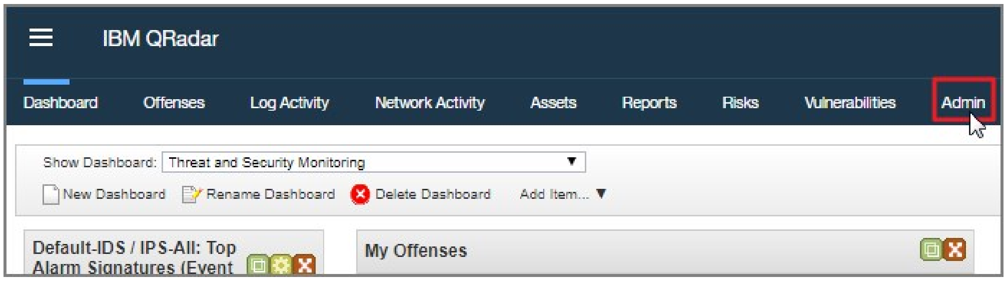
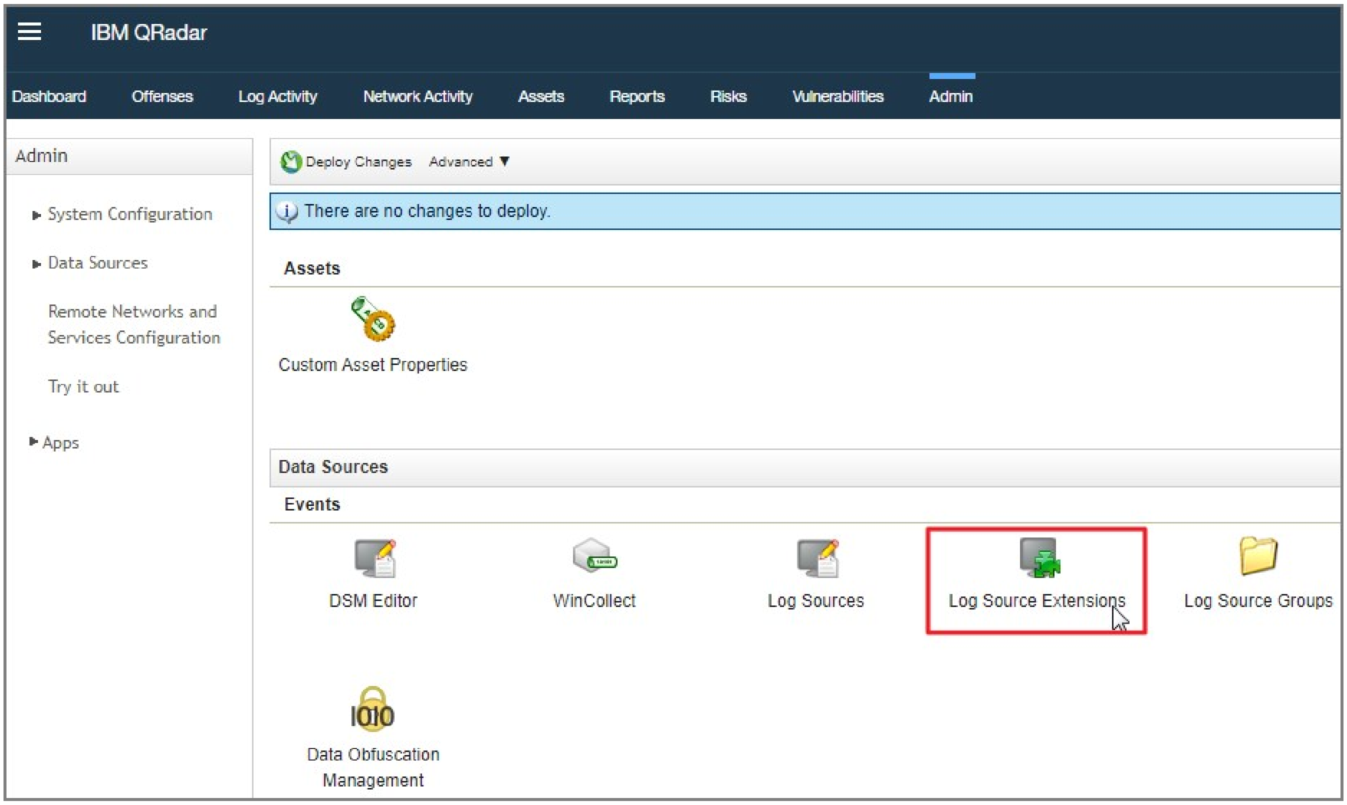
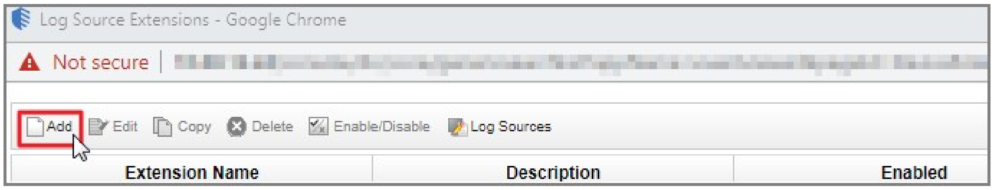


4. Click on **Log Source Extensions**.



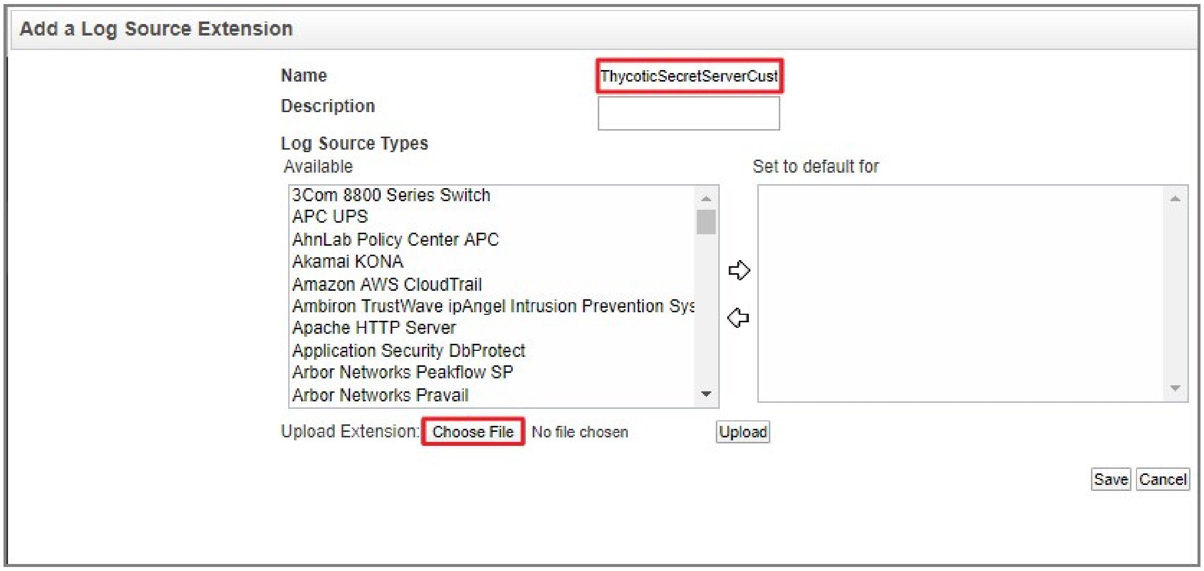
There are no changes to deploy.

5. Click on **Add**.



6. Add a name and description for the Log Source Extension.

7. Click **Choose File**.



Add a Log Source Extension

Name: ThycoticSecretServerCust

Description:

Log Source Types

Available: 3Com 8800 Series Switch, APC UPS, AhnLab Policy Center APC, Akamai KONA, Amazon AWS CloudTrail, Ambiron TrustWave ipAngel Intrusion Prevention Sys, Apache HTTP Server, Application Security DbProtect, Arbor Networks Peakflow SP, Arbor Networks Pravail

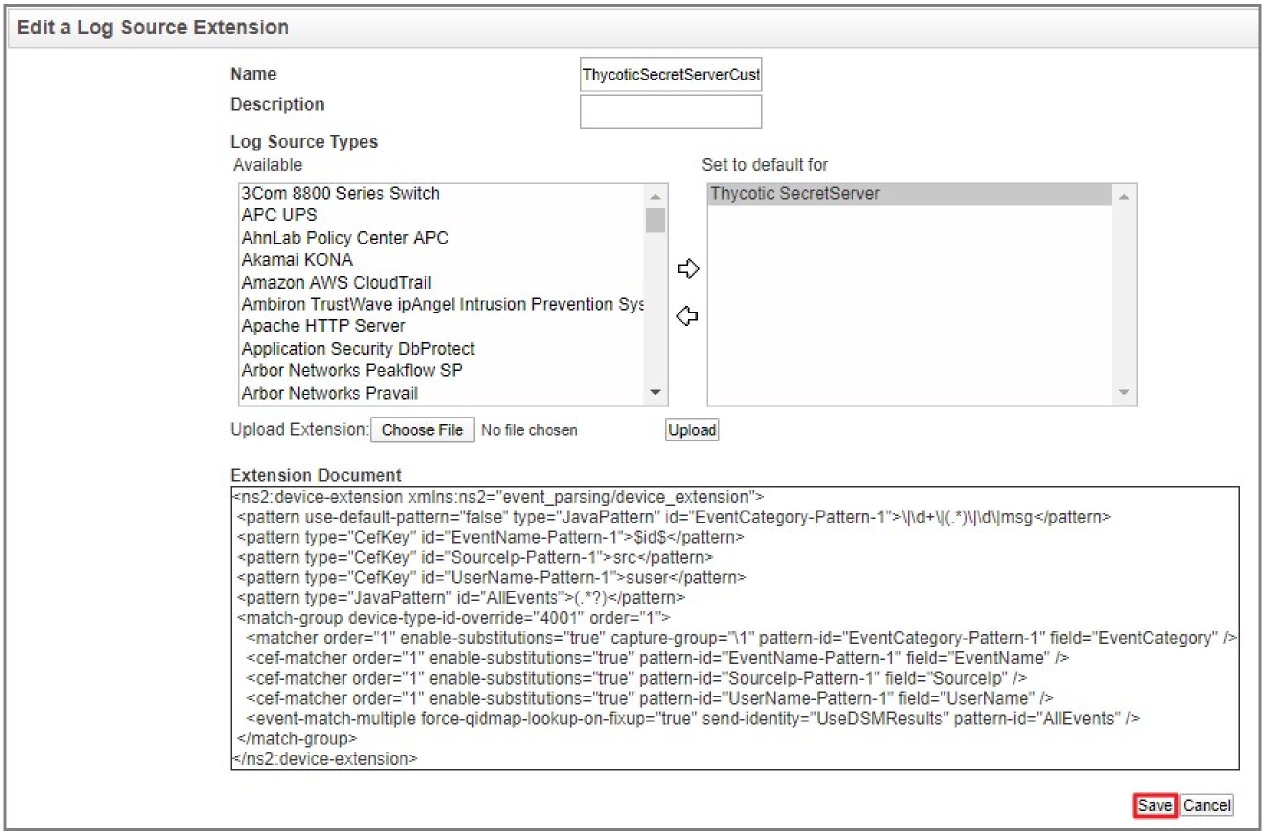
Set to default for:

Upload Extension: No file chosen

8. Select the xml file you created in step 1 with the provided example.

9. Click **Upload**.

10. Select the log source extension and **set it to the default**.



Edit a Log Source Extension

Name: ThycoticSecretServerCust

Description:

Log Source Types

Available: 3Com 8800 Series Switch, APC UPS, AhnLab Policy Center APC, Akamai KONA, Amazon AWS CloudTrail, Ambiron TrustWave ipAngel Intrusion Prevention Sys, Apache HTTP Server, Application Security DbProtect, Arbor Networks Peakflow SP, Arbor Networks Pravail

Set to default for: Thycotic SecretServer

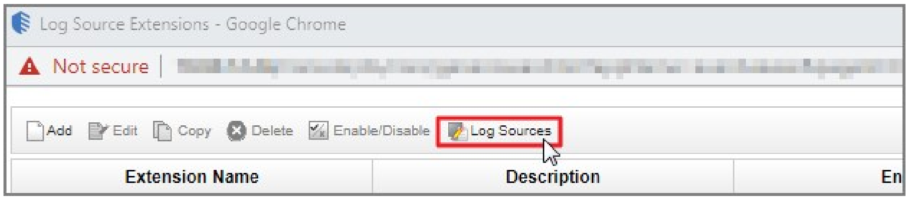
Upload Extension: No file chosen

Extension Document

```
<ns2:device-extension xmlns:ns2="event_parsing/device_extension">
 <pattern use-default-pattern="false" type="JavaPattern" id="EventCategory-Pattern-1">\|\d+\|(.*)\|\d\|msg</pattern>
 <pattern type="CefKey" id="EventName-Pattern-1">$id$</pattern>
 <pattern type="CefKey" id="SourceIp-Pattern-1">src</pattern>
 <pattern type="CefKey" id="UserName-Pattern-1">suser</pattern>
 <pattern type="JavaPattern" id="AllEvents">(.*?)</pattern>
 <match-group device-type-id-override="4001" order="1">
  <matcher order="1" enable-substitutions="true" capture-group="\1" pattern-id="EventCategory-Pattern-1" field="EventCategory" />
  <cef-matcher order="1" enable-substitutions="true" pattern-id="EventName-Pattern-1" field="EventName" />
  <cef-matcher order="1" enable-substitutions="true" pattern-id="SourceIp-Pattern-1" field="SourceIp" />
  <cef-matcher order="1" enable-substitutions="true" pattern-id="UserName-Pattern-1" field="UserName" />
  <event-match-multiple force-qidmap-lookup-on-fixup="true" send-identity="UseDSMResults" pattern-id="AllEvents" />
 </match-group>
</ns2:device-extension>
```

11. Click **Save**.

12. Click on **Log Sources**.

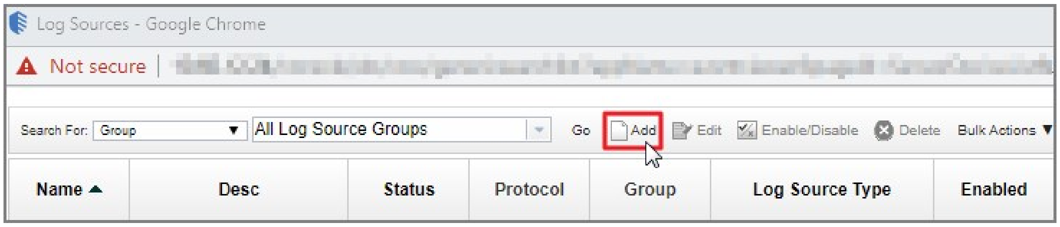


Log Source Extensions - Google Chrome

Not secure

Extension Name	Description	En

13. Click **Add**.



Log Sources - Google Chrome

Not secure

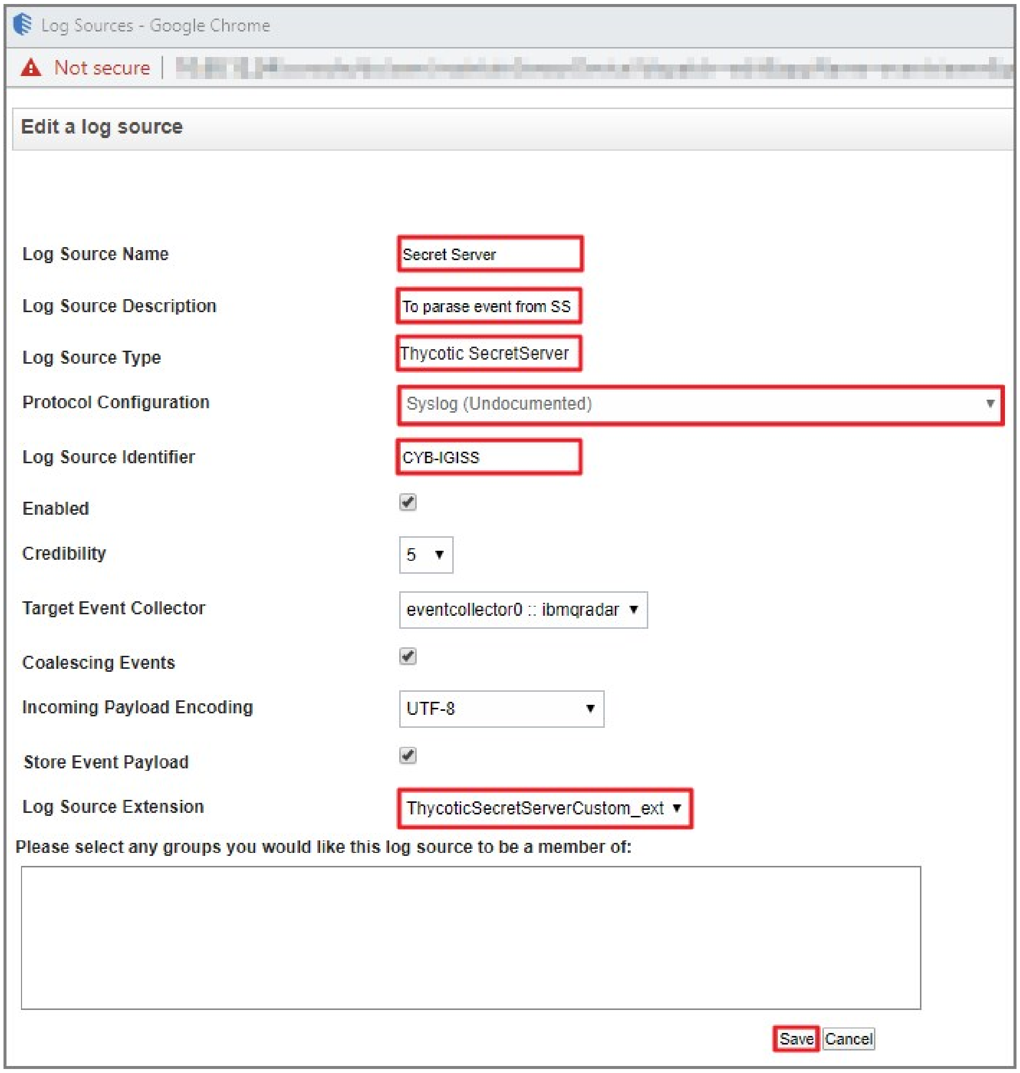
Search For: Group ▼ All Log Source Groups

Name ▲	Desc	Status	Protocol	Group	Log Source Type	Enabled

14. Fill in the required fields:

    - **Log Source Name**
    - **Log Source Description**
    - **Log Source Type**
    - **Protocol Configuration**
    - **Log Source Identifier**
    - **Log Source Extension**: Click the drop-down and choose your custom-built parser (the xml file that was saved as a log source extension).

