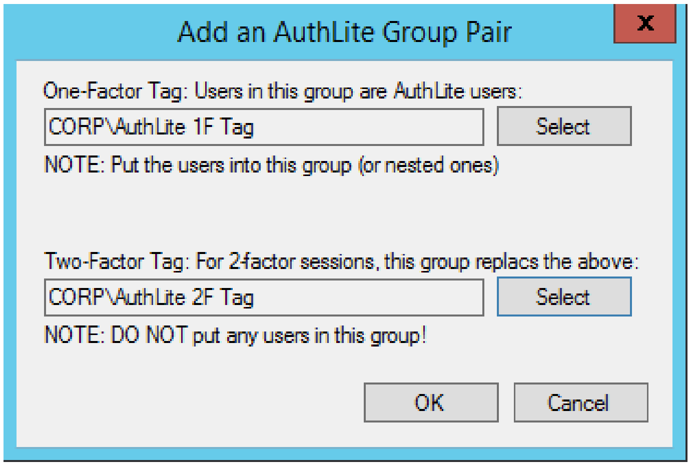
You must place AuthLite users in the *1-Factor* tag group(s). When a user authenticates with 2 factors, AuthLite will edit their token, replacing each 1F tag group with the corresponding 2F tag group. This way, the domain can look at a session and tell if the user logged on with 1 or 2 factors.

NOTES:
- Group nesting can be used, e.g. "AuthLite Users" can be a member of "AuthLite 1F Tag"
- AuthLite users MUST BE represented by at least one of the 1F tag groups.
- NON-AuthLite users must NOT be in any of these groups.
- Placing users into any of these groups BEFORE they have an OTP token will prevent them from logging in from any "Forced" 2-factor systems or processes.

AuthLite [One-Factor tag group] / [Two-Factor tag group] pairs:

b. Click **Add Group Pair**

c. Add the previously created AuthLite 1F and 2F Tag Security Groups to their relevant fields. Click **OK**



One-Factor Tag: Users in this group are AuthLite users:

CORP\AuthLite 1F Tag

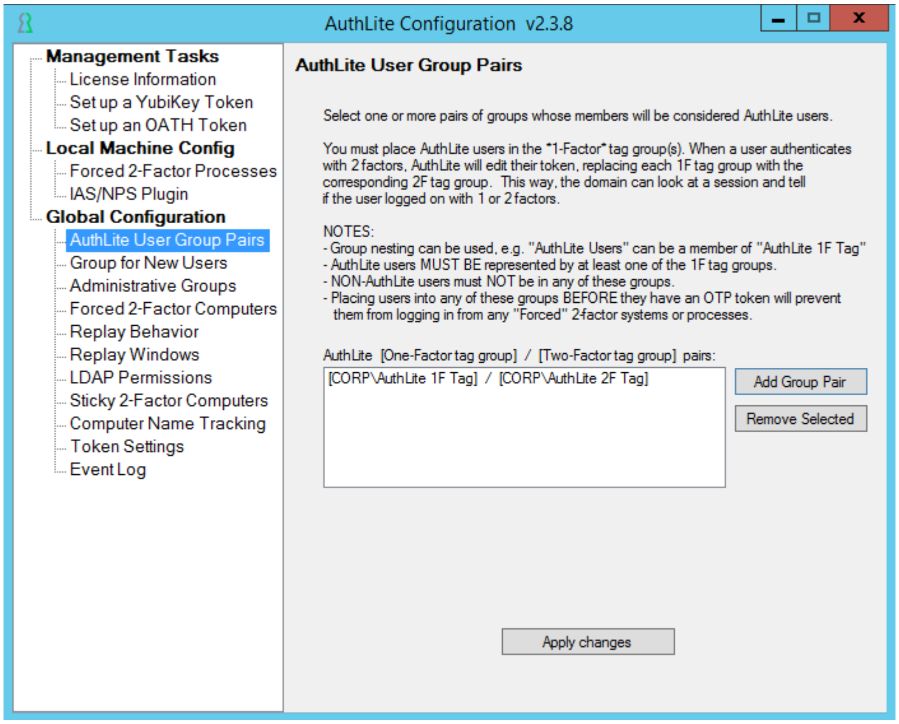
NOTE: Put the users into this group (or nested ones)

Two-Factor Tag: For 2-factor sessions, this group replacs the above:

CORP\AuthLite 2F Tag

NOTE: DO NOT put any users in this group!

d. Click **Apply changes**



AuthLite User Group Pairs

Select one or more pairs of groups whose members will be considered AuthLite users.

You must place AuthLite users in the "1-Factor" tag group(s). When a user authenticates with 2 factors, AuthLite will edit their token, replacing each 1F tag group with the corresponding 2F tag group. This way, the domain can look at a session and tell if the user logged on with 1 or 2 factors.

NOTES:
- Group nesting can be used, e.g. "AuthLite Users" can be a member of "AuthLite 1F Tag"
- AuthLite users MUST BE represented by at least one of the 1F tag groups.
- NON-AuthLite users must NOT be in any of these groups.
- Placing users into any of these groups BEFORE they have an OTP token will prevent them from logging in from any "Forced" 2-factor systems or processes.

AuthLite [One-Factor tag group] / [Two-Factor tag group] pairs:

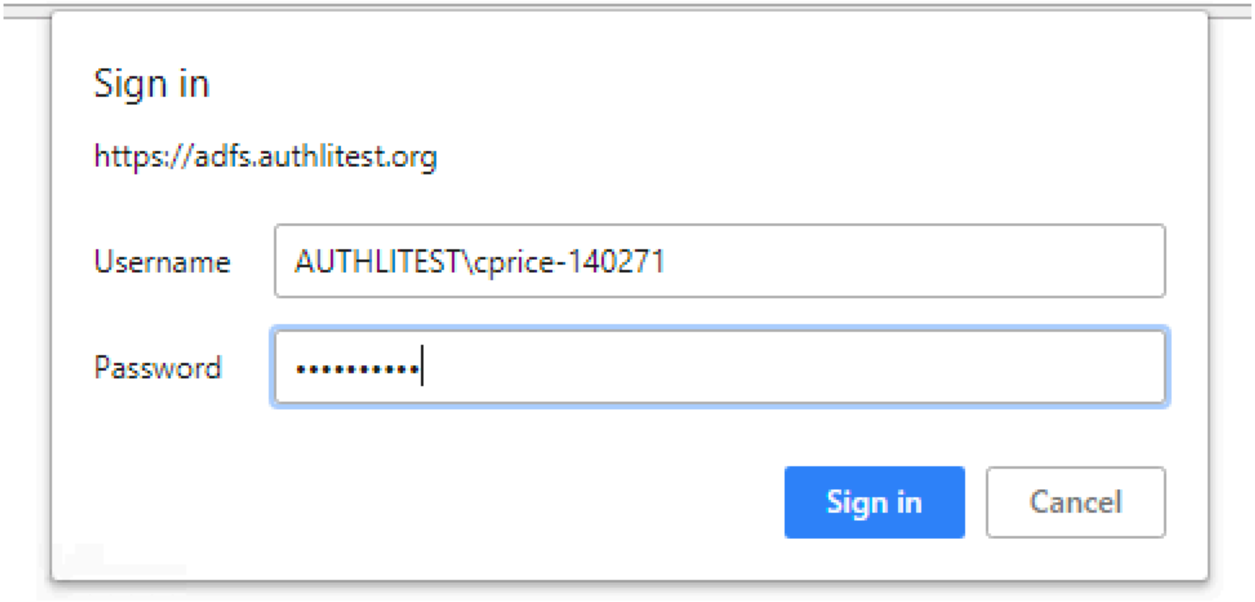
[CORP\AuthLite 1F Tag] / [CORP\AuthLite 2F Tag]

2.5 Logging in with AuthLite Multifactor Authentication

Note: Now any AuthLite users who are authorized into the AD FS environment will be required to use two-factor authentication for specified activities. To verify, attempt to use one-factor authentication and ensure that those requests are blocked.

To log in with AuthLite multifactor authentication, do the following:

1. Get an OTP token code.
2. Append the code to our name as seen in the figure below.
3. Enter your password.



Sign in

https://adfs.authlitest.org

Username AUTHLITEST\cprice-140271

Password ••••••••••

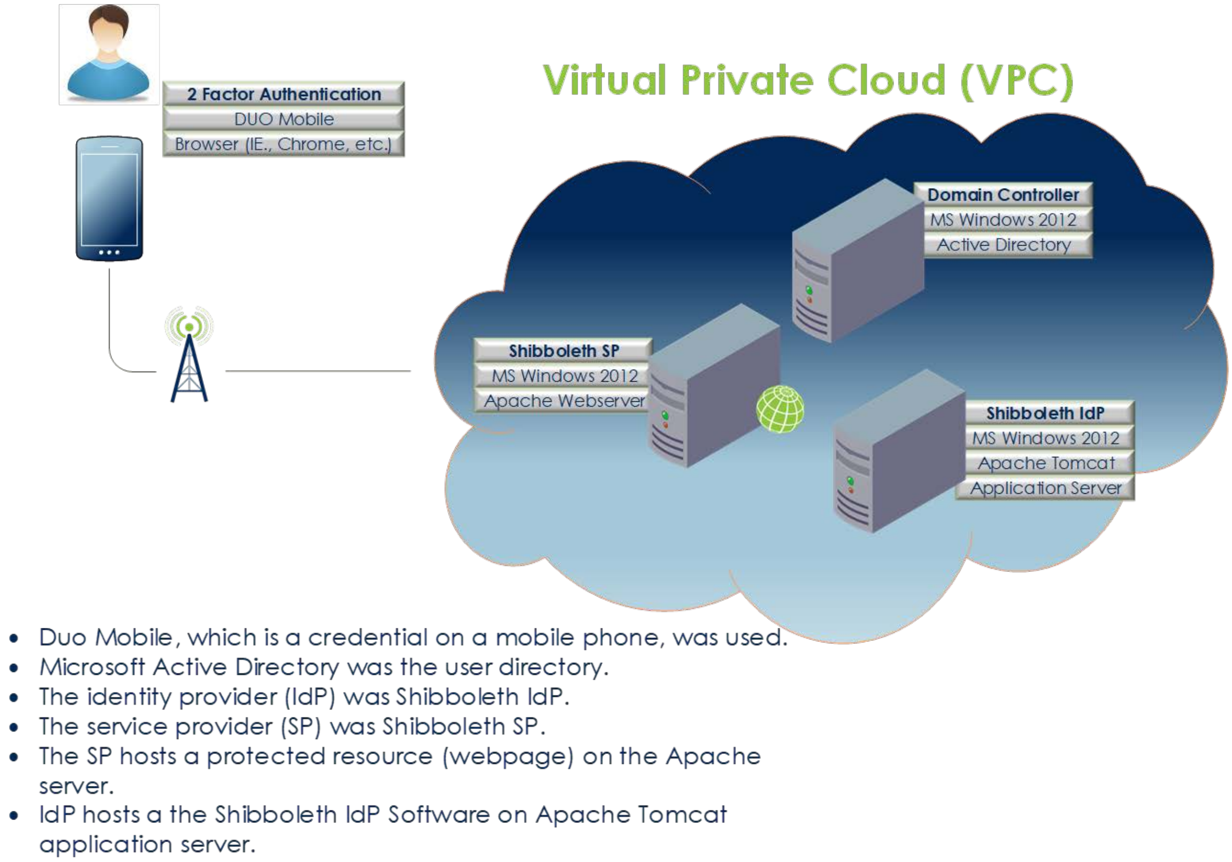
References

AD FS Requirements. (2018, March 05). Retrieved from Microsoft.com: https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/overview/ad-fs-requirements#BKMK_1

Certificate Requirements for Federation Servers. (2017, May 30). Retrieved from Microsoft.com: https://docs.microsoft.com/en-us/windows-server/identity/ad-fs/design/certificate-requirements-for-federation-servers

Use Group Policy to enforce 2-factor on Windows servers/workstations. (2018). Retrieved from AuthLite.com: https://www.authlite.com/docs/2_3/id_1587484890/

3 Shibboleth IdP 3 & SP 2 Implementation Guide



- Duo Mobile, which is a credential on a mobile phone, was used.
- Microsoft Active Directory was the user directory.
- The identity provider (IdP) was Shibboleth IdP.
- The service provider (SP) was Shibboleth SP.
- The SP hosts a protected resource (webpage) on the Apache server.
- IdP hosts a the Shibboleth IdP Software on Apache Tomcat application server.

Figure 3 - System Configured for Multifactor Authentication Using Duo TOTP (Time-based One-Time Password), Shibboleth IdP, Shibboleth SP & MS Windows Active Directory.

Shibboleth is a federated identity solution that provides simplified sign-on (SSO) capabilities, along with multifactor authentication. Because the GFIPM website provides a link to implementation documentation for Shibboleth (a free, open source product), in this build Shibboleth was used as an identity provider and simplified sign-on solution, as well as a service provider.

This build includes a time-based one-time password (TOTP) credential on your phone. Duo was chosen as the credential provider because it interoperates with Shibboleth and is free for up to 10 users. Duo's mobile phone TOTP application allows for a six-digit number found on the phone, a phone call, or a button that pops up on your phone to prove you with the credential.

About

This implementation guide has been created to assist organizations with installing and configuring Shibboleth IdP 3 and SP 2 to authenticate users over Lightweight Directory Access Protocol (LDAP) against Microsoft AD. Additionally, it outlines how to add a second factor to authentication using Duo TOTP. While researching Shibboleth, it was found that much of the publicly available documentation for installing and configuring Shibboleth was not straightforward, especially for those who may be unfamiliar with the software. This guide is for implementation of Shibboleth software in an AWS cloud environment, and should be used as a reference in conjunction with the current publicly available Shibboleth documentation. Throughout this guide, links have been provided to offer additional assistance and information.

What is the Shibboleth Software?

This step-by-step guide provides detailed instructions on how to successfully implement the Shibboleth software components in an AWS cloud environment. The Shibboleth software is a web-based SSO system made up of three components:

- The IdP is responsible for user authentication and providing user information to the SP. It is located at the home organization, which is the organization that maintains the user's account.
- The SP is responsible for protecting an online resource and consuming information from the IdP. It is located at the resource organization, which is the organization that keeps records that will be shared.
- The Discovery Service (DS) helps the SP discover the user's IdP. It may be located anywhere on the web and is not required in all cases.[4]

Shibboleth has two major halves: an IdP and a SP. The IdP supplies information about users to applications, and the SP gathers information about users to protect resources. In the typical use case, a web browser accesses a protected resource, authenticates at the identity provider and once authenticated, is logged into the resource.[5]

- **Web Browser** – represents the user within the SSO process.
- **Resource** – contains restricted access content that the user wants to access.
- **IdP** – authenticates the user.
- **SP** – performs the SSO process for the resource.[6]

[4] Shibboleth Concepts, available at https://wiki.shibboleth.net/confluence/display/CONCEPT.

[5] Shibboleth Wiki: Flows and Configs, https://wiki.shibboleth.net/confluence/display/CONCEPT/FlowsAndConfig.

[6] How Shibboleth Works: Basic Concepts, https://www.shibboleth.net/index/basic.

BEFORE YOU BEGIN – A LIST OF HELPFUL TIPS:

- Although it is encouraged to do so, always be careful when/if copy and pasting code from this guide, and double-check that you have substituted the correct values.
- Always double-check that you are downloading the correct version of any software.
- Sometimes simple missteps can cause big problems; remember to check the logs included with all software utilized during this process to determine the cause of any issues you encounter.
- There are multiple ways to configure Shibboleth depending on what you are doing – one configuration/integration will not necessarily work for everyone.
- There is very limited documentation available for troubleshooting.
- Document the steps you have taken as you go, just in case a component breaks – this will allow you to compare the documentation of your build with the documentation of this guide.
- Setting up a system that utilizes the Shibboleth IdP and/or SP is a long process. It will take a lot of time and resources and will require the assistance of senior-level engineers.
- Remember that https://www.testshib.org/ is a very useful troubleshooting tool during the latter steps of the Shibboleth software configuration.

3.1 PREREQUISITES

You will need to create at least a three server networks for this configuration – an IdP, an SP, and an AD server. The servers should have a base hardware spec of 2GB+ of RAM and 160GB of Storage. All guides are written using Windows Server 2012 as the Operating System.

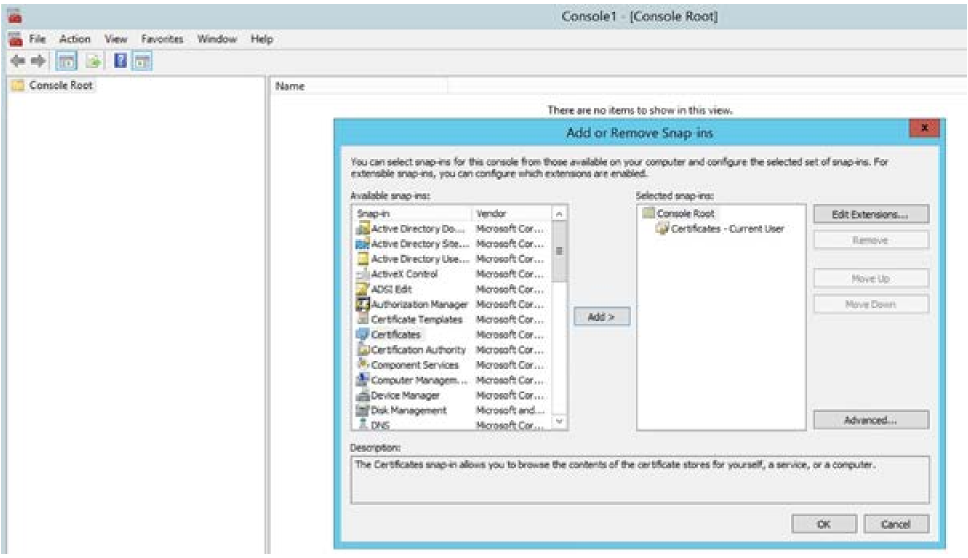
- Disable both Internet Explorer Enhanced Security and Windows Firewall.
- Download and Install Chrome and Notepad++ on both the IdP and SP servers.

ADD ACTIVE DIRECTORY CERTIFICATE SERVICES

Follow Microsoft’s installation guide to add the Certificate Services role to your AD server.

EXPORT CERTIFICATE TO IDP

1. Open **Run** and use the **mmc.exe** command to open the **Microsoft management console (MMC)**.
2. Open **File**, select **Add or Remove Snap In** and select **Certificate Snap In.** Click **OK** .



There are no items to show in this view.

You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
Active Directory Do...	Microsoft Cor...
Active Directory Site...	Microsoft Cor...
Active Directory Use...	Microsoft Cor...
ActiveX Control	Microsoft Cor...
ADSI Edit	Microsoft Cor...
Authorization Manager	Microsoft Cor...
Certificate Templates	Microsoft Cor...
Certificates	Microsoft Cor...
Certification Authority	Microsoft Cor...
Component Services	Microsoft Cor...
Computer Managem...	Microsoft Cor...
Device Manager	Microsoft Cor...
Disk Management	Microsoft and...
DNS	Microsoft Cor...

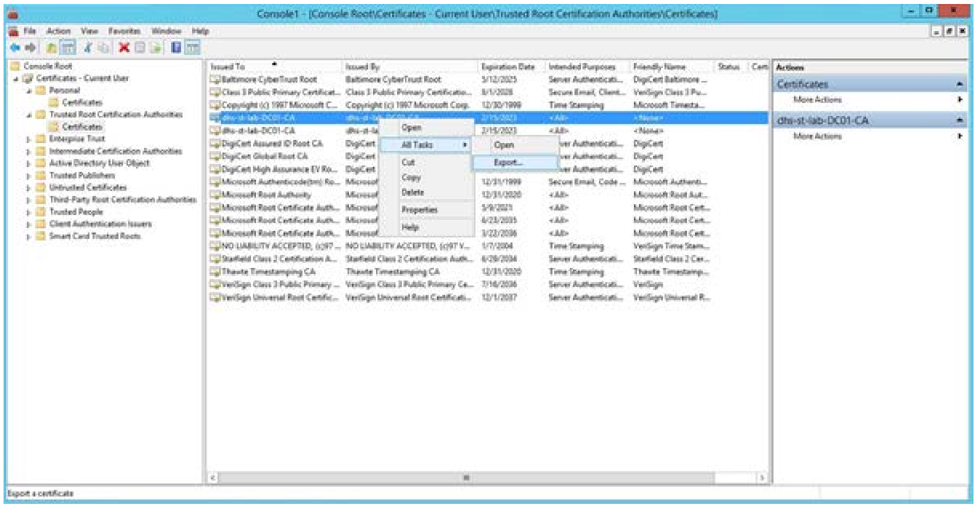
Selected snap-ins:

Console Root
Certificates - Current User

Description:

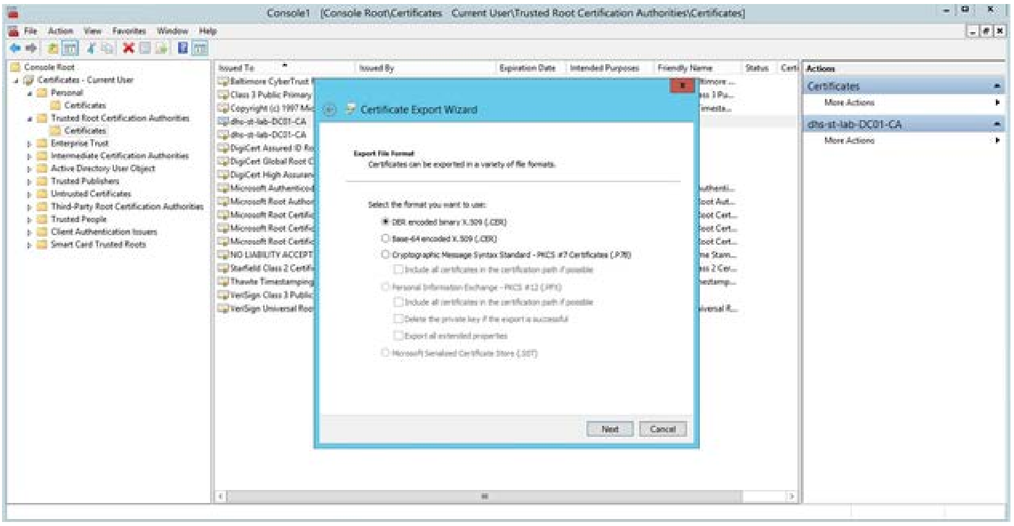
The Certificates snap-in allows you to browse the contents of the certificate stores for yourself, a service, or a computer.

3. Select the **Certificate** that has the name of your **AD Domain**. Right click and select **Export** from **All Tasks**.