15. Click **Save**.



Edit a log source

Log Source Name: Secret Server

Log Source Description: To parase event from SS

Log Source Type: Thycotic SecretServer

Protocol Configuration: Syslog (Undocumented) ▾

Log Source Identifier: CYB-IGISS

Enabled: ☑

Credibility: 5 ▾

Target Event Collector: eventcollector0 :: ibmqradar ▾

Coalescing Events: ☑

Incoming Payload Encoding: UTF-8 ▾

Store Event Payload: ☑

Log Source Extension: ThycoticSecretServerCustom_ext ▾

Please select any groups you would like this log source to be a member of:

QID Mappings

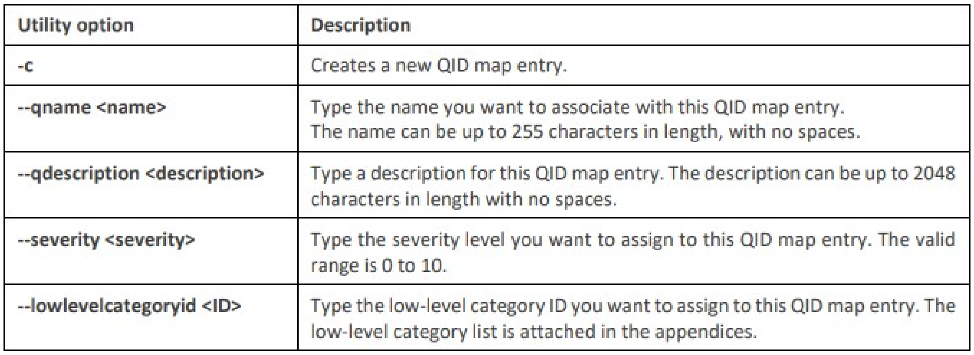
The QID or QRadar Identifier is what QRadar uses to give events their name, high-level category and lowlevel category.

1. We now need to create custom QIDs. Do this by SSH-ing into the QRadar console, changing the directory to **/opt/QRadar/bin** and running the following command:

```
./qidmap_cli.sh -c --qname <name> --qdescription <description> --severity <severity> --lowlevelcategoryid <ID>
```

For Example:

```
./qidmap_cli.sh -c --qname “USER – LOGIN” --qdescription “A user as logged in.” --severity 1 –lowlevelcategoryid 19001
```



Utility option	Description
-c	Creates a new QID map entry.
--qname <name>	Type the name you want to associate with this QID map entry. The name can be up to 255 characters in length, with no spaces.
--qdescription <description>	Type a description for this QID map entry. The description can be up to 2048 characters in length with no spaces.
--severity <severity>	Type the severity level you want to assign to this QID map entry. The valid range is 0 to 10.
--lowlevelcategoryid <ID>	Type the low-level category ID you want to assign to this QID map entry. The low-level category list is attached in the appendices.

2. Alternatively you could use a csv list as demonstrated in the in the **Import List** section and use it with the following command to import several QIDs at once:

```
/opt/QRadar/bin/qidmap_cli.sh -i -f <filename.txt>
```

19001 is used for most of the low-level category IDs as an example.

3. Using the program **sendnow**, send the list of all events to your QRadar box in the .txt file you named (example: **tss events all.txt**) to generate every possible event. The events can be found in the Event List section below.

Event List

CONFIGURATION - EDIT	The main Thycotic Secret Server configuration has been edited	10	19001
FOLDER - CREATE	A Folder has been created	2	19001
FOLDER - DELETE	A Folder has been deleted	5	19001
FOLDER - EDIT PERMISSIONS	The configuration has been edited	10	19001
FOLDER - SECRET POLICY CHANGE	The policy assigned to a folder has been changed	6	19001
FOLDER - SECRET POLICY CHANGE	The Secret policy assigned to a folder has been changed	8	19001
GROUP - OWNERS MODIFIED	The owners of a group have been modified	5	19001
LICENSE - EXPIRES 30 DAYS	Secret servers license will expire in 30 days	1	19001
POWERSHELL SCRIPT - CREATE	A PowerShell script has been created	5	19001
POWERSHELL SCRIPT - DEACTIVATE	A PowerShell script has been deactivated	5	19001

POWERSHELL SCRIPT - EDIT	A PowerShell script has been edited	8	19001
POWERSHELL SCRIPT - REACTIVATE	A PowerShell script has been reactivated	6	19001
POWERSHELL SCRIPT - VIEW	A PowerShell script has been viewed	5	19001
ROLE - ASSIGN USER OR GROUP	A role has been assigned to a user or group	5	19001
ROLE - CREATE	A role has been created	5	19001
ROLE - UNASSIGN USER OR GROUP	A role has been unassigned to a user or group	5	19001
ROLE PERMISSION - ADDED TO ROLE	A permission has been added to a role	5	19001
ROLE PERMISSION - REMOVED FROM ROLE	A permission has been removed from a role	5	19001
SECRET - ACCESS APPROVED	Access to a Secret has been approved	2	19001
SECRET - ACCESS DENIED	Access to a Secret has been denied	6	19001
SECRET - CHECKIN	A Secret has been checked in	1	19001
SECRET - CHECKOUT	A Secret has been checked out	5	19001
SECRET - COPY	A Secret has been copied	1	19001
SECRET - CREATE	A Secret has been created	1	19001
SECRET - CUSTOM AUDIT	A custom audit has been created	1	19001
SECRET - CUSTOM REQUIREMENT ADDED	A custom password requirement has been added to a Secret	2	19001
SECRET - CUSTOM REQUIREMENT REMOVED	A custom password requirement has been removed from a Secret	6	19001
SECRET - DELETE	A Secret has been deleted	5	19001
SECRET - DEPENDENCY ADDED	A dependency has been added	8	19001
SECRET - DEPENDENCY FAILURE	A dependency is missing	5	19001
SECRET - DEPENDENCY REMOVED	A dependency has been removed	8	19001
SECRET - EDIT	A Secret has been edited	5	19001
SECRET - EDIT VIEW	A Secrets view option has been edited	8	19001
SECRET - EXPIRES 1 DAY	A Secret expires in 1 day	5	19001
SECRET - EXPIRES 15 DAYS	A Secret expires in 15 days	1	19001
SECRET - EXPIRES 3 DAYS	A Secret expires in 3 days	1	19001
SECRET - EXPIRES 7 DAYS	A Secret expires in 7 days	1	19001
SECRET - EXPIRES TODAY	A Secret expires today	1	19001

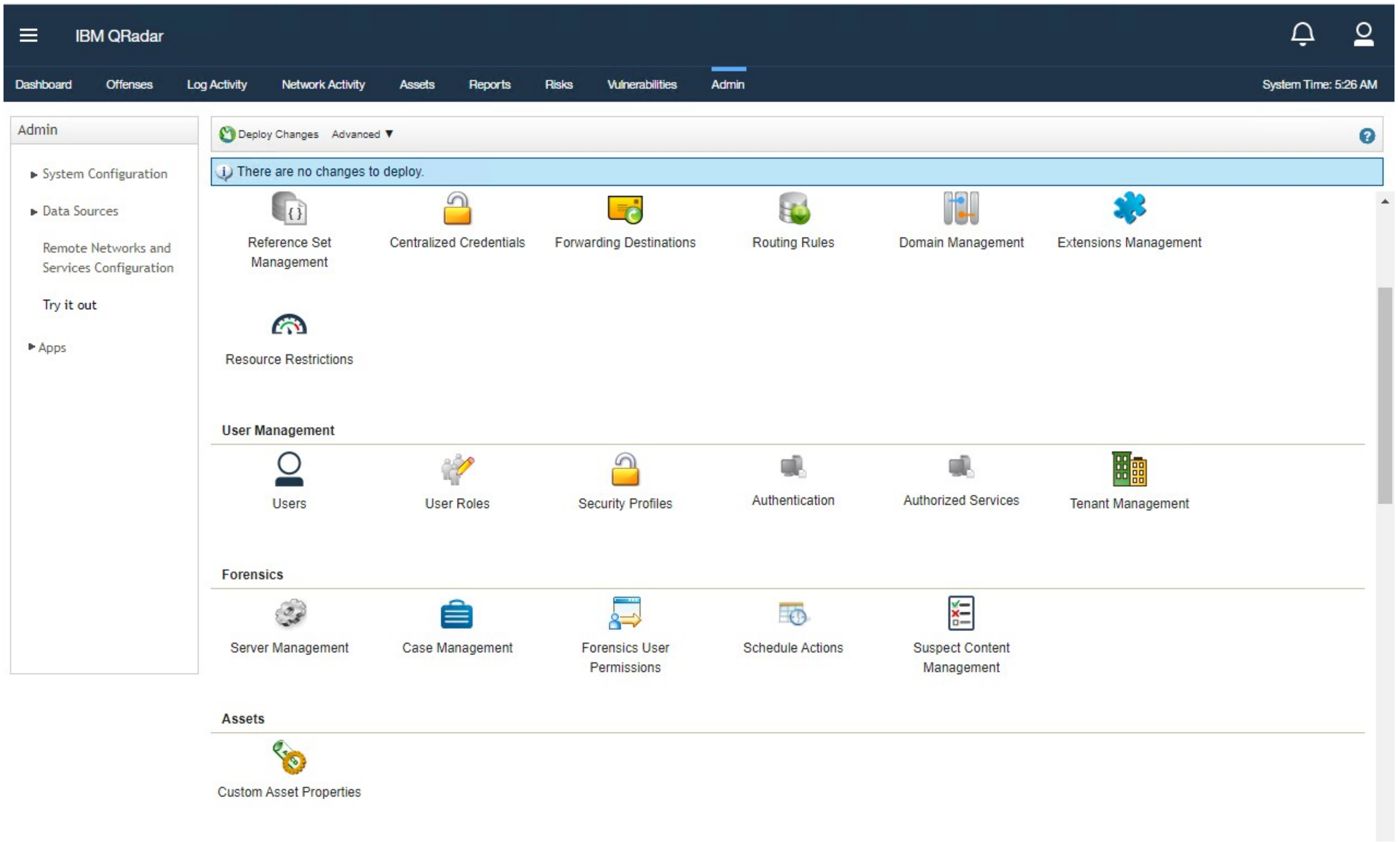
SECRET - HEARTBEAT FAILURE	Heartbeat has not been detected for over 10 seconds	5	19001
SECRET - HEARTBEATSUCCESS	Heartbeat has been detected	1	19001
SECRET - HOOK CREATE	A hook has been created	3	19001
SECRET - HOOK DELETE	A hook has been deleted	8	19001
SECRET - HOOK EDIT	A hook has been edited	6	19001
SECRET - HOOKFAILURE	A hook has failed to initialise a PowerShell script	8	19001
SECRET - HOOKSUCCESS	A hook has successfully initialised a PowerShell script	1	19001
SECRET - LAUNCH	A Secret has been launched	1	19001
SECRET - PASSWORD COPIED TO CLIPBOARD	A password has been copied to the clipboard	5	19001
SECRET - PASSWORD_DISPLAYED	A Secret password has been displayed	5	19001
SECRET - SECRET POLICY CHANGE	The Secret policy assigned to a Secret has been changed	8	19001
SECRET - SESSION RECORDING VIEW	A Secret recording is being viewed	5	19001
SECRET - UNDELETE	A Secret has been restored	1	19001
SECRET - VIEW	A Secret has been viewed	1	19001
SECRET POLICY - CREATE	A Secret policy has been created	1	19001
SECRET POLICY - EDIT	A Secret policy has been edited	6	19001
SECRET TEMPLATE - COPY	A Secret template has been copied	1	19001
SECRET TEMPLATE - CREATE	A Secret template has been created	1	19001
SECRET TEMPLATE - EDIT	A Secret template has been edited	1	19001
SECRET TEMPLATE - FIELD ENCRYPTED	A field in a template has been encrypted	1	19001
SECRET TEMPLATE - FIELD EXPOSED	A field in a template has been exposed	6	19001
SECRETS EXPORTD	Secrets have been exported	10	19001
SECRETS IMPORTED	Secrets have been imported	1	19001
SYSTEM LOG	Thycotic Secret server system logs	1	19001
UNLIMITED ADMIN - DISABLED	Unlimited admin has been disabled	10	19001
UNLIMITED ADMIN - ENABLED	Unlimited admin has been enabled	10	19001
USER - ADDED TO GROUP	A user account has been added to a group	8	19001

USER - CREATE	A user account has been created	5	19001
USER - DISABLE	A user account has been disabled	5	19001
USER - ENABLE	A user account has been enabled	5	19001
USER - LOCKOUT	A user account has been locked out see payload for information	10	19001
USER - LOGIN	A user has logged on	1	19001
USER - LOGIN FAILURE	A user has entered an incorrect password	8	19001
USER - LOGOUT	A user has logged out	1	19001
USER - PASSWORD CHANGE	A users password has been changed	8	19001
USER - REMOVED FROM GROUP	A user account has been removed from a group	5	19001
USERAUDIT - EXPIRENOW	All Secrets a user has accessed have expired	5	19001

How to export the event mappings along with the Custom DSM

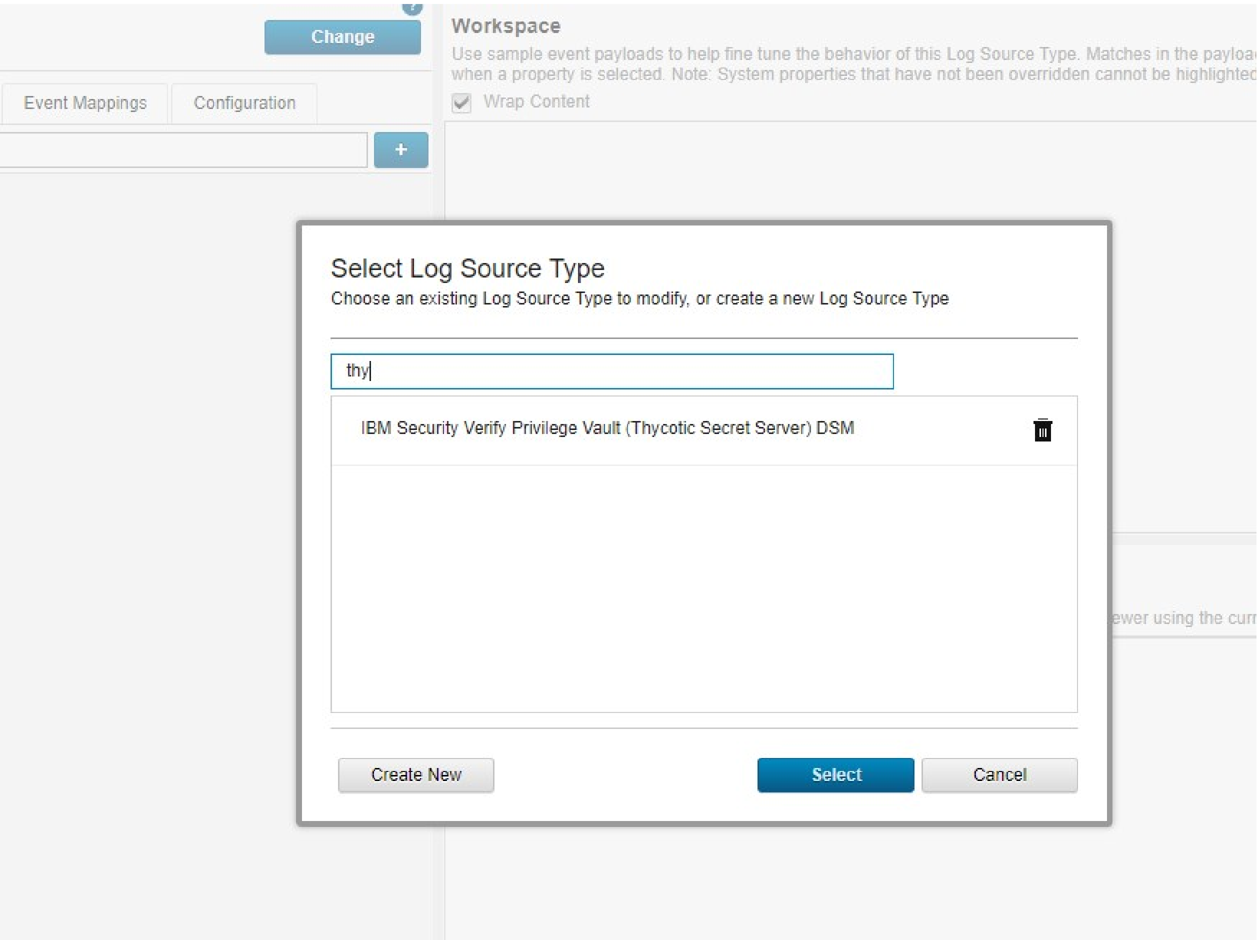
After Creating your QIDmap entries, you can map them to your events using the DSM editor and export them via the export option.