Console1 - [Console Root\Certificates - Current User\Trusted Root Certification Authorities\Certificates]
File
Action
View
Favorites
Window
Help
Console Root
Certificates - Current User
Personal
Certificates
Trusted Root Certification Authorities
Certificates
Enterprise Trust
Intermediate Certification Authorities
Active Directory User Object
Trusted Publishers
Untrusted Certificates
Third-Party Root Certification Authorities
Trusted People
Client Authentication Issuers
Smart Card Trusted Roots
Open
All Tasks
Cut
Copy
Delete
Properties
Help
Open
Export...
Actions
Certificates
More Actions
dhs-st-lab-DC01-CA
More Actions
Export a certificate

Issued To	Issued By	Expiration Date	Intended Purposes	Friendly Name	Status	Cert
Baltimore CyberTrust Root	Baltimore CyberTrust Root	5/12/2025	Server Authenticati...	DigiCert Baltimore ...		
Class 3 Public Primary Certificat...	Class 3 Public Primary Certificatio...	8/1/2028	Secure Email, Client...	VeriSign Class 3 Pu...		
Copyright (c) 1997 Microsoft C...	Copyright (c) 1997 Microsoft Corp.	12/30/1999	Time Stamping	Microsoft Timesta...		
dhs-st-lab-DC01-CA	dhs-st-lab-DC01-CA	2/15/2023	<All>	<None>		
dhs-st-lab-DC01-CA	dhs-st-la	2/15/2023	<All>	<None>		
DigiCert Assured ID Root CA	DigiCert		ver Authenticati...	DigiCert		
DigiCert Global Root CA	DigiCert		ver Authenticati...	DigiCert		
DigiCert High Assurance EV Ro...	DigiCert		ver Authenticati...	DigiCert		
Microsoft Authenticode(tm) Ro...	Microsof	12/31/1999	Secure Email, Code ...	Microsoft Authenti...		
Microsoft Root Authority	Microsof	12/31/2020	<All>	Microsoft Root Aut...		
Microsoft Root Certificate Auth...	Microsof	5/9/2021	<All>	Microsoft Root Cert...		
Microsoft Root Certificate Auth...	Microsof	6/23/2035	<All>	Microsoft Root Cert...		
Microsoft Root Certificate Auth...	Microsof	3/22/2036	<All>	Microsoft Root Cert...		
NO LIABILITY ACCEPTED, (c)97 ...	NO LIABILITY ACCEPTED, (c)97 V...	1/7/2004	Time Stamping	VeriSign Time Stam...		
Starfield Class 2 Certification A...	Starfield Class 2 Certification Auth...	6/29/2034	Server Authenticati...	Starfield Class 2 Cer...		
Thawte Timestamping CA	Thawte Timestamping CA	12/31/2020	Time Stamping	Thawte Timestamp...		
VeriSign Class 3 Public Primary ...	VeriSign Class 3 Public Primary Ce...	7/16/2036	Server Authenticati...	VeriSign		
VeriSign Universal Root Certific...	VeriSign Universal Root Certificati...	12/1/2037	Server Authenticati...	VeriSign Universal R...		

4. Press **Next** and save the certificate (Choose a name easy to remember and find, i.e., ADCERT).



Console1 - [Console Root\Certificates - Current User\Trusted Root Certification Authorities\Certificates]

Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

Select the format you want to use:

● DER encoded binary X.509 (.CER)

○ Base-64 encoded X.509 (.CER)

○ Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)

□ Include all certificates in the certification path if possible

○ Personal Information Exchange - PKCS #12 (.PFX)

□ Include all certificates in the certification path if possible

□ Delete the private key if the export is successful

□ Export all extended properties

○ Microsoft Serialized Certificate Store (.SST)

5. Copy the newly created file to the IdP desktop through the Remote Desktop clipboard functionality.

3.2 Configuring the IdP Server

Download and install Java 8

1. Follow the hyperlink above to download **Java Development Kit (JDK) for Windows 64 bit** and run the installer.
2. Set the `JAVA_HOME` system variable to JDK install path (i.e., c:\Program Files\Java\jdkxx).
    a. From the Start Menu right click computer.
    b. Click **Properties**.
    c. Click **Advanced System Settings**.
    d. Click **Environment Variables**.
    e. Under System Variables, click **New…**”
    f. Name the variable `JAVA_HOME` and point it to your JDK install directory.

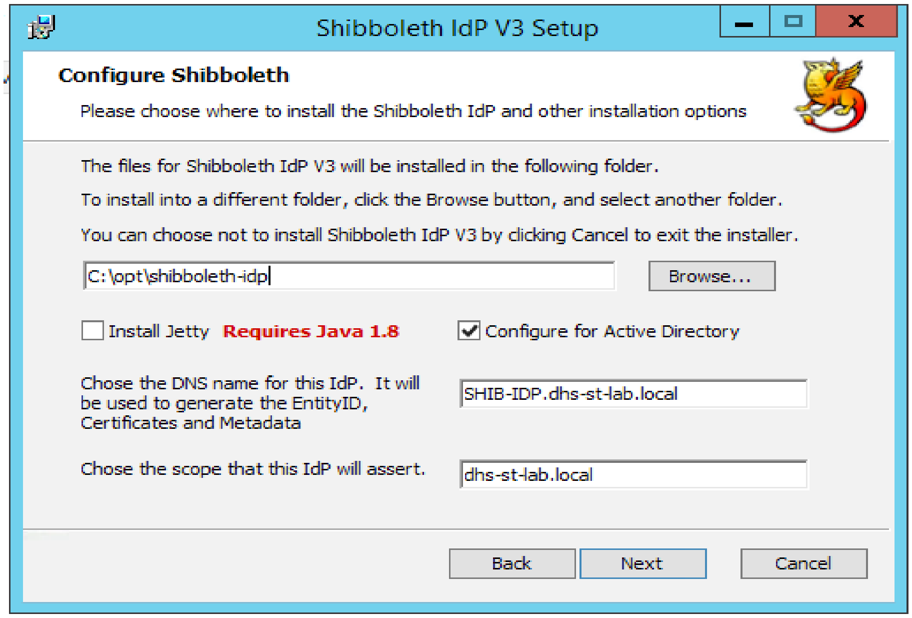
Download and install Tomcat 8.5

The identity provider is a standard Java web application and runs on a compatible servlet container. Tomcat 8.5 is the container documented in this guide to deploy and run the IdP within our environment.

1. Follow the hyperlink above to download **Tomcat 8.5 32-bit/64-bit Windows Service Installer**.
2. Run the installer, set **administrator username/password**, and accept all other defaults.
3. Navigate to your Tomcat 8.5 install directory (default is C:\Apache Software Foundation).
4. Click the **checkbox** that enables Tomcat to run **immediately after install**.
5. Open http://localhost:8080/ with your browser (if you can see the Tomcat homepage, then you have successfully installed Tomcat 8.5).

Installing the Shibboleth IdP v3

1. Download Shibboleth IdP v3 onto your IdP server instance with Tomcat 8.5 installed on it.
2. Run the **IdP Windows Installer**. Set the install path to C:\opt\shibboleth-idp./



Shibboleth IdP V3 Setup

Configure Shibboleth

Please choose where to install the Shibboleth IdP and other installation options

The files for Shibboleth IdP V3 will be installed in the following folder.

To install into a different folder, click the Browse button, and select another folder.

You can choose not to install Shibboleth IdP V3 by clicking Cancel to exit the installer.

C:\opt\shibboleth-idp

□ Install Jetty **Requires Java 1.8**

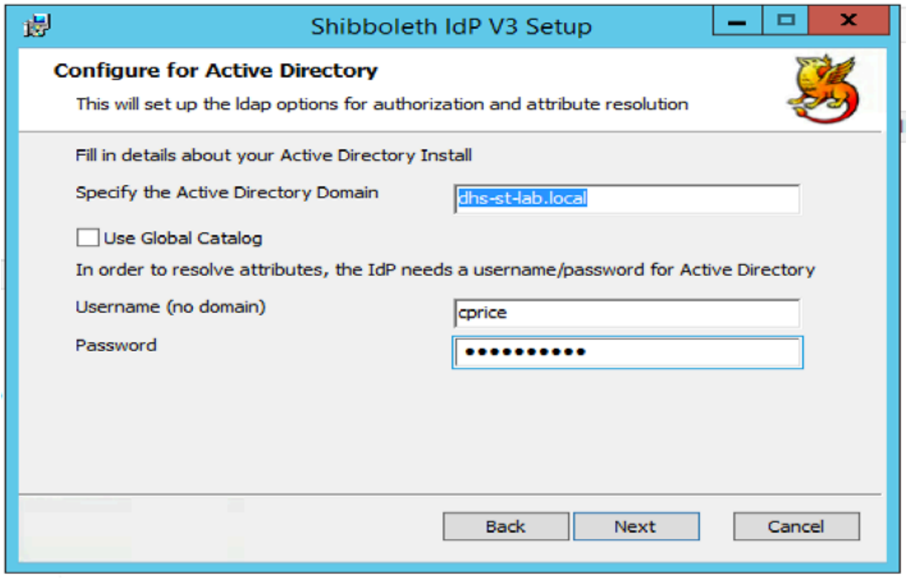
☑ Configure for Active Directory

Chose the DNS name for this IdP. It will be used to generate the EntityID, Certificates and Metadata: SHIB-IDP.dhs-st-lab.local

Chose the scope that this IdP will assert.: dhs-st-lab.local

3. Check the **Configure for Active Directory** checkbox.
4. Specify the **DNS name** for your IdP (i.e., SHIB-IDP.dhs-st-lab.local).
5. Specify the **scope** (i.e. dhs-st-lab.local).

6. Click **Next**.



Shibboleth IdP V3 Setup

Configure for Active Directory

This will set up the ldap options for authorization and attribute resolution

Fill in details about your Active Directory Install

Specify the Active Directory Domain: dhs-st-lab.local

□ Use Global Catalog

In order to resolve attributes, the IdP needs a username/password for Active Directory

Username (no domain): cprice

Password: ●●●●●●●●●●●

7. Specify the **Active Directory Domain** (i.e., dhs-st-lab.local).
8. Leave the **Use Global Catalog** checkbox **unchecked**.
9. Enter the credentials for the Active Directory Admin account that will be used to perform LDAP lookups.
10. Click **Next**.

CONFIGURING SSL ON TOMCAT

1. Create and initialize **Java keystore**.
   a. Open cmd (Command Prompt) as Administrator and run the following keytool command:

```
%JAVA_HOME%\bin\keytool” -genkey -alias tomcat -keyalg RSA
```

   b. Set a password.

2. Provide answers to the questions asked when creating the certificate store.
   a. Enable secure sockets layer (SSL) by adding the following connector to server.xml:

```
<Connector
protocol="org.apache.coyote.http11.Http11NioProtocol"
port="8443" maxThreads="200"
scheme="https" secure="true" SSLEnabled="true"
keystoreFile="<PATH TO KEYSTORE>" keystorePass="<PASSWORD>"
clientAuth="false" sslProtocol="TLS"/>
```

   b. Open **Services** and restart **Tomcat**.
   c. To check if https is enabled, navigate to https://localhost:8443.

CONFIGURING THE SHIBBOLETH IDP V3

1. Create an xml file named **idp.xml** used to dynamically deploy the idp.war file when it is rebuilt.
   a. Copy and paste the following into idp.xml

```
 <Context docBase="C:/opt/shibboleth-idp/war/idp.war"
privileged="true"
antiResourceLocking="false"
```

```
swallowOutput="true">

 <!-- Work around lack of Max-Age support in IE/Edge -->

 <CookieProcessor alwaysAddExpires="true" />
 </Context>
```

   b. Save idp.xml into <tomcat dir>/conf/Catalina/localhost/.

2. Open the **ldap.properties** files and update the **AD server name** and **ldapURL** to reflect your instance. Verify that the other values are correct.
3. Modify C:\opt\shibboleth-idp\webapp\WEB-INF\web.xml to include:

```
<context-param>
  <param-name>idp.home</param-name>
  <param-value>/opt/shibboleth-idp</param-value>
 </context-param>
```

4. Add **JavaServer Pages Standard Tag Library** (JSTL) Libraries to Tomcat.
   a. Download the latest .jar files:
      - Impl: taglibs-standard-impl-1.2.5.jar (pgp, md5)
      - Spec: taglibs-standard-spec-1.2.5.jar (pgp, md5)
      - EL: taglibs-standard-jstlel-1.2.5.jar (pgp, md5)
      - Compat: taglibs-standard-compat-1.2.5.jar (pgp, md5)

   b. Copy `.jar` files to `opt/shibboleth-idp/edit-webapp/web-inf/lib`.
   c. Open a command prompt window as an administrator.
   d. Navigate to the **Shibboleth bin/directory** and run **build.bat** to generate a new war file.
   e. Open **Services**, restart the **Tomcat service**.
   f. Navigate to https://<server URL>:8443/idp/shibboleth. The IdP metadata page should now display.Configuring the SP Server for the IdP Server

3.3 CONFIGURING THE SP SERVER FOR THE IDP SERVER

DOWNLOAD AND INSTALL APACHE (ON SP SERVER)

Apache is an open-source HyperText Transfer Protocol (HTTP) server for modern operating systems, including UNIX and Windows. The SP is a web server module (installed and configured to work with a web server, such as Apache) that intercepts the user’s request to access a protected resource. Windows may present an error stating that the version of the software is not compatible with your system. If this happens and you have already verified the version downloaded, then re-download it and try installing it again.

1. Install **Apache** from Apache Lounge (Apache Lounge is an organization that provides pre-package installation bundles for Windows systems; this download of Apache will include openssl).
2. Download and install **C++ redistributable library**.
3. Extract the folder to **C:/Apache24**.
   a. Run cmd (Command Prompt) as an Administrator and install Apache 2.4 as a service.

```
httpd.exe -k install
```

   b. Create an `OPENSSL_CONF` Windows System environment variable to point to **openssl.cnf** (located within the Apache bin/install directory).
      i. From the Start Menu, right click **Computer**.
      ii. Click **Properties**.
      iii. Click **Advanced System Settings**.
      iv. Click **Environment Variables**.

v. Under **System Variables**, click **New...**

c. Create an **SSL self-signed certificate** using the following Openssl commands (use "changeit" as a password where needed). Make sure that you input your fully qualified domain name as the common name when you respond to the prompts to create your ssl certificate information:

```
openssl genrsa -des3 -out server.key 1024
openssl req -new -key server.key -out server.csr
copy server.key server.key.org
openssl rsa -in server.key.org -out server.key
openssl x509 -req -days 365 -in server.csr -signkey server.key
-out server.crt
```

d. Create a folder within the Apache install directory named **Certificates**. Move the four files that you just created into this directory from the conf folder (if you sort the conf folder by date modified, these files should come to the top).

e. Open **conf/http.conf** and modify virtual host at port 553 where the SP is called to resemble the following:

```
<VirtualHost *:553>
ServerName <SERVERNAME>
# Include C:\opt\shibboleth-sp\etc\shibboleth\apache24.config
DocumentRoot C:/Apache24/htdocs
ErrorLog C:/Apache24/logs/error_ssl.log
CustomLog C:/Apache24/logs/access_ssl.log combined
SSLEngine on
SSLCertificateFile "C:/Apache24/certificates/server..crt"
SSLCertificateKeyFile "C:/Apache24/certificates/server.key"
</VirtualHost>
```

f. Enable the **mod_ssl.so module** in httd.conf:

```
LoadModule ssl_module modules/mod_ssl.so
```

(uncomment the corresponding LoadModule line)

g. Setup to listen on **port 553**

By adding `Listen 553`

h. Now create a folder within htdocs named **Secure** and place a protected resource within it (in our example, we created secure.html within the folder).

i. Restart **Apache**. At this point, Apache should start without error and you should be able to access the webpage by navigating to:

http://localhost/secure/secure.html

https://localhost:553/secure/secure.html

Now you can access secure.html page without logging in because it is not protected by Shibboleth.

3.4 FINAL IDP CONFIGURATION

1. Navigate to the **IdP metadata** folder (the default path to C:\<shibboleth install directory>\metadata).
2. Modify **idp-metadata.xml**.
3. If necessary, edit all the **Location=** attributes point to IdP app on Tomcat 8 (port 8443) (https://<idp server>:8443/idp...).
4. Modify **conf/metadata-providers.xml**.

   a. Add a similar metadata-provider to the one below for SP metadata provider:

```
<MetadataProvider xsi:type="FilesystemMetadataProvider"
xmlns="urn:mace:shibboleth:2.0:metadata"
id="MyMetadata1"
```

```
metadataFile="C:/opt/shibboleth-idp/metadata/<sp servername>-
metadata.xml" />
```

5. From your browser, launch https://<sp server>/Shibboleth.sso/Metadata.
6. Save the **xml content** that is returned to C:/opt/shibboleth-idp/metadata/<sp server>-metadata.xml.
7. Add connection to your SSL certificate in the ldap.properties file on the IdP server, add certificate file we previously placed on the desktop, and make sure the user and password in the file are the AD Admin user we created in the previous steps.
8. Rebuild the WAR by running **build.bat** again.
9. Restart **Tomcat** service.

3.5 INSTALLING AND CONFIGURING THE SHIBBOLETH SP

1. Download and install **Shibboleth SP** onto your SP server instance with Apache installed on it as instructed by its msi (shibboleth-sp-2.4.2-win32.msi).
2. Install to default location (C:\opt\shibboleth-sp).
3. Navigate to the Shibboleth directory in your file explorer.
4. Modify the **shibboleth2.xml** file.
   a. Remove or comment the "`<InProcess logger="native.logger">…</InProcess>`" block.
   b. Modify <Host> to include hostname
   ```
   <Host name="<SP SERVERNAME>">
   <Path name="secure" authType="shibboleth" requireSession="true"/>
   </Host>
   ```
   c. Provide Application Defaults by modifying the entityID portion of that tag to be the FQDN (Fully Qualified Domain Name).
   d. Provide SSO attributes (do not add port 553 to the following entry)
   ```
   <SSO entityID="https://<IdP SERVERNAME>/idp/shibboleth">
   SAML2 SAML1
   </SSO>
   ```
   e. Enter MetaData provider
   ```
   <MetadataProvider type="XML" file="<IdP SERVERNAME>-
   metadata.xml"/>
   ```
   Once the IdP is a running file, you should be able to navigate to this address and it will return a metadata file: `https://<IdP SERVERNAME>:8443/idp/shibboleth`
   f. Save the metadata (i.e., C:\opt\shibboleth-sp\etc\shibboleth\metadata\my-metadata.xml).
5. Change `showAttributeValues="true"`.
6. Modify the **apache24.config** file. This way you will be able to test the header content.
   a. Replace
   ```
    <Location /secure>
      AuthType shibboleth
      ShibRequestSetting requireSession 1
      require valid-user
    </Location>
   ```
   with
   ```
   <Location /secure>
     AuthType shibboleth
     ShibRequestSetting requireSession 1
     require shib-session
     ShibRequireSession On
     ShibUseHeaders On
   </Location>
   ```
7. Modify the **attribute-map.xml** file.
   a. Add an entry for uid similar to the following (you can copy it from under the LDAP examples)
   ```
   <Attribute name="urn:oid:0.9.2342.19200300.100.1.1" id="uid"/>
   ```
8. Uncomment the following line from **httpd.conf**:
   ```
   Include C:\opt\shibboleth-sp\etc\shibboleth\apache24.config
   ```
9. Go to Windows services and restart the **Shibboleth 2 Daemon(Default)**.
10. Restart the **Apache 2.4 service**.

Testing your configuration

1. Once the IdP and SP have been configured, navigate to https://<your hostname>:553/secure/secure.html. You should see a login screen if it has been configured correctly.
2. Enter **Active Directory account credentials** for a valid user.
3. If you can see the **secure.html** webpage, then you have successfully installed and configured the Shibboleth IdP and SP.

3.6 Configuring Duo as a Second Factor Credential for Shibboleth

1. Sign up for a Duo account.
2. Log in to the Duo Admin Panel and navigate to Applications.
3. Click **Protect an Application** and locate **Shibboleth** in the applications list.
4. Click **Protect this Application** to get your **integration key**, **secret key** and **API hostname**.
5. Generate an **akey**. Your application secret key or akey is a string that you should generate and keep secret from Duo. It should be at least 40 characters long and stored alongside your integration key and secret key.
    a. Create a random string of 40 characters (akey value).
6. Configure **[idp]/conf/authn/duo.properties**
    a. Input the following keys and the API unique to your install.
        i. ikey (idp.duo.integrationKey)
        ii. skey (idp.duo.secretKey)
        iii. akey (idp.duo.applicationKey)
        iv. API hostname (idp.duo.apiHost)
7. Use NTP to ensure that your server's time is correct.
8. Verify that the **properties file** is being referenced by the top level [idp]/conf/idp.properties. The variable idp.additionalProperties should contain **/conf/authn/duo.properties**.
9. `Set idp.authn.identitySwitchIsError = true` in idp.properties.
10. Within idp.properties set `idp.authn.flows = MFA`
11. Navigate to the **[idp]/conf/authn/mfa-authn-config.xml** file and replace the current content with the following:

```
<util:map id="shibboleth.authn.MFA.TransitionMap">
<!-- First rule runs the Password login flow. -->
<entry key="">
<bean parent="shibboleth.authn.MFA.Transition"
p:nextFlow="authn/Password" />
</entry>
<!-- Second rule runs a function if Password succeeds, to determine
whether an additional factor is required. -->
<entry key="authn/Password">
<bean parent="shibboleth.authn.MFA.Transition" p:nextFlowStrategy-
ref="checkSecondFactor" />
</entry>
<!-- An implicit final rule will return whatever the final flow returns.
-->
</util:map>
<!-- Example script to see if second factor is required. -->
<bean id="checkSecondFactor"
parent="shibboleth.ContextFunctions.Scripted" factory-
method="inlineScript">
```

```
<constructor-arg>
<value>
<![CDATA[
nextFlow = "authn/Duo";
nextFlow;   // pass control to second factor or end with the first
]]>
</value>
</constructor-arg>
</bean>
```

12. Place the following beans into **idp]/conf/authn/general-authn.xml**:

```
<bean id="authn/Duo" parent="shibboleth.AuthenticationFlow"
p:forcedAuthenticationSupported="true" p:nonBrowserSupported="false">
<!--
The list below should be changed to reflect whatever locally- or
community-defined values are appropriate to represent MFA. It is
strongly advised that the value not be specific to Duo or any
particular technology.
-->
<property name="supportedPrincipals">
<list>
<bean parent="shibboleth.SAML2AuthnContextClassRef"
c:classRef="http://example.org/ac/classes/mfa" />
<bean parent="shibboleth.SAML1AuthenticationMethod"
c:method="http://example.org/ac/classes/mfa" />
</list>
</property>
</bean>

<bean id="authn/MFA" parent="shibboleth.AuthenticationFlow"
p:passiveAuthenticationSupported="false"
p:forcedAuthenticationSupported="true" p:nonBrowserSupported="false">
<!--
The list below almost certainly requires changes, and should generally
be the
union of any of the separate factors you combine in your particular MFA
flow
rules. The example corresponds to the example in mfa-authn-config.xml
that
combines IPAddress with Password.
-->
<property name="supportedPrincipals">
<list>
<bean parent="shibboleth.SAML2AuthnContextClassRef"
c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTran
sport" />
<bean parent="shibboleth.SAML2AuthnContextClassRef"
c:classRef="urn:oasis:names:tc:SAML:2.0:ac:classes:Password" />
<bean parent="shibboleth.SAML1AuthenticationMethod"
c:method="urn:oasis:names:tc:SAML:1.0:am:password" />
<bean parent="shibboleth.SAML2AuthnContextClassRef"
c:classRef="http://example.org/ac/classes/mfa" />
<bean parent="shibboleth.SAML1AuthenticationMethod"
c:method="http://example.org/ac/classes/mfa" />
</list>
</property>
</bean>
```

13. Download the **Duo Mobile** application onto a mobile device and configure your Duo account for **TOTP/push usage**.
14. Rebuild the WAR by running **build.bat** again.
15. Restart the **Tomcat** service.
16. Try to access the protected resource again. This time, there should be a second factor prompt using Duo.