1. Navigate and login to **Qradar**.
2. Click on **Admin**.



There are no changes to deploy.

3. Click on the **DSM editor** option.
4. Select the created log source, search for **"Thy"**.



Change
Event Mappings
Configuration
+
Workspace
Use sample event payloads to help fine tune the behavior of this Log Source Type. Matches in the payloa
when a property is selected. Note: System properties that have not been overridden cannot be highlighted
Wrap Content
ewer using the curr

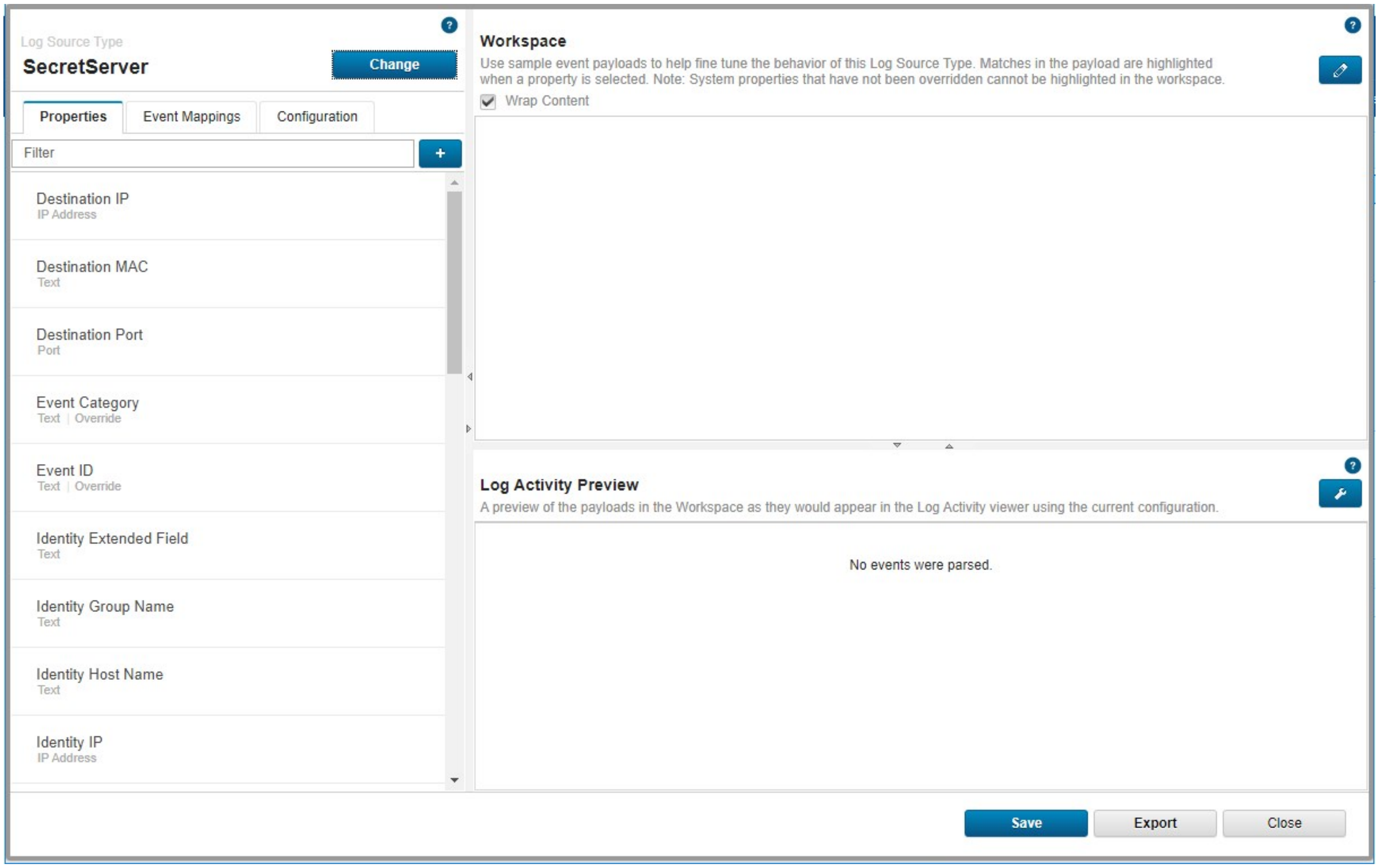
Select Log Source Type

Choose an existing Log Source Type to modify, or create a new Log Source Type

thy

IBM Security Verify Privilege Vault (Thycotic Secret Server) DSM

5. Click on **Select**.
6. Click on **Export.**


Log Source Type

SecretServer

Workspace

Use sample event payloads to help fine tune the behavior of this Log Source Type. Matches in the payload are highlighted when a property is selected. Note: System properties that have not been overridden cannot be highlighted in the workspace.

☑ Wrap Content

Destination IP	IP Address
Destination MAC	Text
Destination Port	Port
Event Category	Text | Override
Event ID	Text | Override
Identity Extended Field	Text
Identity Group Name	Text
Identity Host Name	Text
Identity IP	IP Address

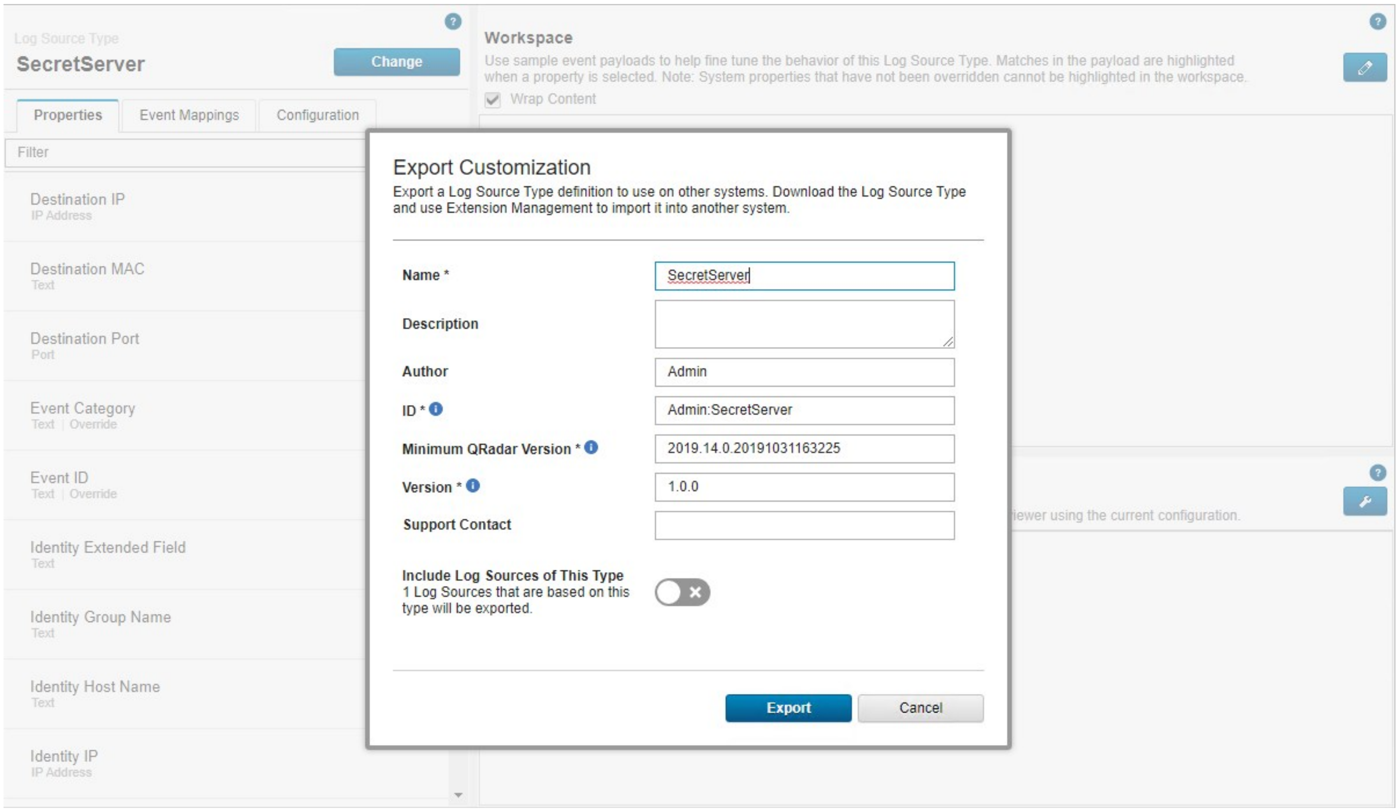
Log Activity Preview

A preview of the payloads in the Workspace as they would appear in the Log Activity viewer using the current configuration.

No events were parsed.

7. Enter in the required details.

8. Click on **Export**.



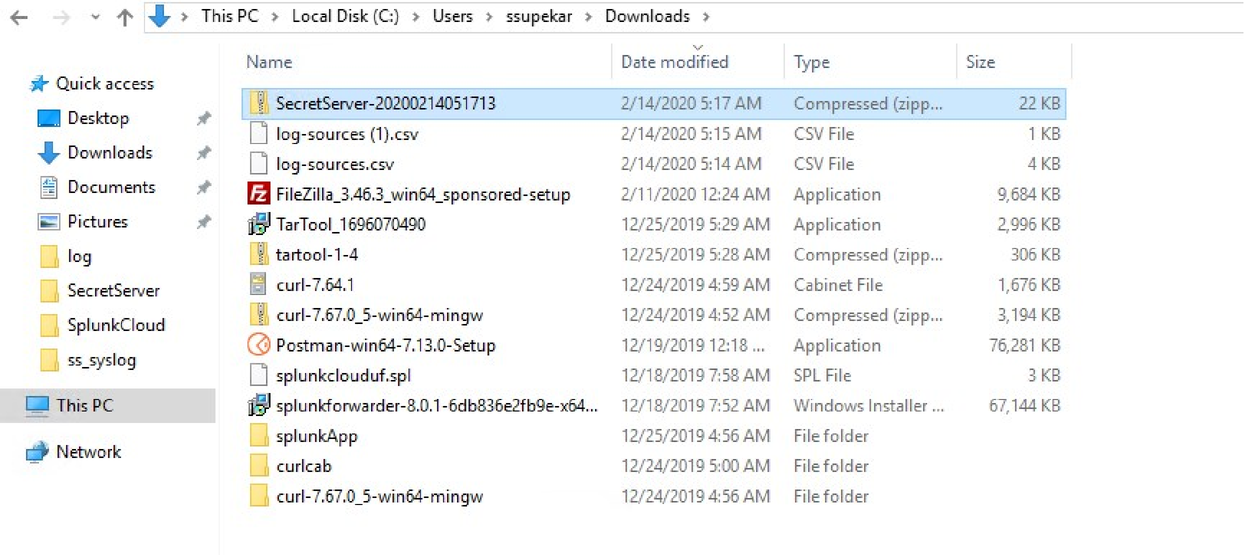
Export Customization

Export a Log Source Type definition to use on other systems. Download the Log Source Type and use Extension Management to import it into another system.

Name *	SecretServer
Description	
Author	Admin
ID *	Admin:SecretServer
Minimum QRadar Version *	2019.14.0.20191031163225
Version *	1.0.0
Support Contact	

Include Log Sources of This Type
1 Log Sources that are based on this type will be exported.

9. The zip will be downloaded.

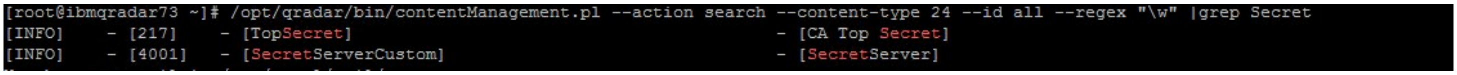


Name	Date modified	Type	Size
SecretServer-20200214051713	2/14/2020 5:17 AM	Compressed (zipp...	22 KB
log-sources (1).csv	2/14/2020 5:15 AM	CSV File	1 KB
log-sources.csv	2/14/2020 5:14 AM	CSV File	4 KB
FileZilla_3.46.3_win64_sponsored-setup	2/11/2020 12:24 AM	Application	9,684 KB
TarTool_1696070490	12/25/2019 5:29 AM	Application	2,996 KB
tartool-1-4	12/25/2019 5:28 AM	Compressed (zipp...	306 KB
curl-7.64.1	12/24/2019 4:59 AM	Cabinet File	1,676 KB
curl-7.67.0_5-win64-mingw	12/24/2019 4:52 AM	Compressed (zipp...	3,194 KB
Postman-win64-7.13.0-Setup	12/19/2019 12:18 ...	Application	76,281 KB
splunkclouduf.spl	12/18/2019 7:58 AM	SPL File	3 KB
splunkforwarder-8.0.1-6db836e2fb9e-x64...	12/18/2019 7:52 AM	Windows Installer ...	67,144 KB
splunkApp	12/25/2019 4:56 AM	File folder	
curlcab	12/24/2019 5:00 AM	File folder	
curl-7.67.0_5-win64-mingw	12/24/2019 4:56 AM	File folder	

To search for your DSM using the ContentManagement Tool

Enter in the following command:

```
[root\@qradar \~]\# /opt/qradar/bin/contentManagement.pl --action search --content-type 24 --id all --regex "\\w" \|grep Secret
```



```
[root@ibmqradar73 ~]# /opt/qradar/bin/contentManagement.pl --action search --content-type 24 --id all --regex "\w" |grep Secret
[INFO]    - [217]    - [TopSecret]                                         - [CA Top Secret]
[INFO]    - [4001]   - [SecretServerCustom]                                - [SecretServer]
```

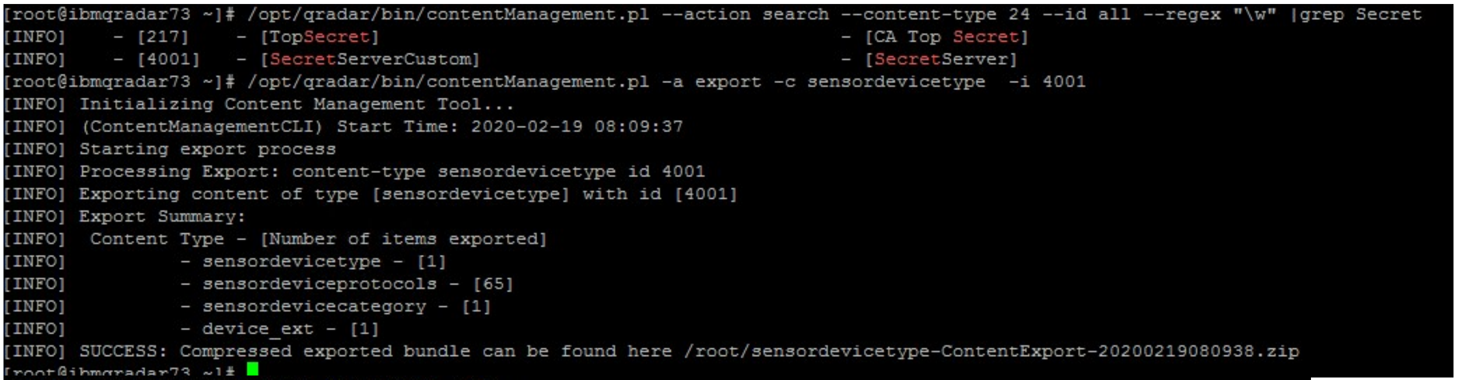
To export the custom mappings

Enter in the following command:

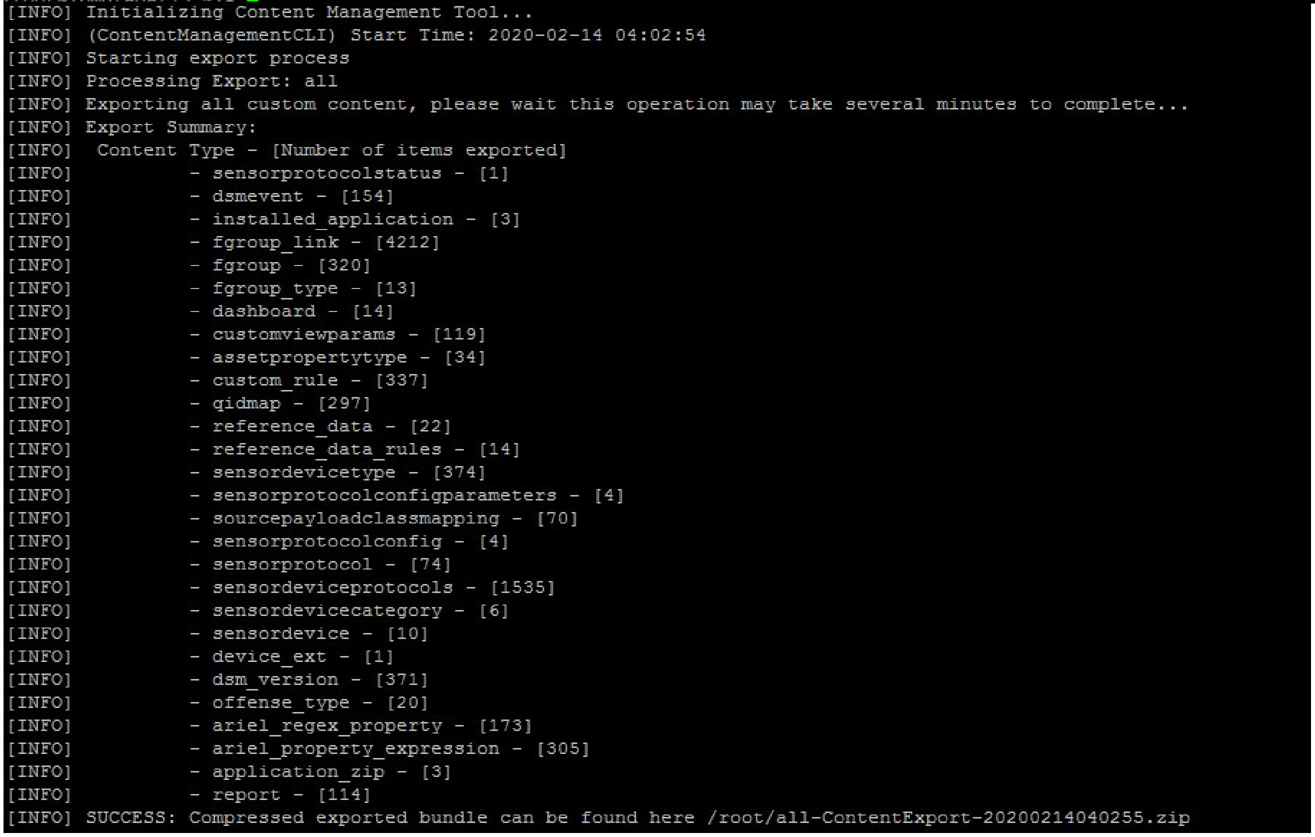
[root\@qradar \~]\# /opt/qradar/bin/contentManagement.pl -a export -c all

/opt/qradar/bin/contentManagement.pl -a export -c sensordevicetype -i 4001

Result



```
[root@ibmqradar73 ~]# /opt/qradar/bin/contentManagement.pl --action search --content-type 24 --id all --regex "\w" |grep Secret
[INFO]    - [217]    - [TopSecret]                                         - [CA Top Secret]
[INFO]    - [4001]   - [SecretServerCustom]                                - [SecretServer]
[root@ibmqradar73 ~]# /opt/qradar/bin/contentManagement.pl -a export -c sensordevicetype  -i 4001
[INFO] Initializing Content Management Tool...
[INFO] (ContentManagementCLI) Start Time: 2020-02-19 08:09:37
[INFO] Starting export process
[INFO] Processing Export: content-type sensordevicetype id 4001
[INFO] Exporting content of type [sensordevicetype] with id [4001]
[INFO] Export Summary:
[INFO]  Content Type - [Number of items exported]
[INFO]          - sensordevicetype - [1]
[INFO]          - sensordeviceprotocols - [65]
[INFO]          - sensordevicecategory - [1]
[INFO]          - device_ext - [1]
[INFO] SUCCESS: Compressed exported bundle can be found here /root/sensordevicetype-ContentExport-20200219080938.zip
[root@ibmqradar73 ~]#
```



```
[INFO] Initializing Content Management Tool...
[INFO] (ContentManagementCLI) Start Time: 2020-02-14 04:02:54
[INFO] Starting export process
[INFO] Processing Export: all
[INFO] Exporting all custom content, please wait this operation may take several minutes to complete...
[INFO] Export Summary:
[INFO]  Content Type - [Number of items exported]
[INFO]          - sensorprotocolstatus - [1]
[INFO]          - dsmevent - [154]
[INFO]          - installed_application - [3]
[INFO]          - fgroup_link - [4212]
[INFO]          - fgroup - [320]
[INFO]          - fgroup_type - [13]
[INFO]          - dashboard - [14]
[INFO]          - customviewparams - [119]
[INFO]          - assetpropertytype - [34]
[INFO]          - custom_rule - [337]
[INFO]          - qidmap - [297]
[INFO]          - reference_data - [22]
[INFO]          - reference_data_rules - [14]
[INFO]          - sensordevicetype - [374]
[INFO]          - sensorprotocolconfigparameters - [4]
[INFO]          - sourcepayloadclassmapping - [70]
[INFO]          - sensorprotocolconfig - [4]
[INFO]          - sensorprotocol - [74]
[INFO]          - sensordeviceprotocols - [1535]
[INFO]          - sensordevicecategory - [6]
[INFO]          - sensordevice - [10]
[INFO]          - device_ext - [1]
[INFO]          - dsm_version - [371]
[INFO]          - offense_type - [20]
[INFO]          - ariel_regex_property - [173]
[INFO]          - ariel_property_expression - [305]
[INFO]          - application_zip - [3]
[INFO]          - report - [114]
[INFO] SUCCESS: Compressed exported bundle can be found here /root/all-ContentExport-20200214040255.zip
```

1. Rename the zip file to MyExport.zip.

2. On the new Qradar install, copy the .zip file and reimport it.

   Enter in the following command:

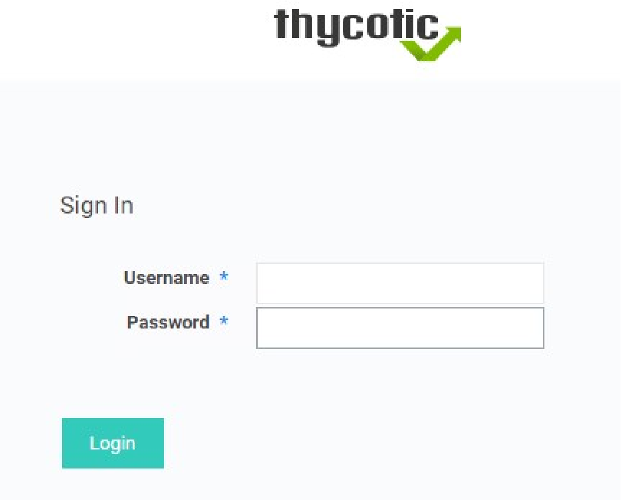
   ```
   [root\@qradar \~]\# /opt/qradar/bin/contentManagement.pl --action import --file MyExport.zip
   ```

Secret Server Dashboard Extension

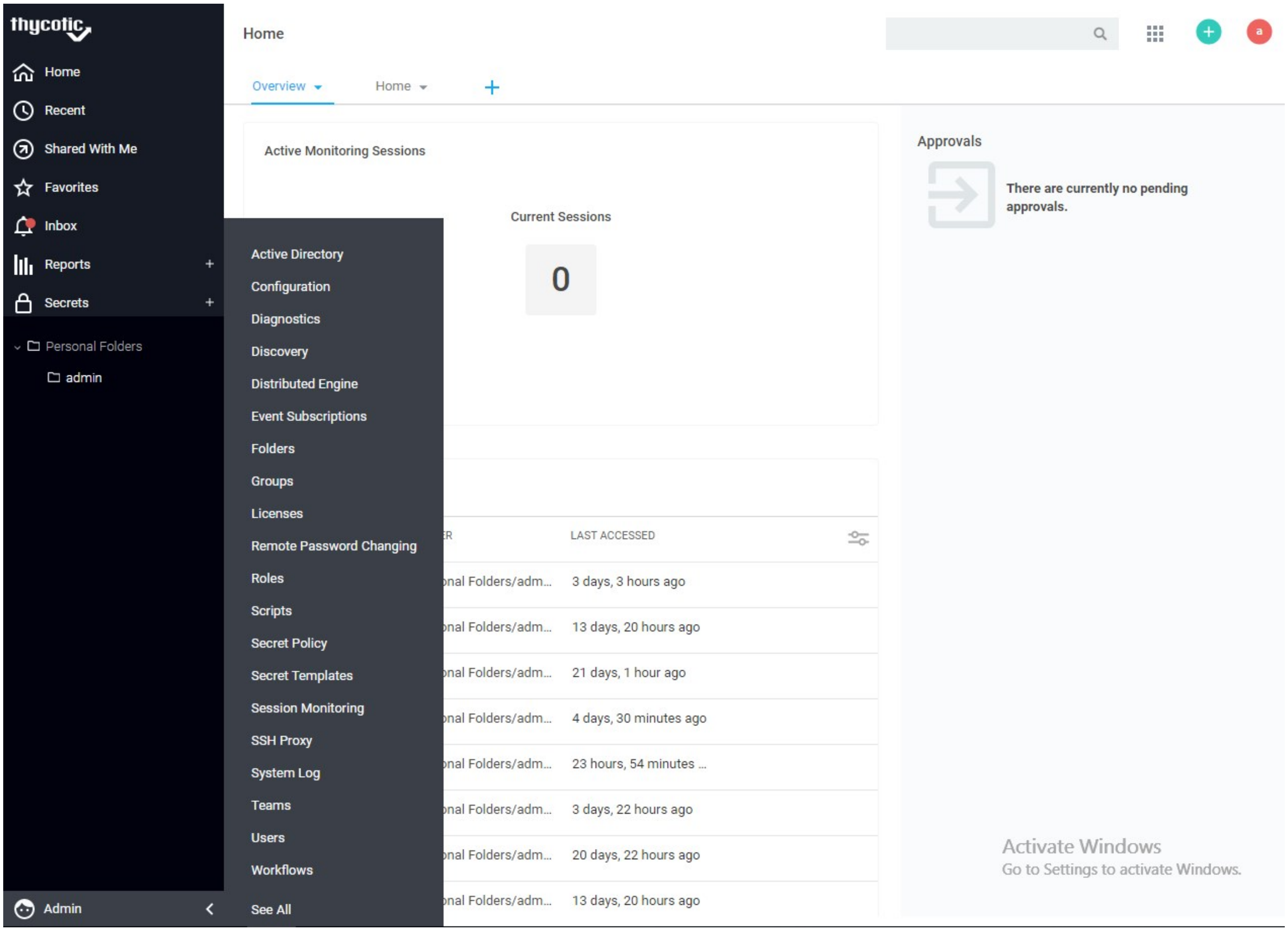
- Configuring Secret Server settings
- Accessing Secret Server Events in the Secret Server Application within QRadar
- Pulse Dashboard Setup

Configuring Secret Server settings

1. Sign into **Secret Server**.

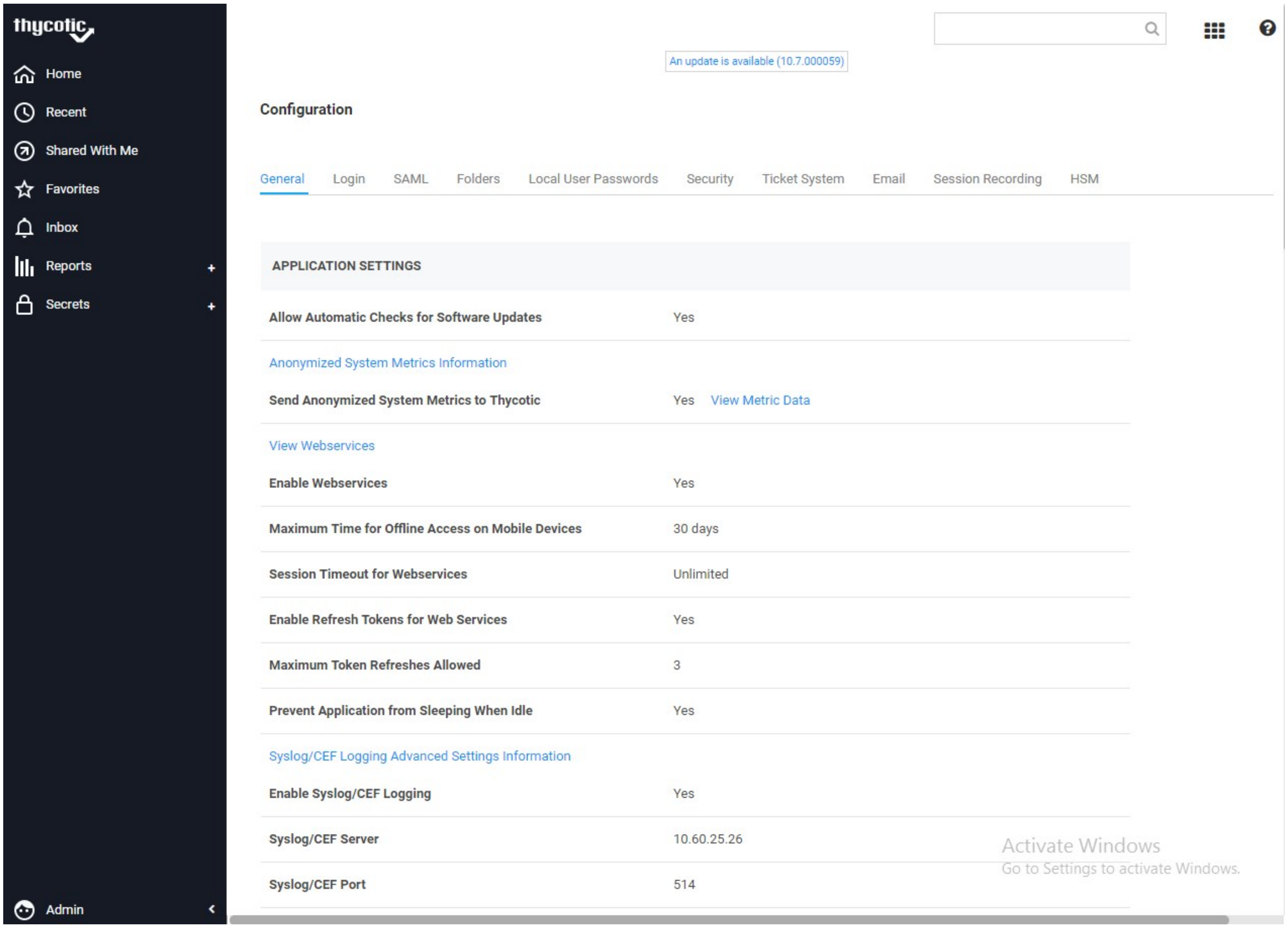


2. The **Home** page appears.



ER	LAST ACCESSED
onal Folders/adm...	3 days, 3 hours ago
onal Folders/adm...	13 days, 20 hours ago
onal Folders/adm...	21 days, 1 hour ago
onal Folders/adm...	4 days, 30 minutes ago
onal Folders/adm...	23 hours, 54 minutes ...
onal Folders/adm...	3 days, 22 hours ago
onal Folders/adm...	20 days, 22 hours ago
onal Folders/adm...	13 days, 20 hours ago

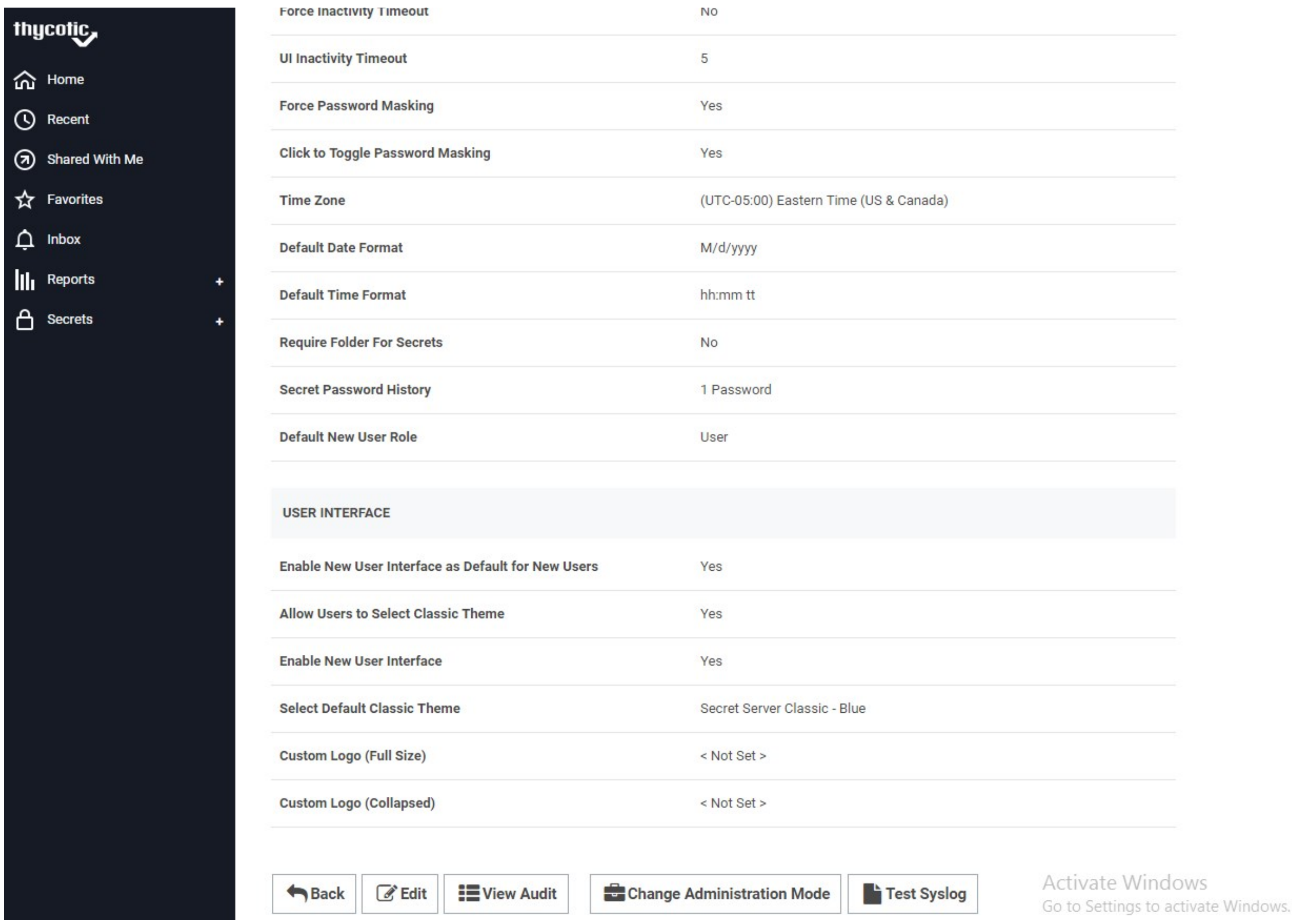
3. Click **Admin** | **Configuration**.