Duo configuration is now complete. Refer to this page on the Shibboleth wiki for additional information.

STILL NOT SEEING A LOGIN SCREEN?

- Ensure that all entity IDs match up in metadata files.
- Use Test-Shib.org to verify the operation of both SP and IdP.
- We have found that sometimes Java and Tomcat do not automatically install system level variables, and this can cause many issues during the Shibboleth install.
  - Within this guide, we tell you to self-install system level variables.
  - If you may have forgotten to do this, double check that you have followed these steps and/or have the correct variable name/path inputted.
- Use Tomcat 8.5 as opposed to Tomcat 9 (Tomcat 9 appears to have bugs that prevent Shibboleth from working properly).

RESOURCES

Baker, C. (2017, March 10). *Implementing MFA Using Native IDPv3.3 Duo Plugin (for IDPs who upgraded from 2.x)*. Retrieved from wiki.shibboleth.net: https://wiki.shibboleth.net/confluence/pages/viewpage.action?pageId=32112643

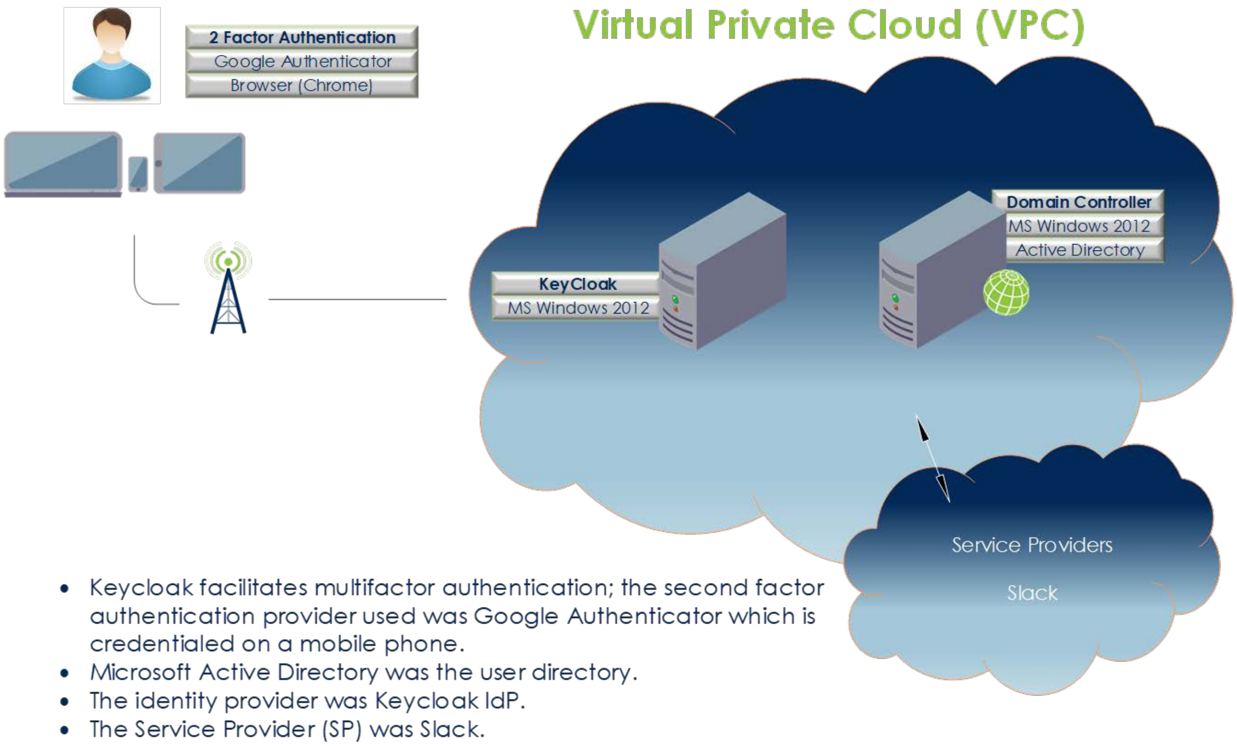
Duo for Shibboleth Identity Provider v3. (n.d.). Retrieved from duo.com: https://duo.com/docs/shibboleth

How Shibboleth Works: Basic Concepts. (n.d.). Retrieved from www.shibboleth.net: https://www.shibboleth.net/index/basic/

User, U. (2017, 8 May). *FlowsAndConfig*. Retrieved from wiki.shibboleth.net: https://wiki.shibboleth.net/confluence/display/CONCEPT/FlowsAndConfig

User, U. (2017, May 8). *Home*. Retrieved from wiki.shibboleth.net: https://wiki.shibboleth.net/confluence/display/CONCEPT

4 KEYCLOAK IMPLEMENTATION GUIDE



- Keycloak facilitates multifactor authentication; the second factor authentication provider used was Google Authenticator which is credentialed on a mobile phone.
- Microsoft Active Directory was the user directory.
- The identity provider was Keycloak IdP.
- The Service Provider (SP) was Slack.

Figure 4 - System Configured for Multifactor Authentication Using Duo Google Authenticator, Keycloak & MS Windows Active Directory.

SUMMARY

Keycloak is an open-source ICAM solution that can facilitate the management and security of users with little to no cost. Keycloak allows for several ICAM methodologies, depending on the needs of your business. It can manage SSO for your business applications and social networks. It can seamlessly connect to existing LDAP and AD servers for federation purposes. It can even serve as a user store to manage the fine-grained authorization policies your organization enacted.

For the purposes of this build, we used Keycloak to establish multifactor authentication and SSO with Google Apps and Slack as the SPs. Although Keycloak can be used for a variety of different SPs, this guide focuses on Slack. The build criteria area is to connect Keycloak via SSO to Slack utilizing Google Authenticator for multifactor authentication.

BEFORE YOU BEGIN – A LIST OF HELPFUL TIPS

- Make sure you have Java installed and your JAVA_HOME variable is set
  - Open the start menu and select “View Advanced System Settings”
  - Select environment variables and create/edit JAVA_HOME to point to where your JDK is located.
- Keycloak and Commercial SPs require HTTPS communication. Before starting, obtain a certificate for the host name you are utilizing and create a JKS file for it. Instructions for this procedure is here.

- Create a cname DNS entry that points to your Keycloak server IP Address. This will allow you to access the Administration Console from your computer after initial configuration.
- Install either Firefox or Chrome on the Keycloak server. The admin console interface does not work with IE.

4.1 Keycloak Install and Initial Configuration

1. Download the **Keycloak Server** software from here. The software is delivered as a .zip file and contains both the IdP software and the server component that hosts it.
2. Unzip the **Keycloak file** to a directory of your choice.
3. Edit the configuration file **standalone.xml** located in the [Keycloak]/standalone/configuration directory.
4. If you are hosting the system in a cloud platform, you will need to update the **jboss.bind.address** value. Locate the code snippet and update that value with the VM instance IP address. Example below:

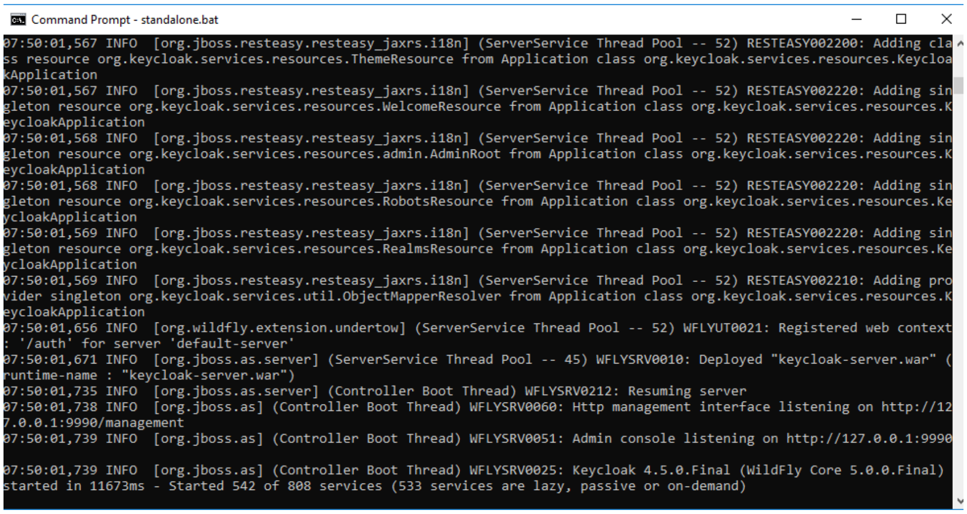
   Original:

   ```
   <interface name="public">
           <inet-address value="${jboss.bind.address:127.0.0.1}"/>
   </interface>
   ```

   Updated:

   ```
   <interface name="public">
           <inet-address value="${jboss.bind.address:10.10.10.10}"/>
   </interface>
   ```

5. If you are using an external DNS you will need to change 3 lines of code
   a. Change the “property name” value from “localhost” to your specific hostname
      i. Ex: <property name="hostname" value="<enter your hostname>"/>
   b. Change the “host name” alias from “localhost” to your specific hostname
      i. Ex: <host name="default-host" alias="<enter your hostname>">
   c. Change the “remote destination host” from localhost to your hostname
      i. Ex: <remote-destination host="<enter your hostname>" port="25"/>
6. Using the command prompt, navigate to the **Keycloak/bin** folder. Run the **standalone.bat** command to start the server. Keycloak does not have a built-in shutdown script. To shut down the Keycloak server, you will need to Ctrl-C in the command prompt window.



```
07:50:01,567 INFO  [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 52) RESTEASY002200: Adding cla
ss resource org.keycloak.services.resources.ThemeResource from Application class org.keycloak.services.resources.Keycloa
kApplication
07:50:01,567 INFO  [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 52) RESTEASY002220: Adding sin
gleton resource org.keycloak.services.resources.WelcomeResource from Application class org.keycloak.services.resources.K
eycloakApplication
07:50:01,568 INFO  [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 52) RESTEASY002220: Adding sin
gleton resource org.keycloak.services.resources.admin.AdminRoot from Application class org.keycloak.services.resources.K
eycloakApplication
07:50:01,568 INFO  [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 52) RESTEASY002220: Adding sin
gleton resource org.keycloak.services.resources.RobotsResource from Application class org.keycloak.services.resources.Ke
ycloakApplication
07:50:01,569 INFO  [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 52) RESTEASY002220: Adding sin
gleton resource org.keycloak.services.resources.RealmsResource from Application class org.keycloak.services.resources.Ke
ycloakApplication
07:50:01,569 INFO  [org.jboss.resteasy.resteasy_jaxrs.i18n] (ServerService Thread Pool -- 52) RESTEASY002210: Adding pro
vider singleton org.keycloak.services.util.ObjectMapperResolver from Application class org.keycloak.services.resources.K
eycloakApplication
07:50:01,656 INFO  [org.wildfly.extension.undertow] (ServerService Thread Pool -- 52) WFLYUT0021: Registered web context
: '/auth' for server 'default-server'
07:50:01,671 INFO  [org.jboss.as.server] (ServerService Thread Pool -- 45) WFLYSRV0010: Deployed "keycloak-server.war" (
runtime-name : "keycloak-server.war")
07:50:01,735 INFO  [org.jboss.as.server] (Controller Boot Thread) WFLYSRV0212: Resuming server
07:50:01,738 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0060: Http management interface listening on http://12
7.0.0.1:9990/management
07:50:01,739 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0051: Admin console listening on http://127.0.0.1:9990

07:50:01,739 INFO  [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: Keycloak 4.5.0.Final (WildFly Core 5.0.0.Final)
started in 11673ms - Started 542 of 808 services (533 services are lazy, passive or on-demand)
```

7. You will need to create an admin account to access the admin console. Open a new command windows and navigate to the **Keycloak\bin** folder. Run the below script with the required username and password:

   add-user-keycloak.bat -r master -u <username> -p <password>

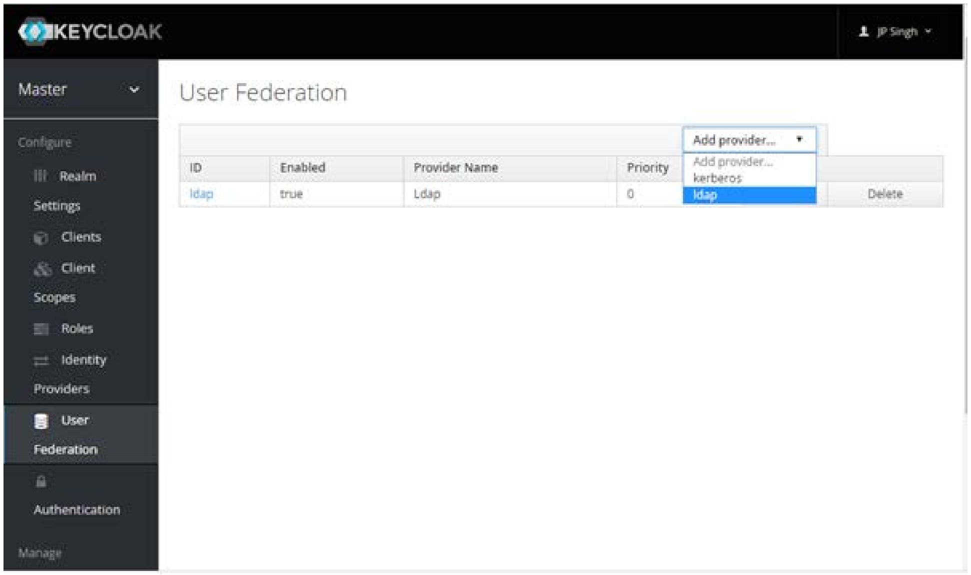
8. Verify access to admin console by logging in at URL *YourLocalIP*:8080/auth from the server.

4.2 Keycloak User Federation and Configuration

The Keycloak configuration and administration console is where all configurations are stored. The admin console can be accessed on the server using the URL *YourLocalIP*:8080/auth/admin. The Keycloak admin console can be accessed externally from https://*Hostname*:8443/auth/admin.

To configure a User Federation and import its LDAP users:

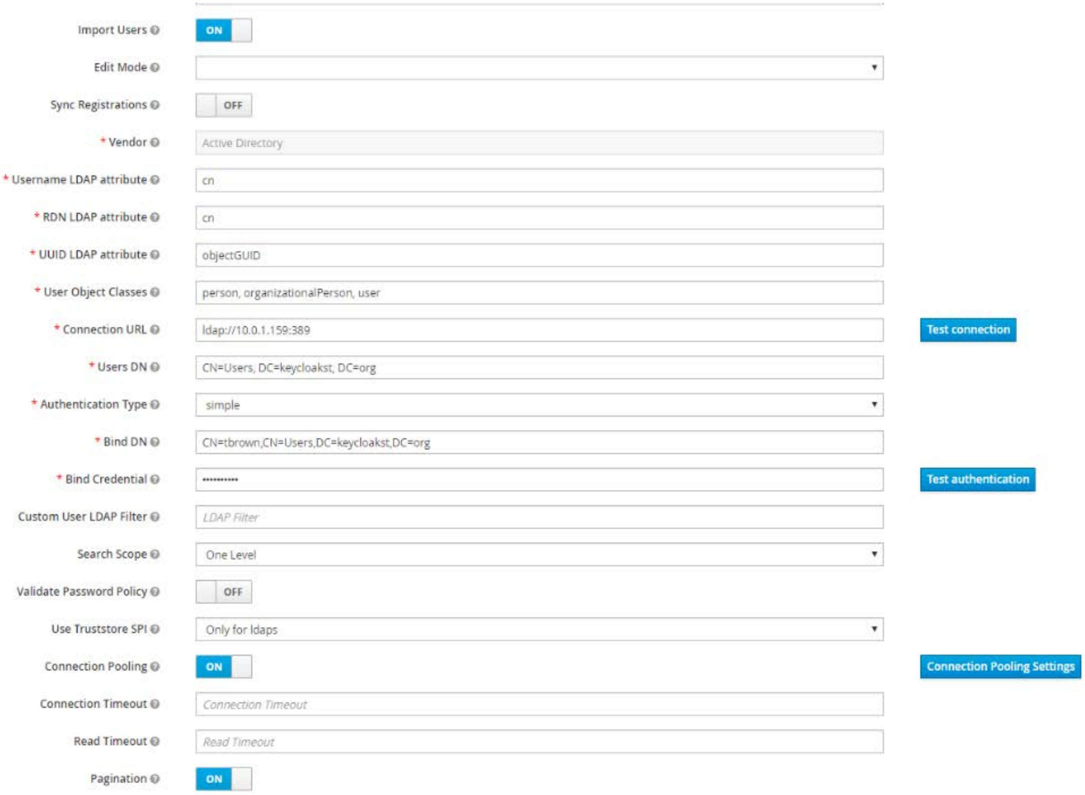
1. Log into the **Admin Console** and click **User Federation** from the left navigation.
2. Click the **Add Provider** button on the top right of the console and select **ldap**. The window will refresh and display all the fields necessary to configure access to the LDAP.



ID	Enabled	Provider Name	Priority		
ldap	true	Ldap	0		Delete

3. Configure the **fields** to access your **LDAP service**. An example is given below to connect to a Microsoft AD instance for an AD forest named keycloakst.
   - Set Sign Assertions: ON
   - Set Force Name ID Format: ON
   - Set Name ID Format: email
   - Username LDAP attribute: cn
   - RDN LDAP attribute: cn
   - UUID LDAP attribute: objectGUID
   - User Object Classes: person, organizationalPerson, user
   - Connection URL: <URL to your AD server> ex: ldap://10.0.X.XXX:389
   - Users DN: CN=Users, DC=keycloakst, DC=org
   - Authentication Type: simple
   - Bind DN: distinct name of your LDAP admin ex: admin@keycloakst.org
   - Bind Credential: <password of admin account>

All other fields can be left as default. There are test buttons associated with the Connection URL and Bind Credential fields to verify connection to the LDAP.



Import Users	ON
Edit Mode	
Sync Registrations	OFF
* Vendor	Active Directory
* Username LDAP attribute	cn
* RDN LDAP attribute	cn
* UUID LDAP attribute	objectGUID
* User Object Classes	person, organizationalPerson, user
* Connection URL	ldap://10.0.1.159:389
* Users DN	CN=Users, DC=keycloakst, DC=org
* Authentication Type	simple
* Bind DN	CN=tbrown,CN=Users,DC=keycloakst,DC=org
* Bind Credential	••••••••••
Custom User LDAP Filter	LDAP Filter
Search Scope	One Level
Validate Password Policy	OFF
Use Truststore SPI	Only for ldaps
Connection Pooling	ON
Connection Timeout	Connection Timeout
Read Timeout	Read Timeout
Pagination	ON

4. Press **Save** at the bottom when finished the page will reload.
5. Scroll to the bottom of the page and click **Synchronize all users**. You will see a confirmation message when the process is complete
6. Click **Users** from the left navigation
7. Click **View All Users** to ensure that the User Federation has updated the user store appropriately

4.3 ADDING A CLIENT TO KEYCLOAK

Keycloak refers to service providers as “Clients.” In our case the SP we are using is Slack utilizing SAML for our client protocol.

Part 1 on Keycloak

1. Log into Keycloak with an administrator account
2. Create a new **Realm** to manage your clients and associated users.
3. Roll over **Master** in the top left corner and click **Add Realm**.
4. Give the Realm a **Name**. Realm names must not contain spaces. Click **Create**. The page will reload.
5. Select the **Login** tab on the top navigation.
6. Set User Registration to **ON**. Set Require SSL to **none**. Click **Save**.
7. Select the **Tokens** tab from the top navigation.
8. Change Default Signature Algorithm to **RS256**. Click **Save**.
9. Select the **Client Registration** tab from the top navigation.
10. Click **Create** in the **Initial Access Tokens** tab.
11. Leave the fields as default. Click **Save**.
12. Click **Client** from the left navigation and click **Create**.
13. Fill in the Client ID section with the **URL** of your service provider (e.g. https://wso2st.slack.com) and specify the **client protocol** (saml) from the dropdown menu. Leave the Client SAML Endpoint blank.
14. Click **Save**. The page will refresh automatically with additional fields.
15. Verify that the Signature Algorithm field matches the Default Signature Algorithm you configured earlier. This can be checked under Realm Settings.
16. Change SAML Signature Key Name to **NONE**.
17. Update Name ID Format to **email**.
18. Fill in the **Valid Redirect URI** field. This should be the same URL as your service provider. See the example below. The /* at the end of the URL is required.

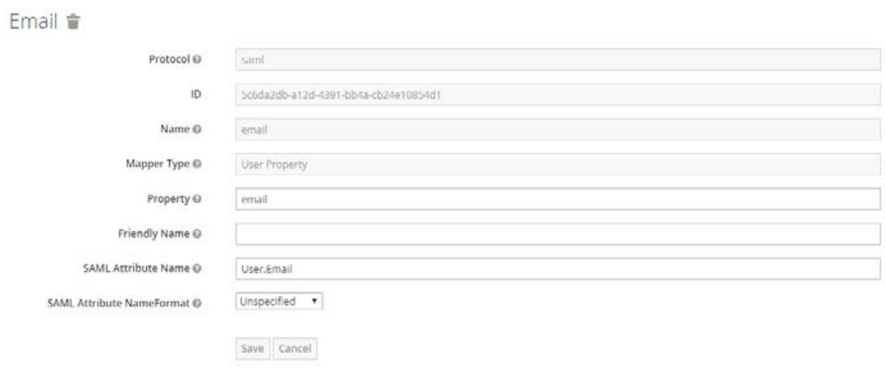
    Ex: https://wso2st.slack.com/*
19. Specify a **Master SAML Processing URL**. This should be your client url followed by /sso/saml.