Configuration

APPLICATION SETTINGS	
Allow Automatic Checks for Software Updates	Yes
Anonymized System Metrics Information	
Send Anonymized System Metrics to Thycotic	Yes View Metric Data
View Webservices	
Enable Webservices	Yes
Maximum Time for Offline Access on Mobile Devices	30 days
Session Timeout for Webservices	Unlimited
Enable Refresh Tokens for Web Services	Yes
Maximum Token Refreshes Allowed	3
Prevent Application from Sleeping When Idle	Yes
Syslog/CEF Logging Advanced Settings Information	
Enable Syslog/CEF Logging	Yes
Syslog/CEF Server	10.60.25.26
Syslog/CEF Port	514

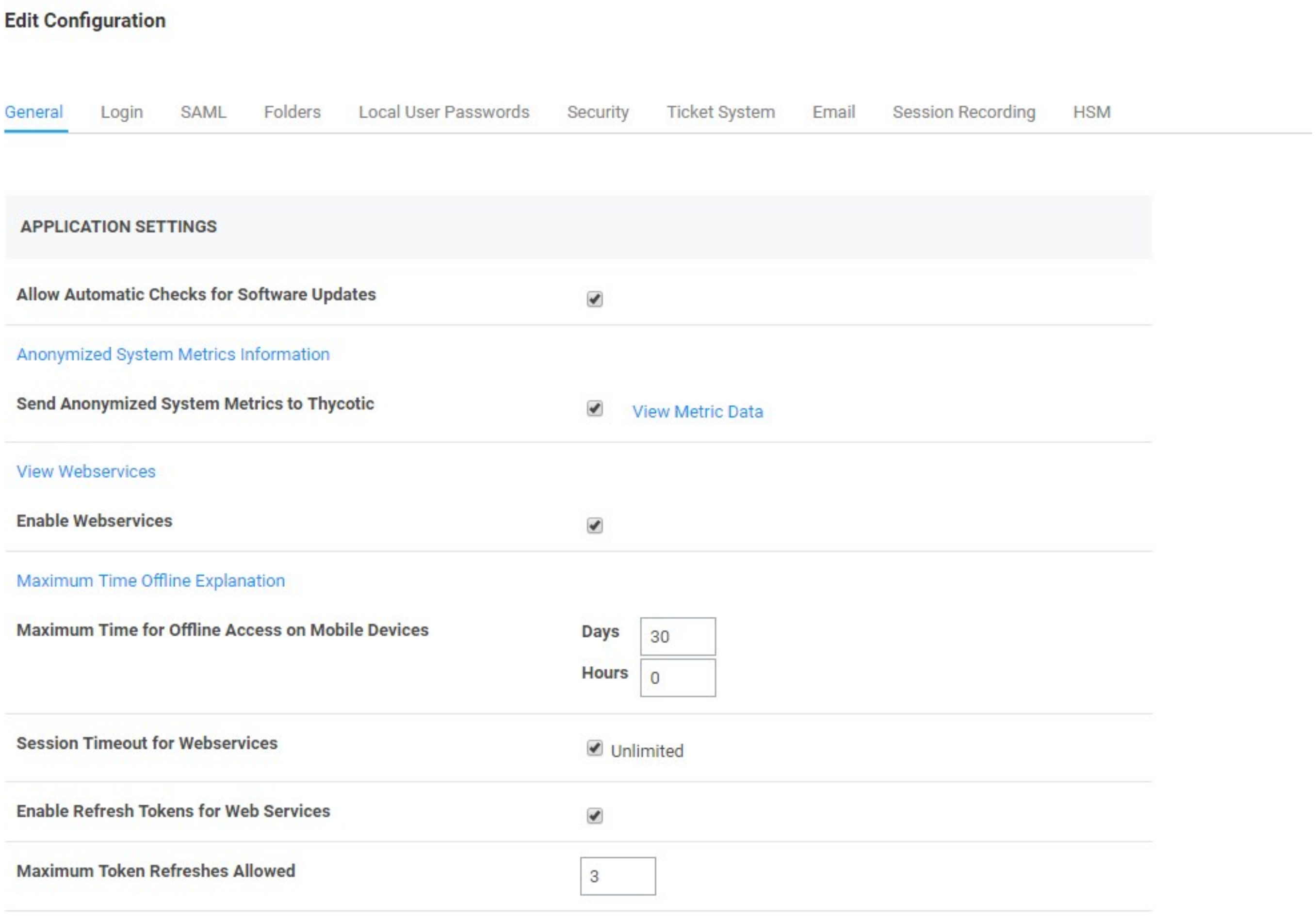
4. At the bottom of the page, click **Edit**.



Force Inactivity Timeout	No
UI Inactivity Timeout	5
Force Password Masking	Yes
Click to Toggle Password Masking	Yes
Time Zone	(UTC-05:00) Eastern Time (US & Canada)
Default Date Format	M/d/yyyy
Default Time Format	hh:mm tt
Require Folder For Secrets	No
Secret Password History	1 Password
Default New User Role	User
USER INTERFACE	
Enable New User Interface as Default for New Users	Yes
Allow Users to Select Classic Theme	Yes
Enable New User Interface	Yes
Select Default Classic Theme	Secret Server Classic - Blue
Custom Logo (Full Size)	< Not Set >
Custom Logo (Collapsed)	< Not Set >

Back
Edit
View Audit
Change Administration Mode
Test Syslog
Activate Windows
Go to Settings to activate Windows.

5. The **Edit Configuration** page appears.



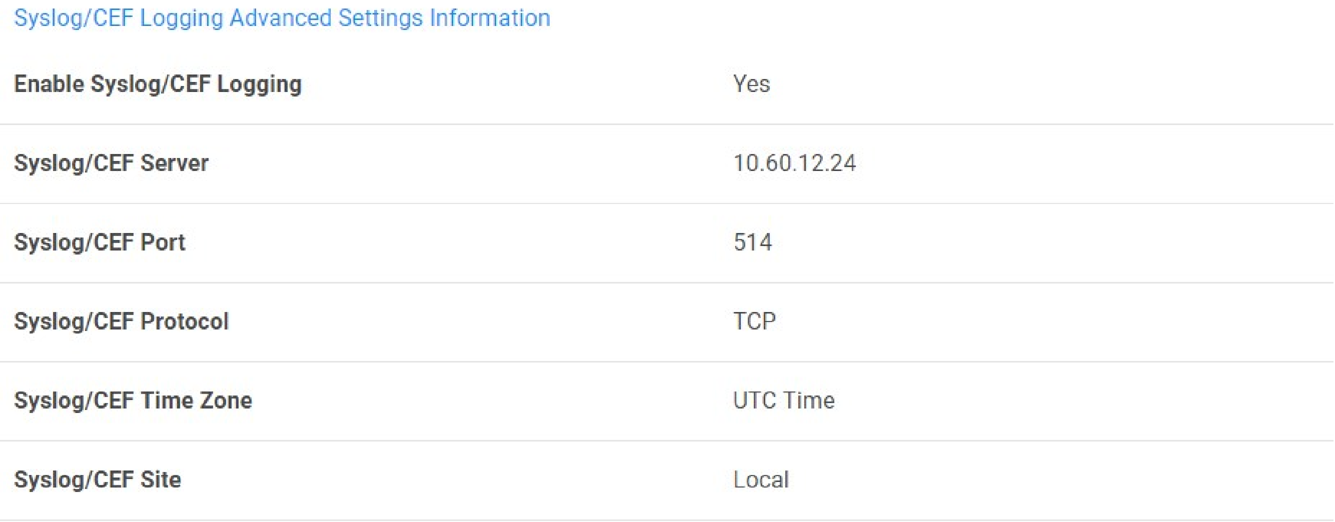
Edit Configuration

APPLICATION SETTINGS

Allow Automatic Checks for Software Updates	☑
Anonymized System Metrics Information	
Send Anonymized System Metrics to Thycotic	☑ View Metric Data
View Webservices	
Enable Webservices	☑
Maximum Time Offline Explanation	
Maximum Time for Offline Access on Mobile Devices	Days 30 Hours 0
Session Timeout for Webservices	☑ Unlimited
Enable Refresh Tokens for Web Services	☑
Maximum Token Refreshes Allowed	3

6. Select the **Enable Webservices** settings check box.
7. Under the **Syslog/CEF Logging Advanced Settings Information** area, select the **Enable Syslog/CEF Logging** check box and enter the **syslog server**.

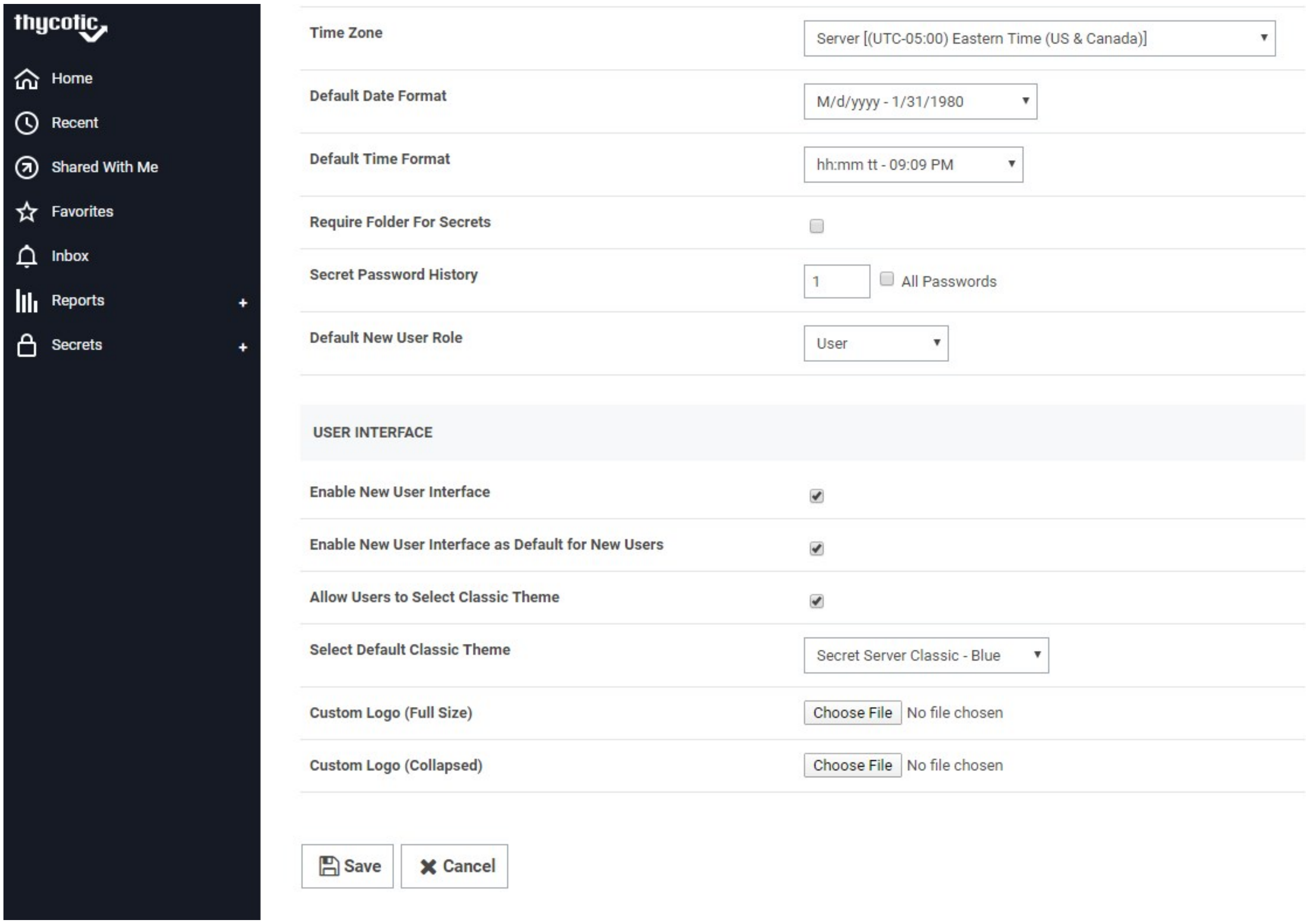
   Note: The syslog server should be the ip of machine/server where universal forwarder is configured), UDP port etc.



Syslog/CEF Logging Advanced Settings Information

Enable Syslog/CEF Logging	Yes
Syslog/CEF Server	10.60.12.24
Syslog/CEF Port	514
Syslog/CEF Protocol	TCP
Syslog/CEF Time Zone	UTC Time
Syslog/CEF Site	Local

8. At the end of the page, click **Save**.

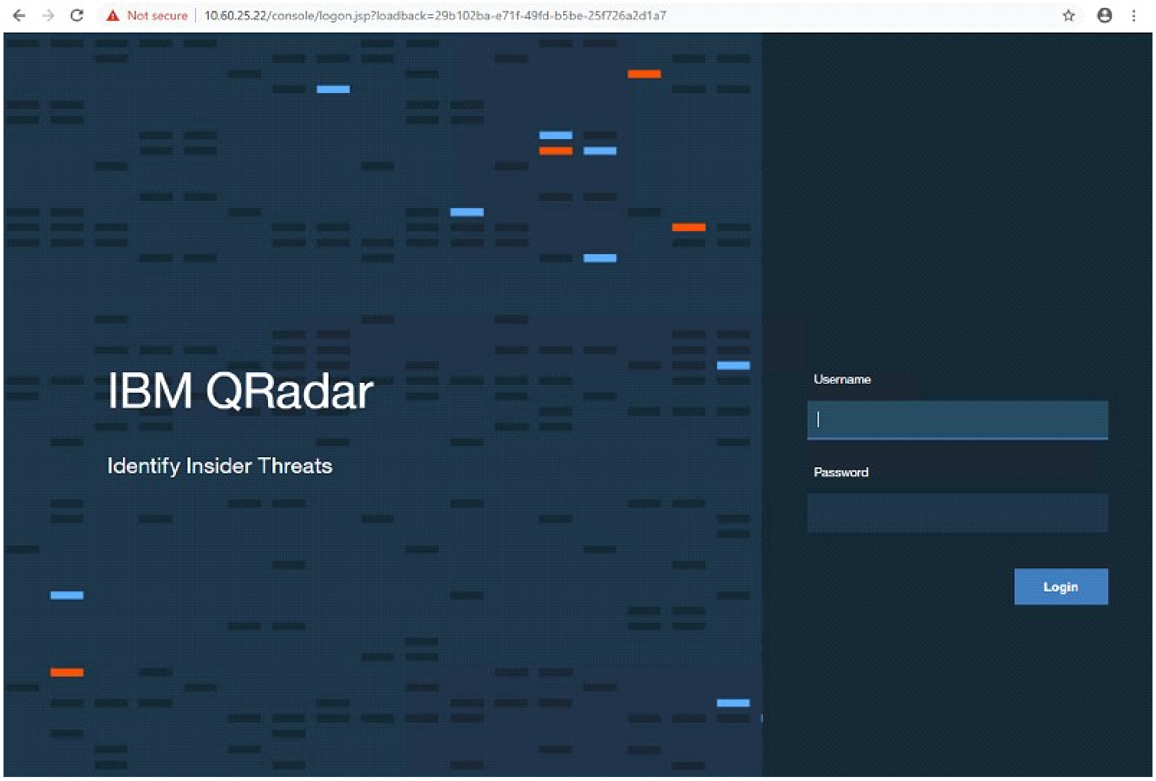


Time Zone	Server [(UTC-05:00) Eastern Time (US & Canada)] ▾
Default Date Format	M/d/yyyy - 1/31/1980 ▾
Default Time Format	hh:mm tt - 09:09 PM ▾
Require Folder For Secrets	□
Secret Password History	1 □ All Passwords
Default New User Role	User ▾
USER INTERFACE	
Enable New User Interface	☑
Enable New User Interface as Default for New Users	☑
Allow Users to Select Classic Theme	☑
Select Default Classic Theme	Secret Server Classic - Blue ▾
Custom Logo (Full Size)	Choose File No file chosen
Custom Logo (Collapsed)	Choose File No file chosen

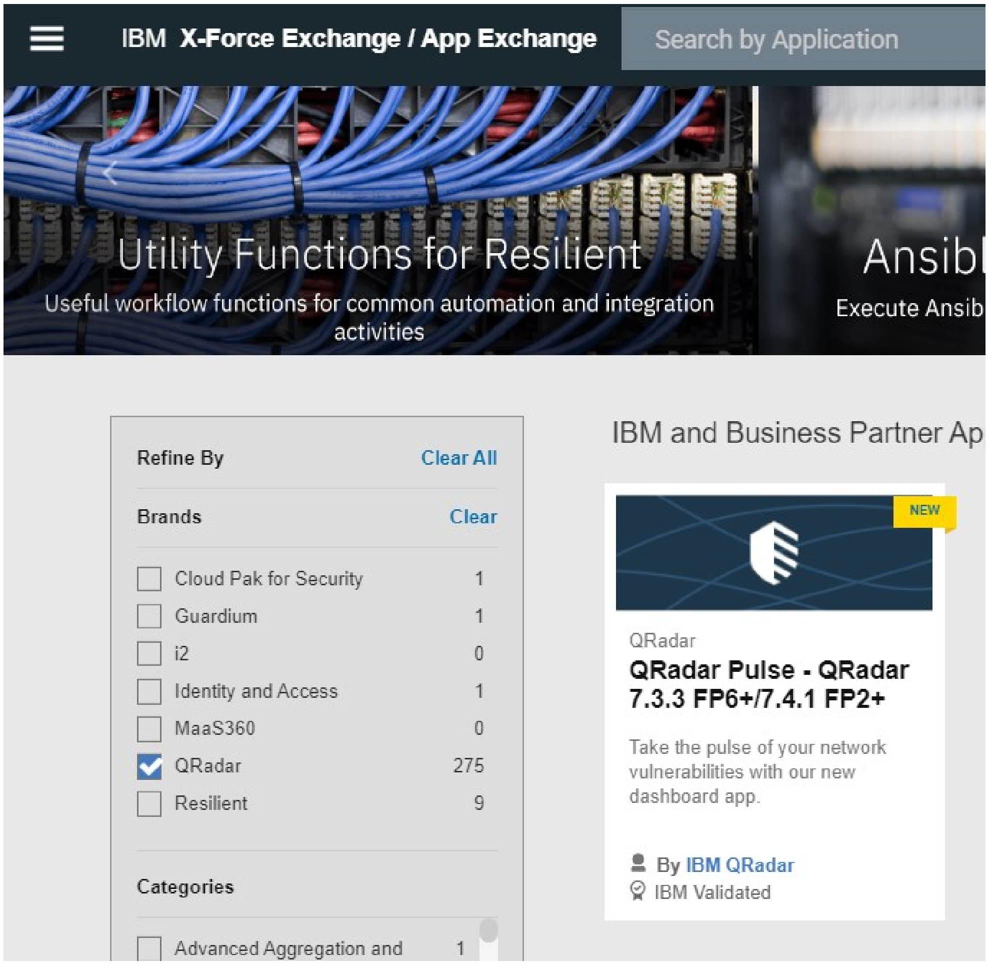
At this point all required configuration to get SysLog information from Secret Server into QRadar is complete.

Accessing Secret Server Events in the Secret Server Application within QRadar

1. Login to QRadar as the admin user: https://<ipaddress>


Not secure
10.60.25.22/console/logon.jsp?loadback=29b102ba-e71f-49fd-b5be-25f726a2d1a7
IBM QRadar
Identify Insider Threats
Username
Password
Login

2. Download **Thycotic Secret Server Dashboard** extension from https://exchange.xforce.ibmcloud.com/.


IBM X-Force Exchange / App Exchange
Search by Application
Utility Functions for Resilient
Useful workflow functions for common automation and integration activities
Ansibl
Execute Ansib
Refine By
Clear All
Brands
Clear
Categories
IBM and Business Partner Ap
NEW

Brands	
□ Cloud Pak for Security	1
□ Guardium	1
□ i2	0
□ Identity and Access	1
□ MaaS360	0
☑ QRadar	275
□ Resilient	9
□ Advanced Aggregation and	1

QRadar

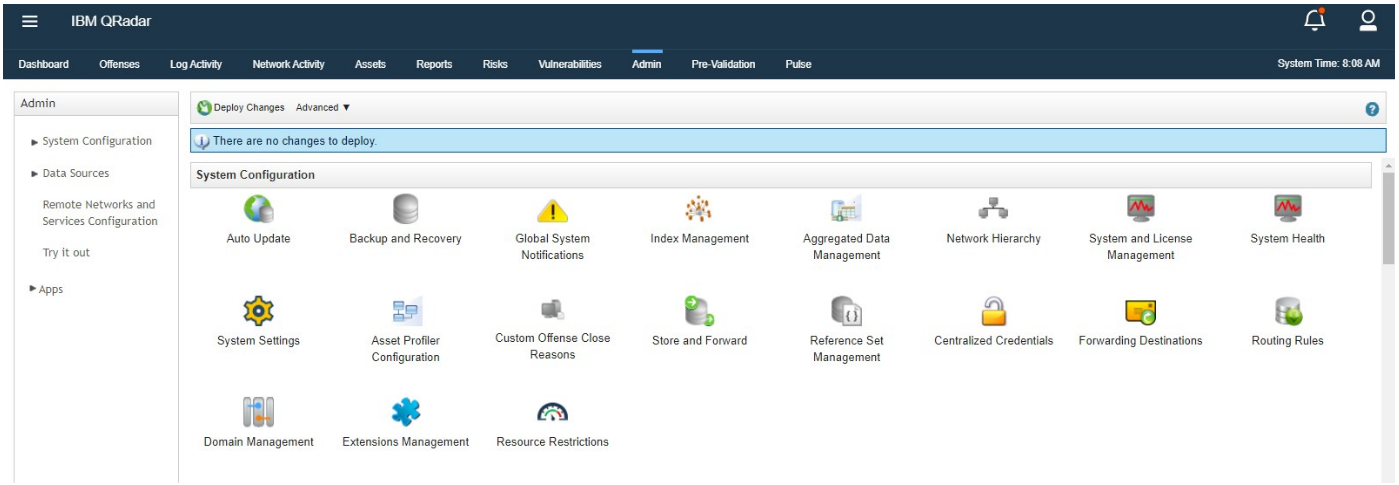
QRadar Pulse - QRadar 7.3.3 FP6+/7.4.1 FP2+

Take the pulse of your network vulnerabilities with our new dashboard app.

By IBM QRadar

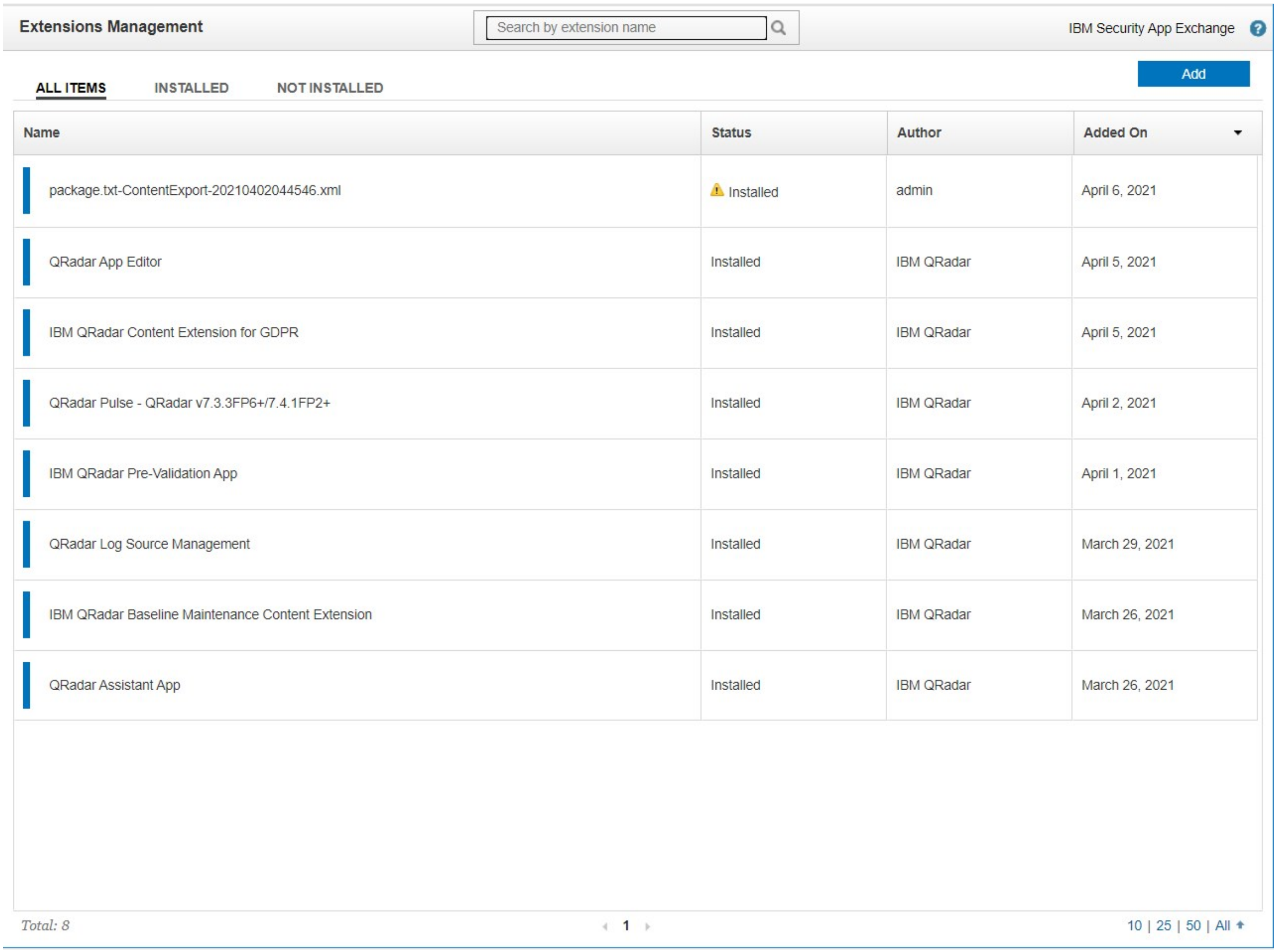
IBM Validated

3. Login to QRadar with Admin role and navigate to the **Extension management** by clicking on **Admin | Extension Management**.



There are no changes to deploy.

4. **Extension Management** will be displayed and click on **Add**.



Extensions Management

Search by extension name

IBM Security App Exchange

ALL ITEMS INSTALLED NOT INSTALLED

Add

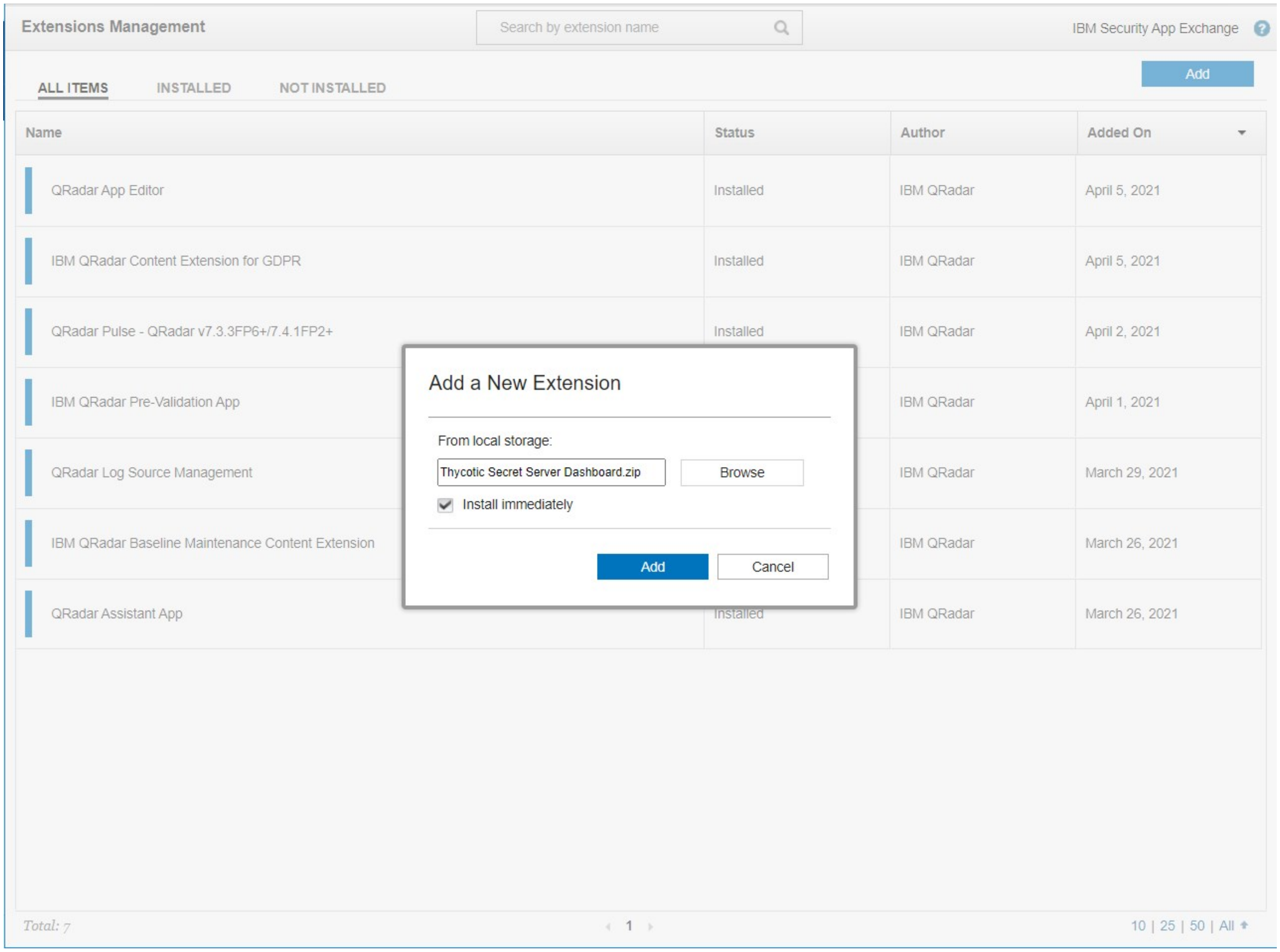
Name	Status	Author	Added On
package.txt-ContentExport-20210402044546.xml	⚠ Installed	admin	April 6, 2021
QRadar App Editor	Installed	IBM QRadar	April 5, 2021
IBM QRadar Content Extension for GDPR	Installed	IBM QRadar	April 5, 2021
QRadar Pulse - QRadar v7.3.3FP6+/7.4.1FP2+	Installed	IBM QRadar	April 2, 2021
IBM QRadar Pre-Validation App	Installed	IBM QRadar	April 1, 2021
QRadar Log Source Management	Installed	IBM QRadar	March 29, 2021
IBM QRadar Baseline Maintenance Content Extension	Installed	IBM QRadar	March 26, 2021
QRadar Assistant App	Installed	IBM QRadar	March 26, 2021

Total: 8

1

10 | 25 | 50 | All

5. Browse to **Thycotic Secret Server Dashboard** extension downloaded from IBM Exchange, click on the **Add button**.

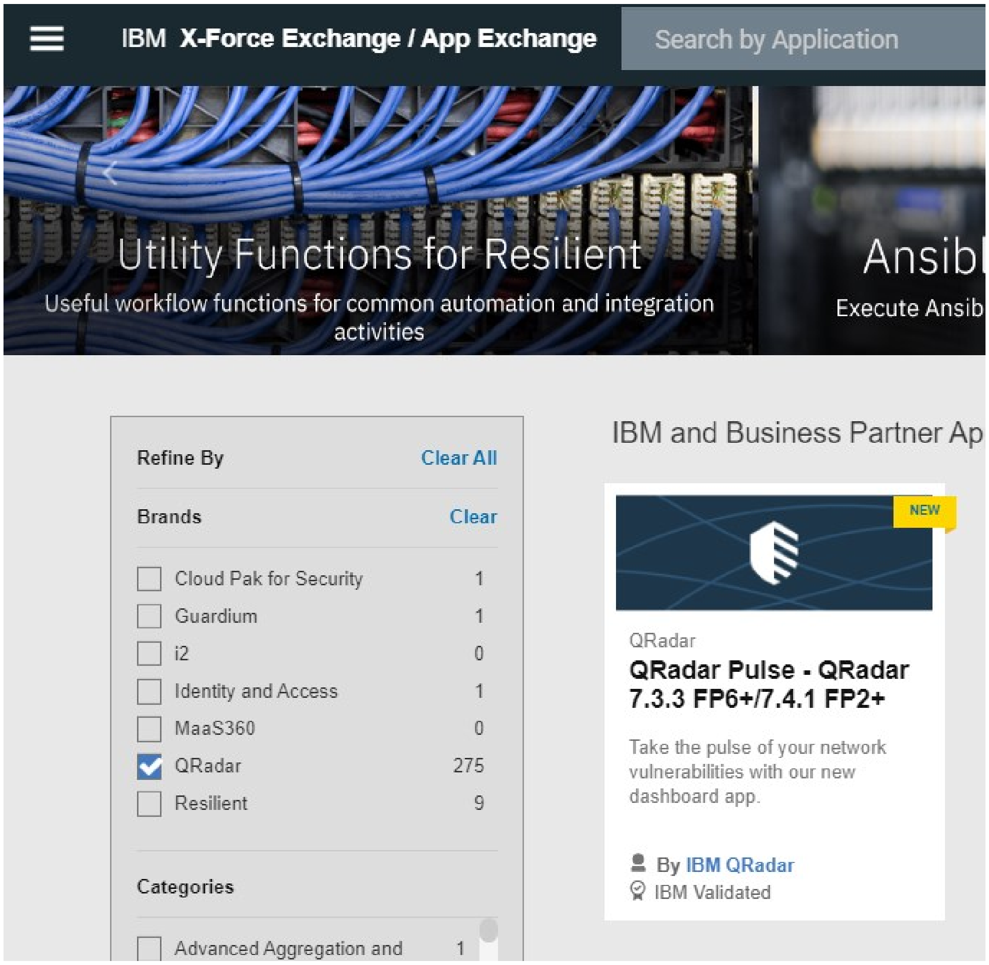


Name	Status	Author	Added On
QRadar App Editor	Installed	IBM QRadar	April 5, 2021
IBM QRadar Content Extension for GDPR	Installed	IBM QRadar	April 5, 2021
QRadar Pulse - QRadar v7.3.3FP6+/7.4.1FP2+	Installed	IBM QRadar	April 2, 2021
IBM QRadar Pre-Validation App		IBM QRadar	April 1, 2021
QRadar Log Source Management		IBM QRadar	March 29, 2021
IBM QRadar Baseline Maintenance Content Extension		IBM QRadar	March 26, 2021
QRadar Assistant App	Installed	IBM QRadar	March 26, 2021

From local storage: Thycotic Secret Server Dashboard.zip

☑ Install immediately

6. Download Pulse App from https://exchange.xforce.ibmcloud.com/ and install the Pulse extension by navigating to extension Management.