    Ex: https://wso2st.slack.com/sso/saml
20. Click **Save**.
21. Select the **Mappers** tab from the top navigation. Click **Create** to configure the user attributes.
22. Click on the **Mapper Type** field and select **User Property** from the dropdown. New fields will appear.
23. Configure the new fields as indicated below:

    Property: email

    SAML Attribute Name: User.Email

    SAML Attribute NameFormat: Unspecified



Email

Protocol: saml

ID: 5c6da2db-a12d-4391-bb4a-cb24e10854d1

Name: email

Mapper Type: User Property

Property: email

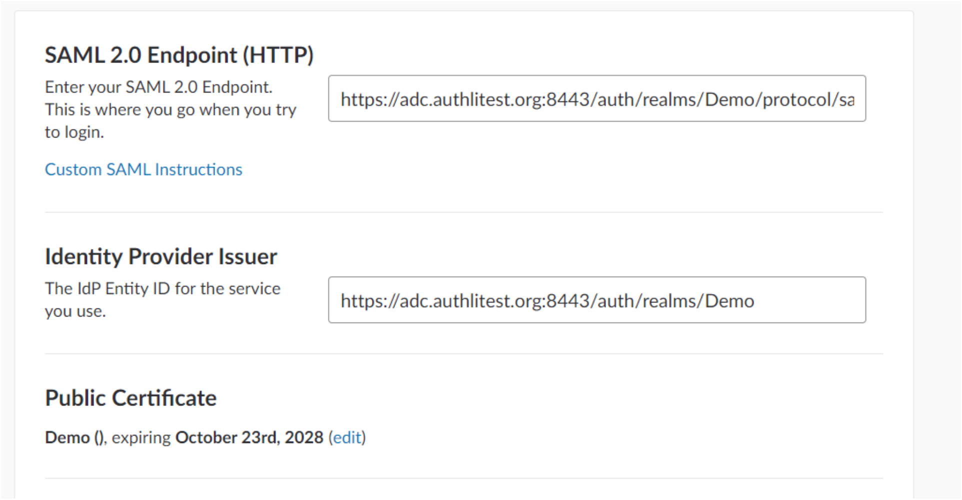
Friendly Name:

SAML Attribute Name: User.Email

SAML Attribute NameFormat: Unspecified

4.4 Configure Keycloak as the MFA Provider for Slack

1. Sign into your Slack instance with an administrator account.
2. Navigate to **Administration - Workplace** settings.
3. Select the **Authentication** tab.
4. Click **Configure** next to the SAML Authentication option.
5. For **SAML 2.0 Endpoint (HTTP)** enter in the public URL for your Keycloak instance followed by /auth/realms/<the realm you created>/protocol/saml/
6. For **Identity provider Issuer** enter the URL of Keycloak followed by /auth/realms/<realm you created>
7. Obtain the public certificate from Keycloak. Log in to the Keycloak admin console and navigate to the **Keys** tab from the top navigation of the Realm Settings.
8. Click **Certificate**. Copy and Paste the **certificate data** into the **Public Certificate** field in Slack.



SAML 2.0 Endpoint (HTTP)

Enter your SAML 2.0 Endpoint. This is where you go when you try to login.

https://adc.authlitest.org:8443/auth/realms/Demo/protocol/sa

Identity Provider Issuer

The IdP Entity ID for the service you use.

https://adc.authlitest.org:8443/auth/realms/Demo

Public Certificate

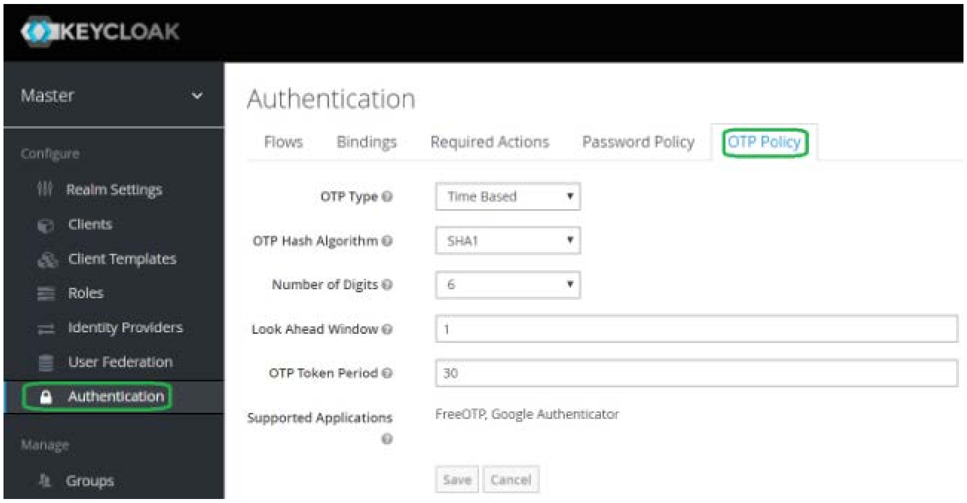
Demo (), expiring October 23rd, 2028 (edit)

9. Under the **Advanced Options** tab, click **Expand** and unselect **Sign Authnrequest** and **Sign Responses**. The **Service Provider Issuer URL** should be the URL of your Slack instance (e.g. wso2st.slack.com).

4.5 ENABLING MULTIFACTOR AUTHENTICATION USING KEYCLOAK

1. Open Keycloak admin page.
2. Click on **Authentication** from the left navigation.
3. In the default **Flows** tab, set OTP form to **Required**.
4. Click on the **OTP Policy** tab from the top navigation.
5. Set **OTP Type** to **Time Based** and update **Look Ahead Window** to **3**.
6. Click **Save**.

Multifactor authentication will now be required for all users accessing the Service Provider (e.g. Slack). See below for instructions on configuring Google Authenticator as the second factor.



OTP Type: Time Based

OTP Hash Algorithm: SHA1

Number of Digits: 6

Look Ahead Window: 1

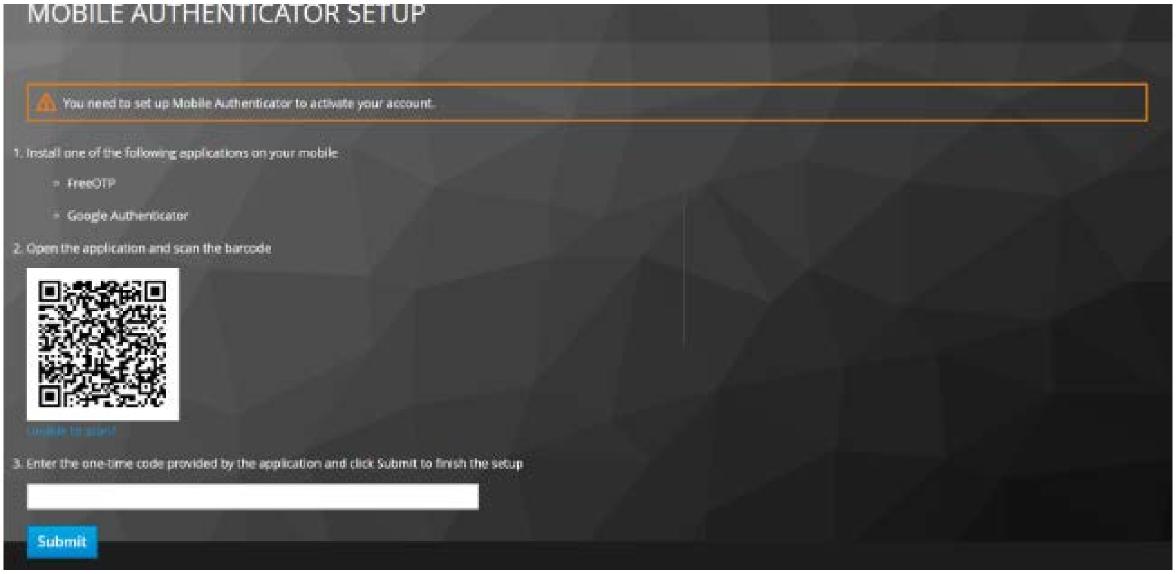
OTP Token Period: 30

Supported Applications: FreeOTP, Google Authenticator

Configure Google Authenticator for User in Slack

1. Download **Google Authenticator** onto the user's phone.
2. Follow the screen prompts.
3. Log into **Slack**. If you are already logged in, sign out and sign back in.

4. After signing in with the username and password, have the user **scan** the one-time **Keycloak QR code** via Google Authenticator.



MOBILE AUTHENTICATOR SETUP

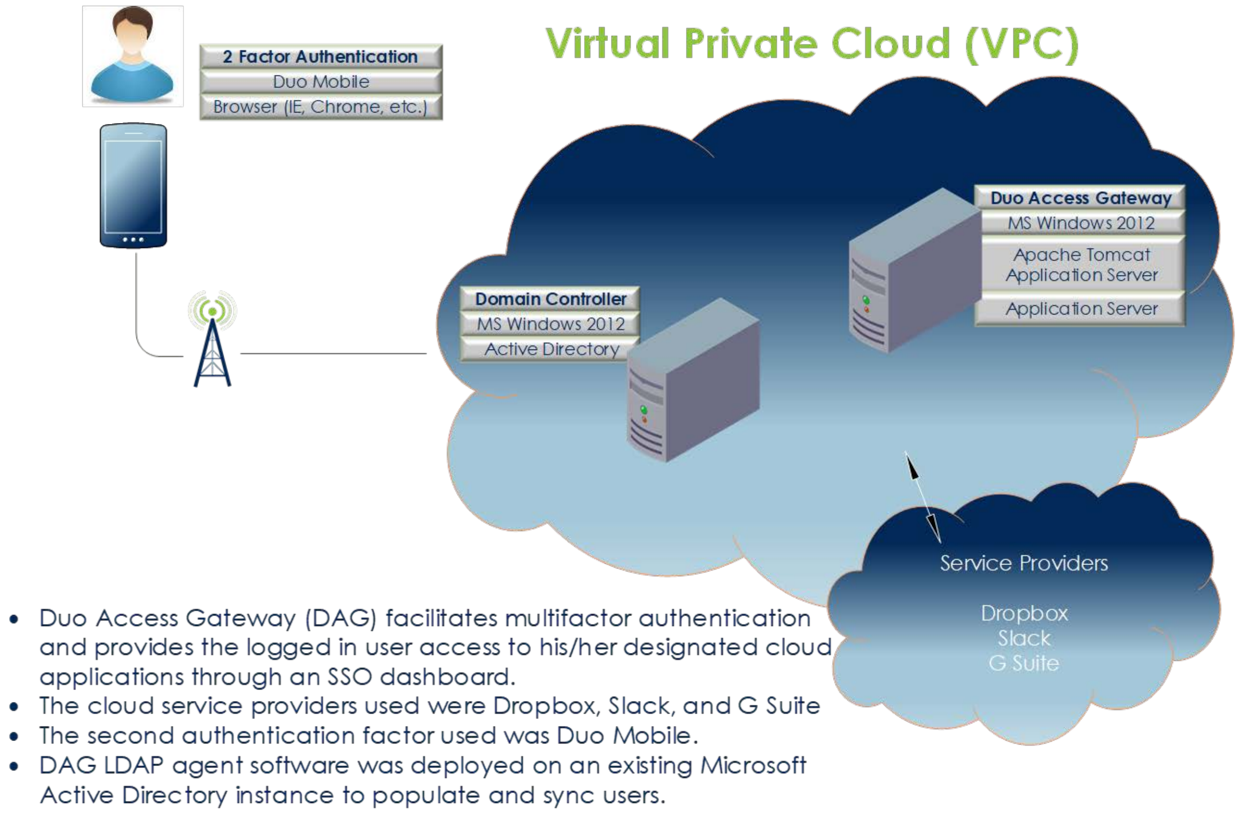
⚠ You need to set up Mobile Authenticator to activate your account.

1. Install one of the following applications on your mobile
   - FreeOTP
   - Google Authenticator
2. Open the application and scan the barcode
3. Enter the one-time code provided by the application and click Submit to finish the setup

5. Enter the code provided by Google Authenticator into Keycloak.

The user will now be prompted to provide a Google Authenticator token upon signing into SP.