Useful workflow functions for common automation and integration activities

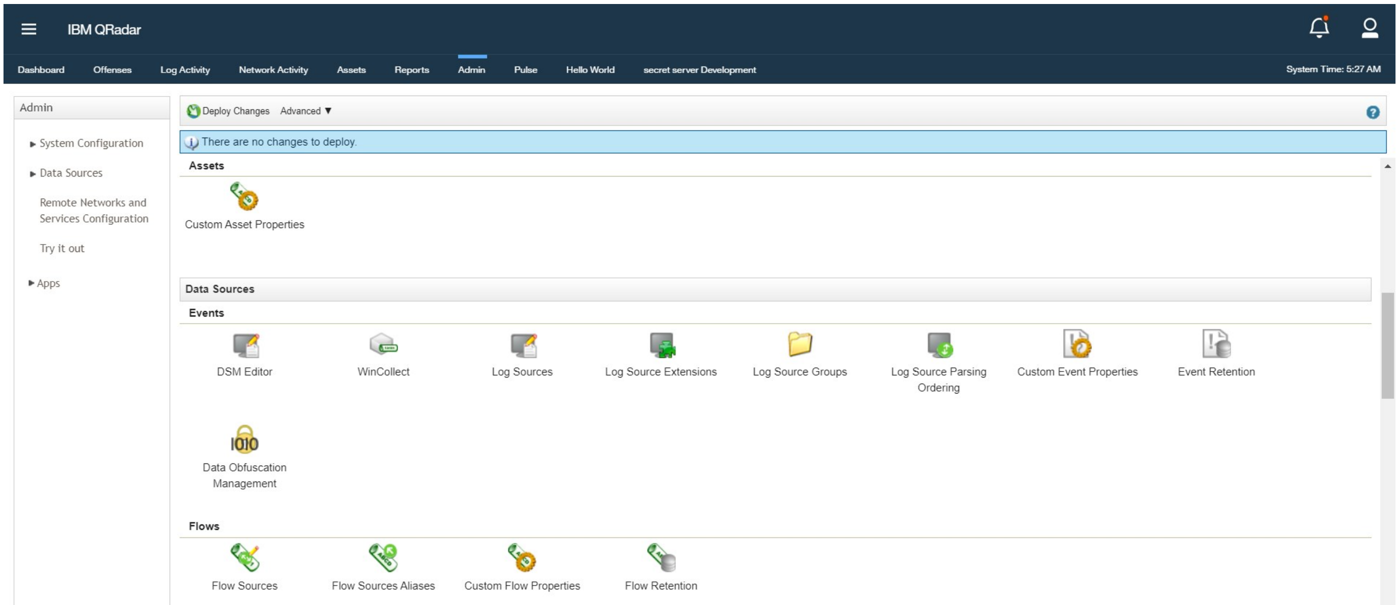
□ Cloud Pak for Security	1
□ Guardium	1
□ i2	0
□ Identity and Access	1
□ MaaS360	0
☑ QRadar	275
□ Resilient	9
□ Advanced Aggregation and	1

QRadar Pulse - QRadar 7.3.3 FP6+/7.4.1 FP2+

Take the pulse of your network vulnerabilities with our new dashboard app.

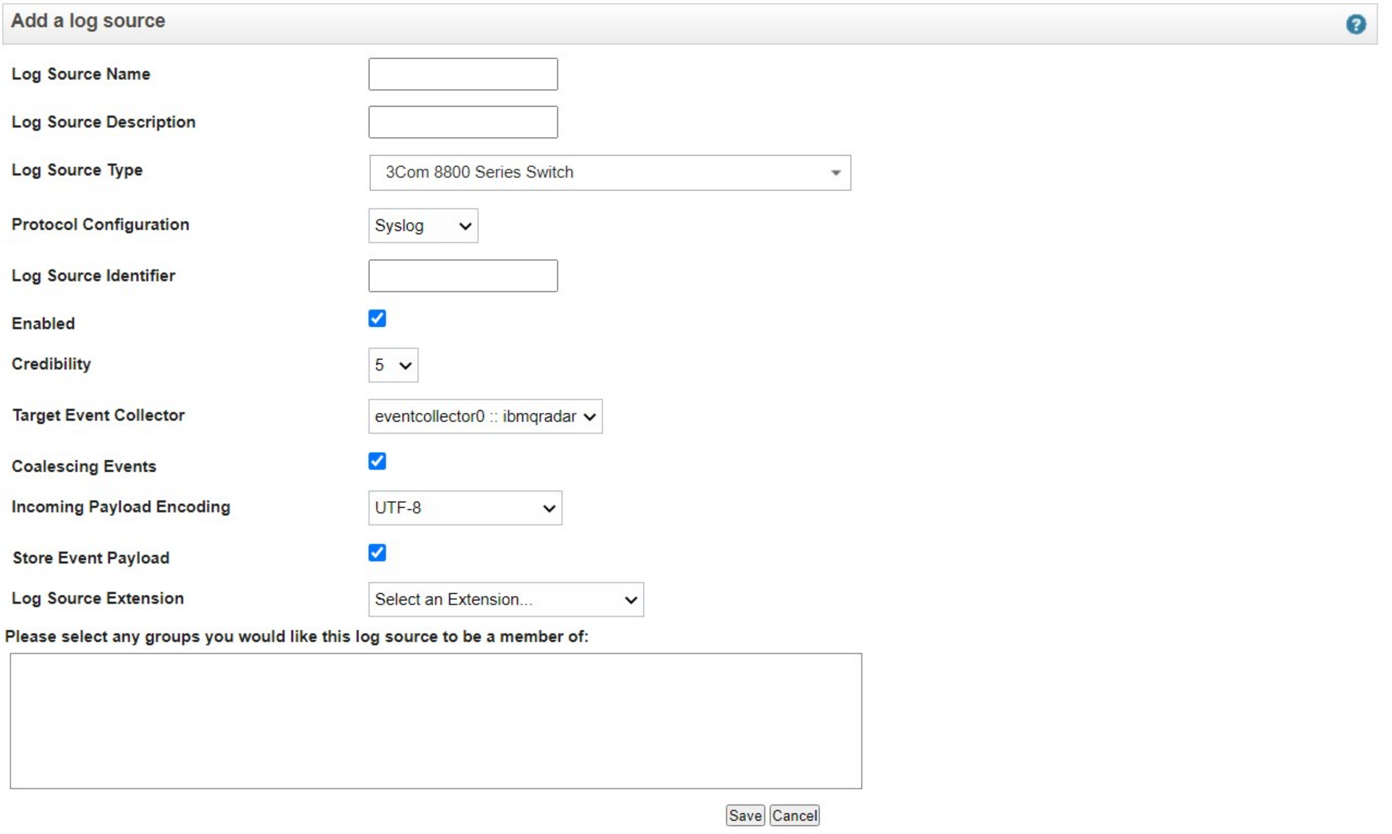
Create log source

1. Click **on Admin I Log source**.



There are no changes to deploy.

2. Click on the **Add** button.
3. Add log source popup will be displayed.
    1. Enter the Name down the Log Source Name which will be required in Step 9 in Pulse Dashboard Setup.
    2. Description.
    3. Select the log source type created in above steps.
    4. Enter the **Log source Identifier** as Host Name of machine where Secret Server is installed.
    5. Click **Save**.



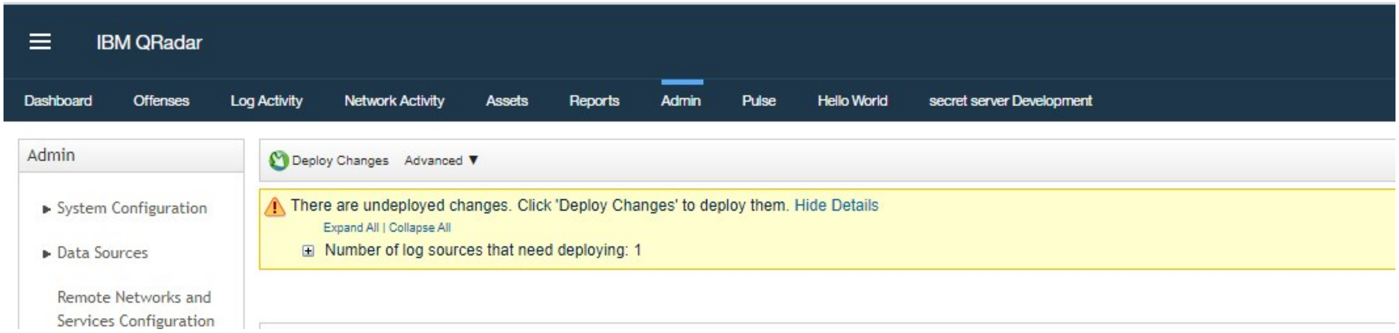
Add a log source

Log Source Name	
Log Source Description	
Log Source Type	3Com 8800 Series Switch
Protocol Configuration	Syslog
Log Source Identifier	
Enabled	☑
Credibility	5
Target Event Collector	eventcollector0 :: ibmqradar
Coalescing Events	☑
Incoming Payload Encoding	UTF-8
Store Event Payload	☑
Log Source Extension	Select an Extension...

Please select any groups you would like this log source to be a member of:

Deploy Log Source

1. On the **Admin Page of QRadar**, click on **deploy changes** button.



There are undeployed changes. Click 'Deploy Changes' to deploy them. Hide Details

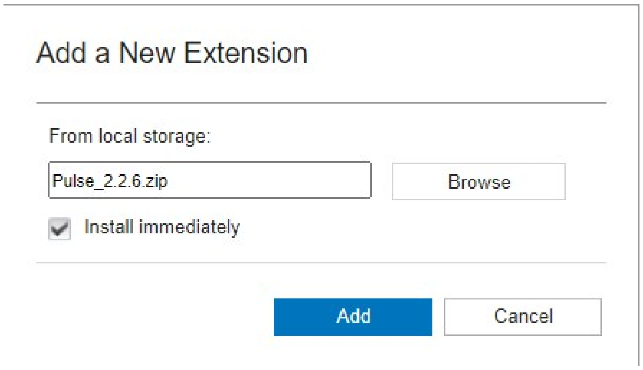
Expand All | Collapse All

⊞ Number of log sources that need deploying: 1

Pulse Dashboard Setup

To Configure Pulse Dashboard Setup (The QRadar integration is also available at Thycotic.com):

1. Click on **Add | Browse the pulse Zip file | Check “install immediately”**.
2. Click on the **Add** button.

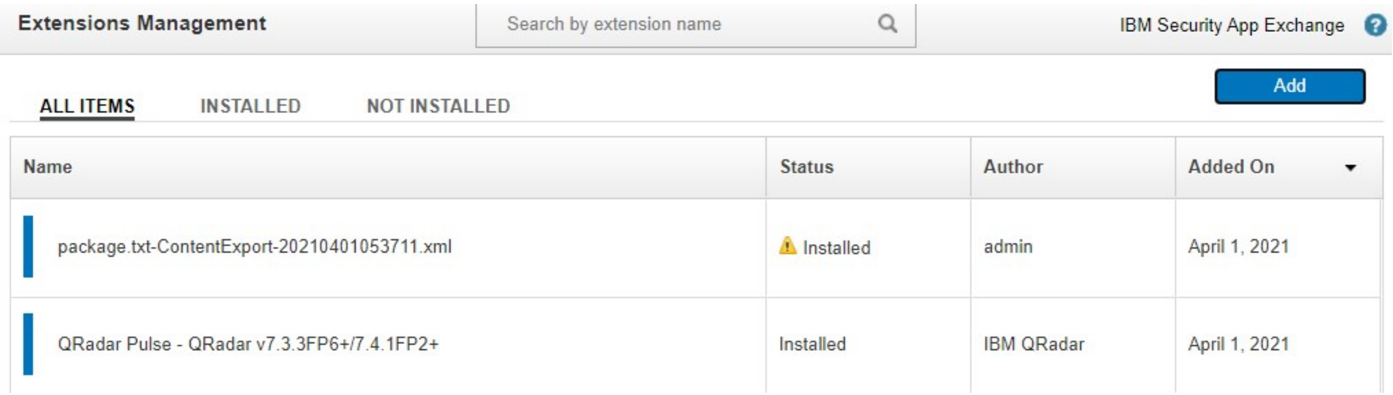


Add a New Extension

From local storage: Pulse_2.2.6.zip

☑ Install immediately

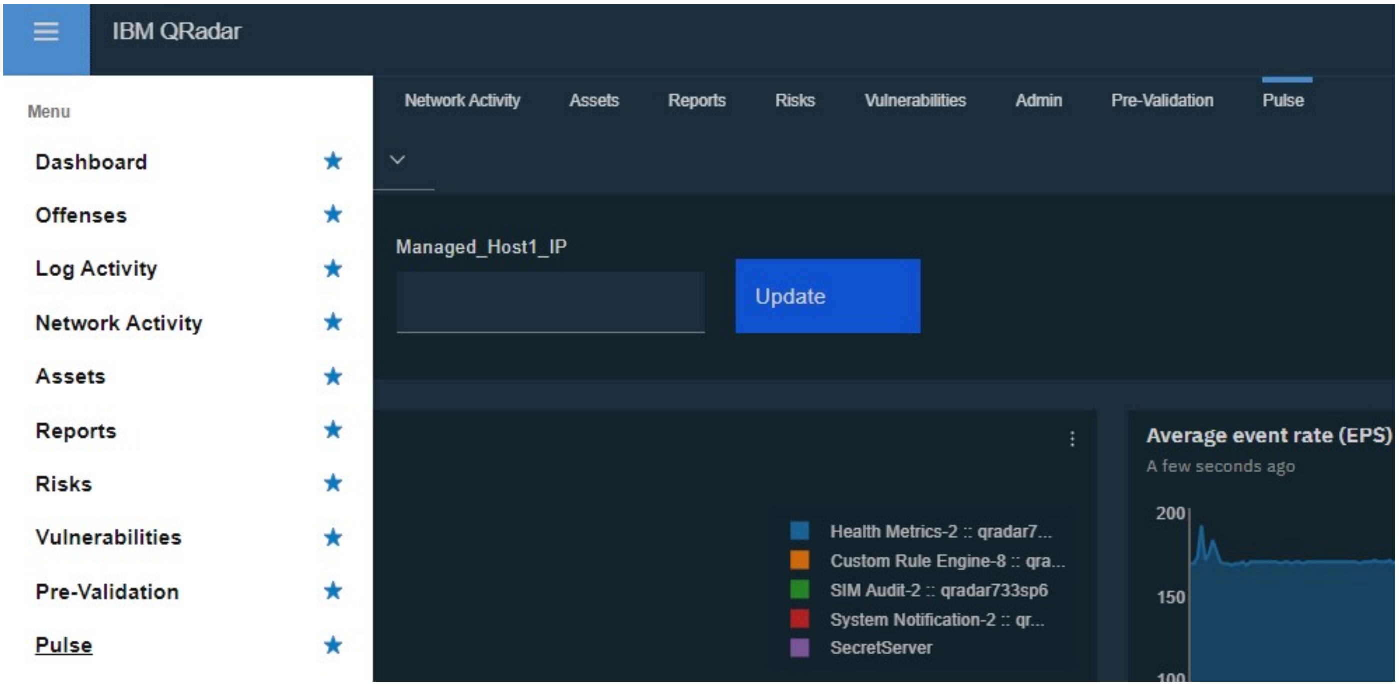
3. You should see the pulse QRadar pulse app in the extension management.