5 Duo Access Gateway Implementation Guide



- Duo Access Gateway (DAG) facilitates multifactor authentication and provides the logged in user access to his/her designated cloud applications through an SSO dashboard.
- The cloud service providers used were Dropbox, Slack, and G Suite
- The second authentication factor used was Duo Mobile.
- DAG LDAP agent software was deployed on an existing Microsoft Active Directory instance to populate and sync users.

Figure 5 - Duo Access Gateway SSO

Duo Access Gateway (DAG) adds two-factor authentication, complete with inline self-service enrollment, to popular cloud services. DAG's capabilities include SSO and Microsoft AD/LDAP integration. This build was performed in an AWS environment and used AD as its connected user source. For the purposes of this build, DAG facilitated multifactor authentication and provided a user-friendly SSO dashboard with access to Slack, Dropbox and G Suite SPs. Duo Mobile was used for the second authentication factor and each SP was configured for SSO using security assertion markup language (SAML) 2.0 protocol.

Because DAG has robust, publicly available documentation, this Implementation Guide provides links that send the reader to the Duo website as a reference for most of the build steps.

BEFORE YOU BEGIN – A LIST OF HELPFUL TIPS

- Ensure that all pre-configuration is accurate to avoid errors throughout setup process.
- For many commercial applications, organizations need to have the right commercial plan to implement SSO.
- Remove any conflicting metadata that may interfere with configuration.

TO BEGIN

Before setting up DAG, organizations should first review all documentation to fully understand the system requirements and prerequisites required for a successful implementation of the software. A good starting point can be found on the Duo Access Gateway for Windows Overview page.

5.1 Install and Configure DAG

Before installation, DAG installer verifies the system prerequisites and exits if any are missing. Installation and configuration documentation, including steps for configuring the organization's authentication source, can be found under the Install Duo Access Gateway header, on the Duo Access Gateway for Windows page.

5.2 Connect Cloud Applications

Create, add and configure your cloud applications to the DAG.

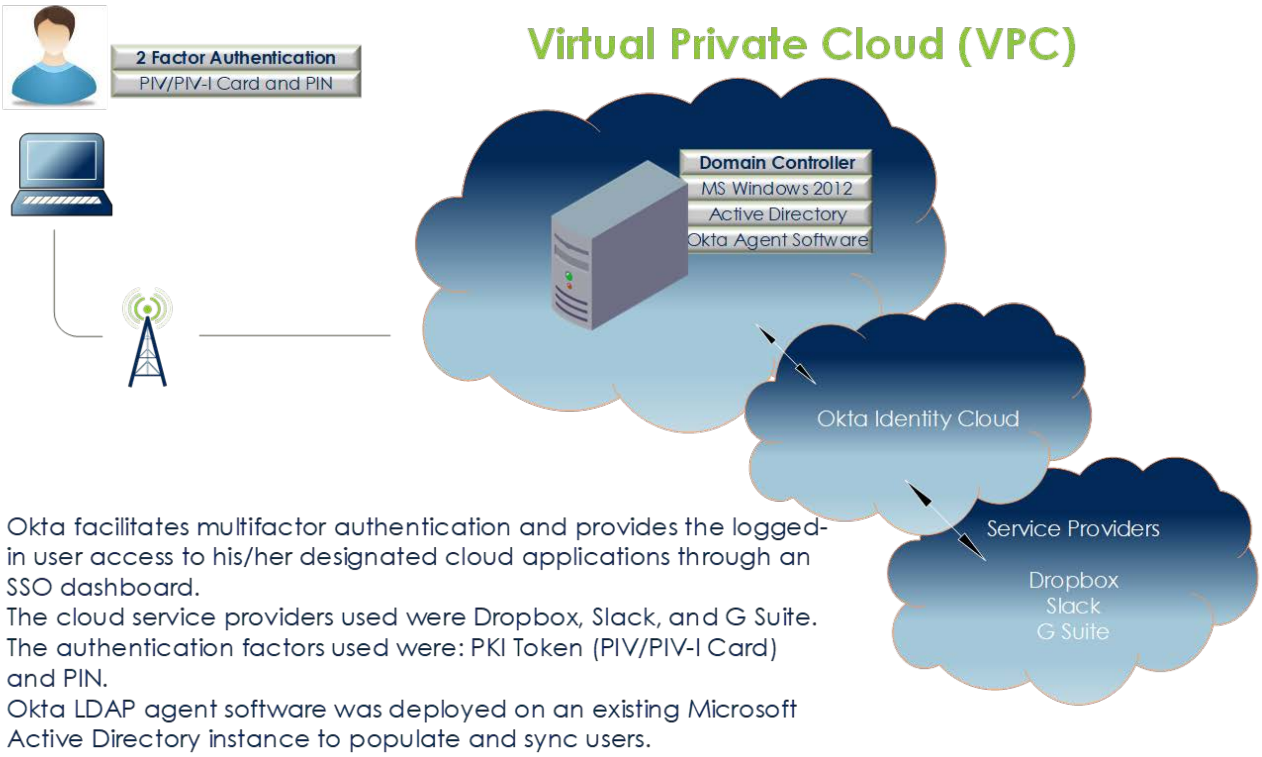
5.3 Configure DAG for Multifactor Authentication

DAG allows for an organization's users to self-enroll in multifactor authentication with their own factors. An organization must first enable the self-service portal. Once enabled, an organization's users can register their devices and other factors for multifactor authentication; documentation can be found on the Duo Guide to Two-Factor Authentication page under the Adding a new device header.

5.4 Enable DAG Launcher Application

Duo Access Gateway Launcher is an application internal to DAG that, when enabled, provides a user-friendly SSO portal from which users can access their personal DAG-protected service provider applications. The Launcher itself can also be configured to require multifactor authentication. The Launcher was set up and used as part of this build. Specific information on the configuration of DAG Launcher can be found on the Duo Access Gateway for Windows Overview page under the Enable the Duo Access Gateway Launcher.

6 OKTA IDENTITY CLOUD IMPLEMENTATION GUIDE



- Okta facilitates multifactor authentication and provides the logged-in user access to his/her designated cloud applications through an SSO dashboard.
- The cloud service providers used were Dropbox, Slack, and G Suite.
- The authentication factors used were: PKI Token (PIV/PIV-I Card) and PIN.
- Okta LDAP agent software was deployed on an existing Microsoft Active Directory instance to populate and sync users.

Figure 6 - Okta Identity Cloud SSO Build Architecture

Okta Identity Cloud (Okta) is a cloud-based IdP that integrates with on-premises and other cloud-based applications. Okta's capabilities include provisioning, SSO and Microsoft AD/LDAP integration. The build created by our engineering team used AD within an AWS environment as its user source. The users on the domain controller were synced to Okta Identity Cloud via the Okta agent software. Okta facilitated multifactor authentication and provided an SSO dashboard with user access to Slack, Dropbox and G Suite SPs. A personal identity verification interoperable (PIV-I) card with a PIN was used as the second authentication factor, and each SP was configured for SSO using SAML 2.0 protocol.

Okta has a robust library of documentation, so instead of specifying every detail, this implementation guide discusses the proper order for installation and configuration, with supplemental links to the relevant Okta pages.

BEFORE YOU BEGIN – A LIST OF HELPFUL TIPS

- Familiarize yourself with Okta terminology through the Okta Help Center, Okta Terminology page.
- Remove any conflicting metadata that may interfere with configuration.
- Customize your organization's parameter values before configuring SAML 2.0.
- Enable the "Multiple Sign-In" feature in G-Suite and use a "Backdoor URL" to allow the administrator access without a SAML assertion.[7]
- Okta supports PIV and PIV-I cards, but that feature needs to be enabled by Okta.

TO BEGIN

Okta is a cloud-based identity management solution. Once a plan is purchased, Okta sends the organization's administrator a link to the application. A good place to begin is on the Okta Help Center's Getting Started as a New Okta Administrator page.

CONNECTING A DOMAIN CONTROLLER

If you do not already have a domain controller installed and configured, follow Microsoft's instructions on how to configure the domain controller.

Once your domain controller is configured and installed for your organization, follow Okta's installation and configuration guide for the Okta Active Directory Agent, found on the Okta Help Center's Install and Configure the Okta Active Directory Agent page.

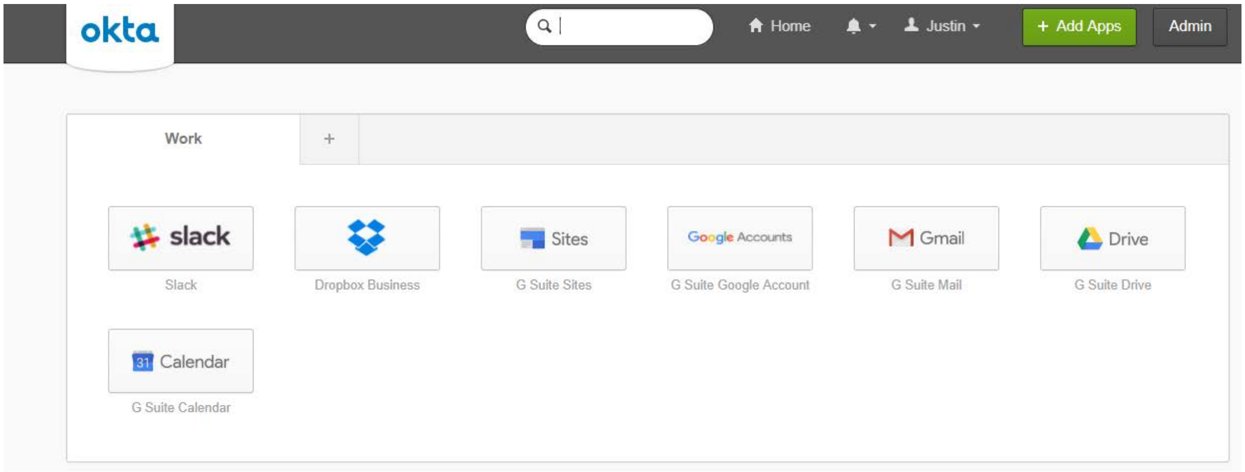
6.1 Service Provider Configuration

Now that Okta is connected to your domain controller, you need to configure your service providers. The three selected below are available, along with thousands of pre-integrated apps.

1. **Slack Configuration**
   a. Configure Slack for SAML 2.0. Documentation can be found on Okta's How to Configure SAML 2.0 for Slack page.
   b. Configure provisioning within Okta for Slack. Documentation can be found on Okta's Configuring Provisioning for Slack page.
2. **Dropbox Configuration**
   a. Configure Dropbox. Documentation can be found on Okta's Configuring Provisioning for Dropbox Business page.
3. **G Suite Configuration**
   a. Configure G Suite for SAML 2.0. Documentation can be found on Okta's How to Configure SAML 2.0 for G Suite.
   b. Configure provisioning within Okta for G Suite. Documentation can be found on Okta Help Center's Configuring Provisioning for G Suite page.

[7] https://support.okta.com/help/Documentation/Knowledge_Article/Google-Apps-Deployment-Guide

Once everything is properly configured, your Okta dashboard will look like this:



6.2 PIV-I CONFIGURATION

Once your service providers are configured, the addition of a second authentication factor is strongly recommended before users obtain access to Okta. We used a PIV-I card. Steps for PIV configuration can be found on Okta's Identify Providers page (click on “Add a PIV card”).

CHANGE MANAGEMENT

Okta Help Center's End User Adoption Toolkit provides helpful information to download the End User Adoption kit.