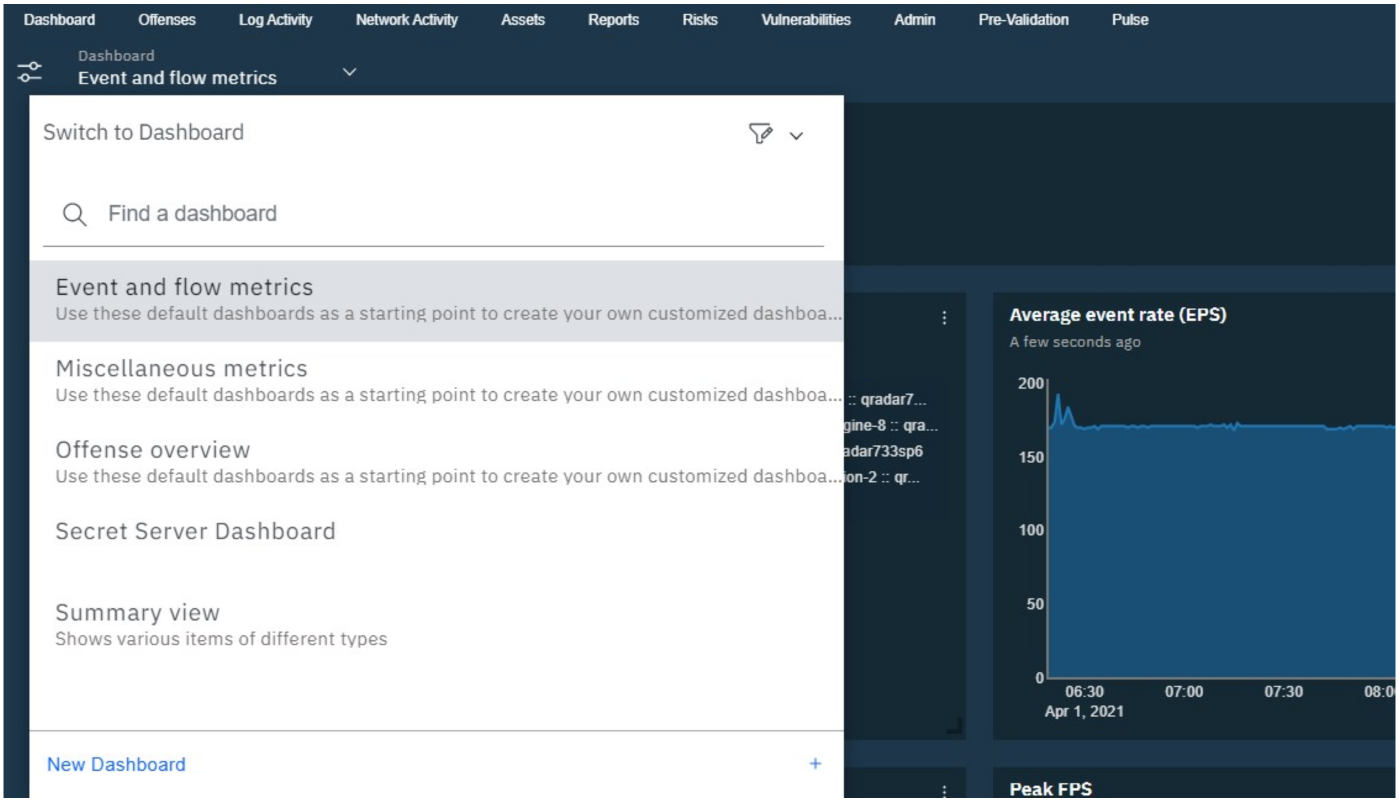
Extensions Management

Name	Status	Author	Added On
package.txt-ContentExport-20210401053711.xml	⚠ Installed	admin	April 1, 2021
QRadar Pulse - QRadar v7.3.3FP6+/7.4.1FP2+	Installed	IBM QRadar	April 1, 2021

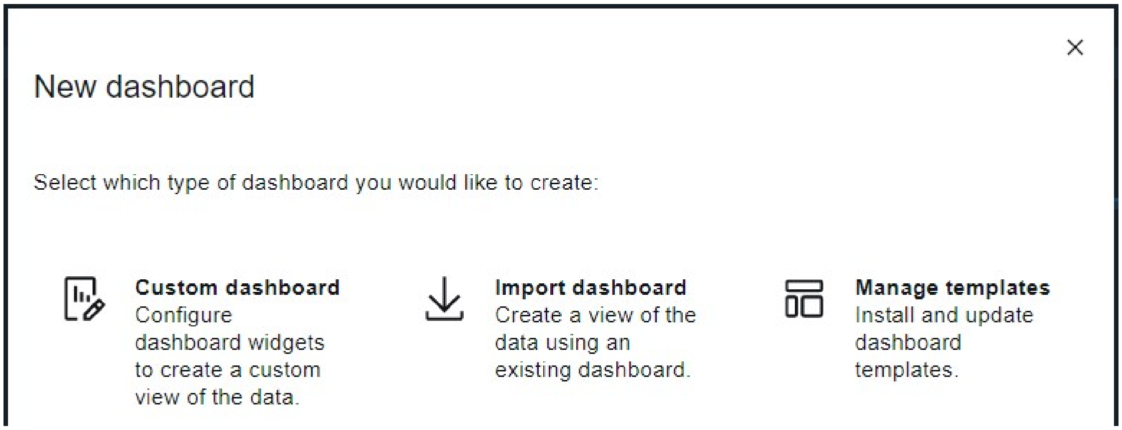
4. Click on the **Pulse** Extension from the menu.



5. Click on the dashboard dropdown and select **New Dashboard**.



6. Click on **Import Dashboard**.



New dashboard

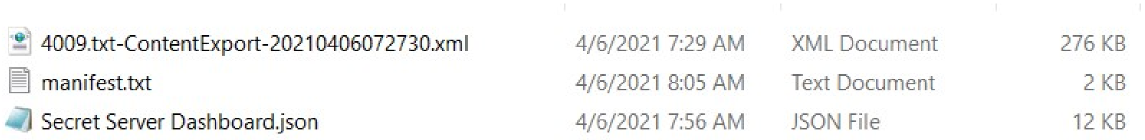
Select which type of dashboard you would like to create:

Custom dashboard Configure dashboard widgets to create a custom view of the data.

Import dashboard Create a view of the data using an existing dashboard.

Manage templates Install and update dashboard templates.

7. Extract the **Thycotic Secret Server dashboard zip**.

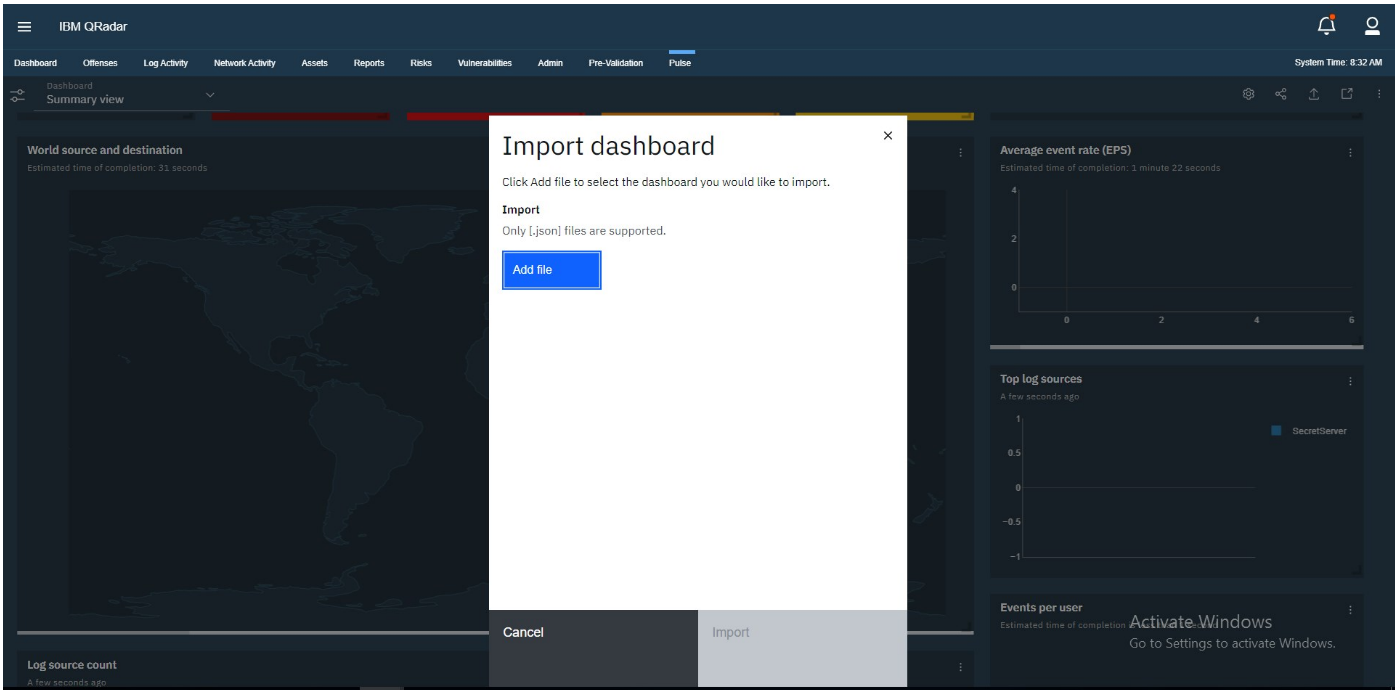


4009.txt-ContentExport-20210406072730.xml	4/6/2021 7:29 AM	XML Document	276 KB
manifest.txt	4/6/2021 8:05 AM	Text Document	2 KB
Secret Server Dashboard.json	4/6/2021 7:56 AM	JSON File	12 KB

8. Click on **Add File**.

9. Navigate to **Secret Server Dashboard.json**.

10. Click on the **Import** button.



Import dashboard

Click Add file to select the dashboard you would like to import.

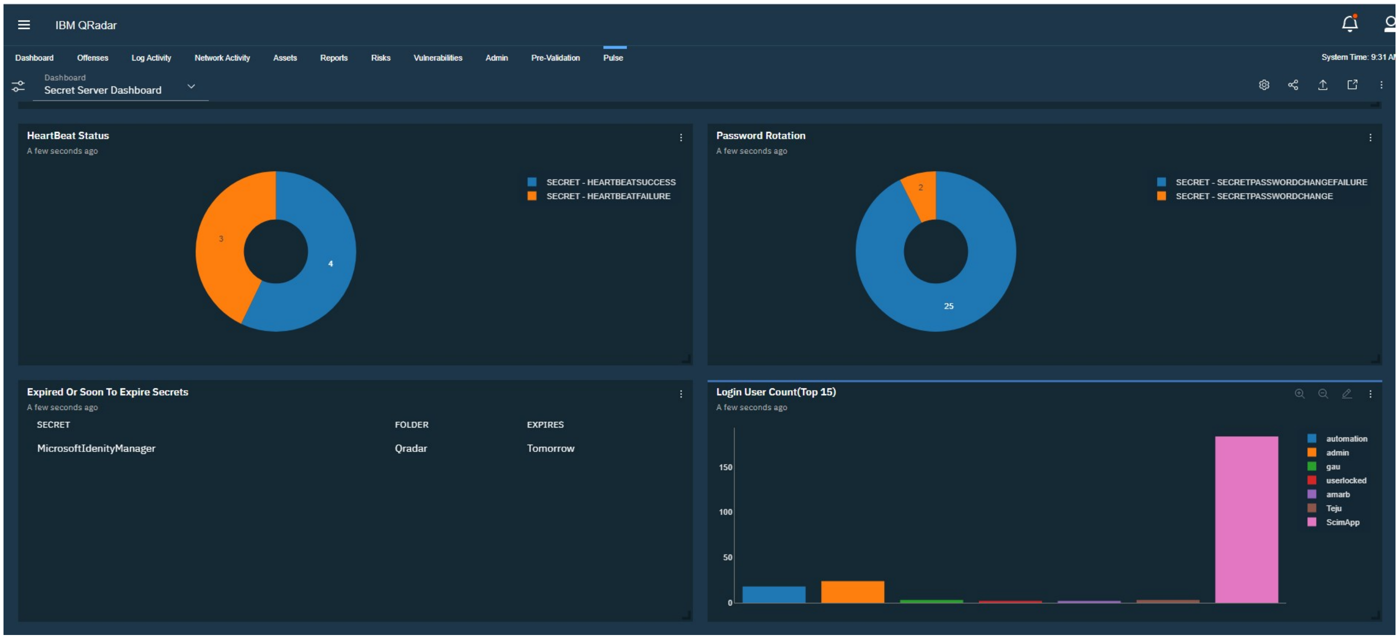
Import

Only [.json] files are supported.

Add file

Cancel | Import

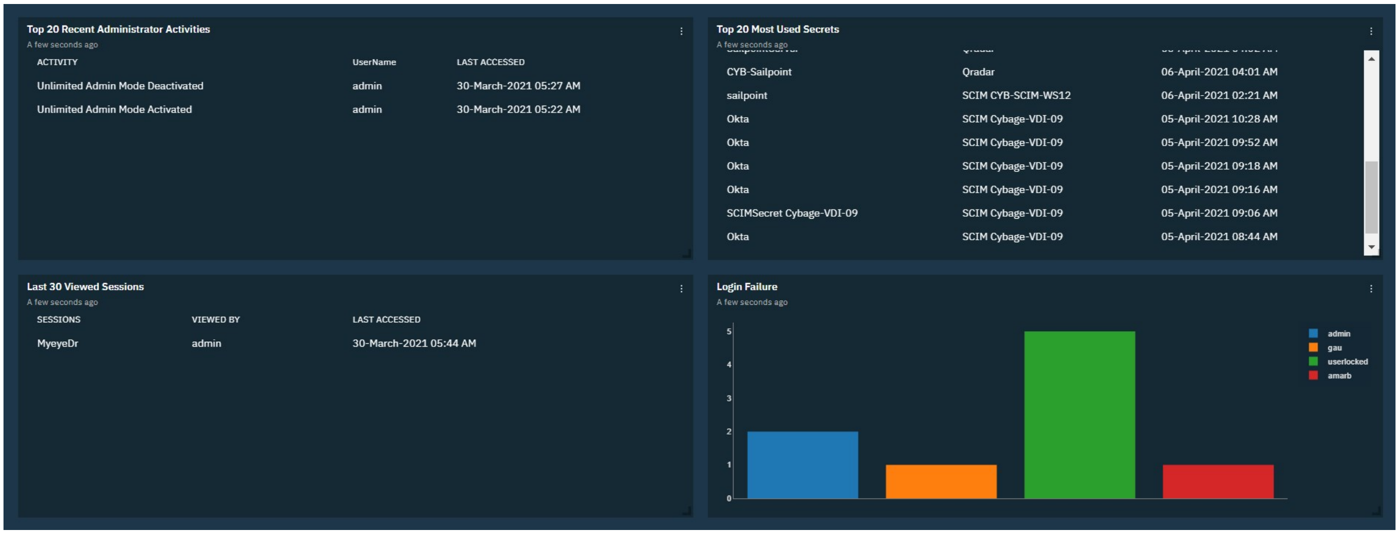
11. The **Secret Server Dashboard** will be displayed in Pulse.



Expired Or Soon To Expire Secrets

A few seconds ago

SECRET	FOLDER	EXPIRES
MicrosoftIdenityManager	Qradar	Tomorrow



Top 20 Recent Administrator Activities

A few seconds ago

ACTIVITY	UserName	LAST ACCESSED
Unlimited Admin Mode Deactivated	admin	30-March-2021 05:27 AM
Unlimited Admin Mode Activated	admin	30-March-2021 05:22 AM

Top 20 Most Used Secrets

A few seconds ago

CYB-Sailpoint	Qradar	06-April-2021 04:01 AM
sailpoint	SCIM CYB-SCIM-WS12	06-April-2021 02:21 AM
Okta	SCIM Cybage-VDI-09	05-April-2021 10:28 AM
Okta	SCIM Cybage-VDI-09	05-April-2021 09:52 AM
Okta	SCIM Cybage-VDI-09	05-April-2021 09:18 AM
Okta	SCIM Cybage-VDI-09	05-April-2021 09:16 AM
SCIMSecret Cybage-VDI-09	SCIM Cybage-VDI-09	05-April-2021 09:06 AM
Okta	SCIM Cybage-VDI-09	05-April-2021 08:44 AM

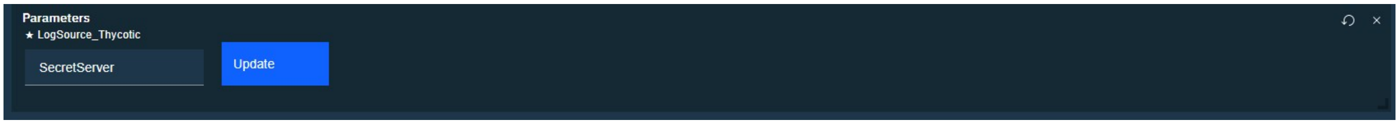
Last 30 Viewed Sessions

A few seconds ago

SESSIONS	VIEWED BY	LAST ACCESSED
MyeyeDr	admin	30-March-2021 05:44 AM

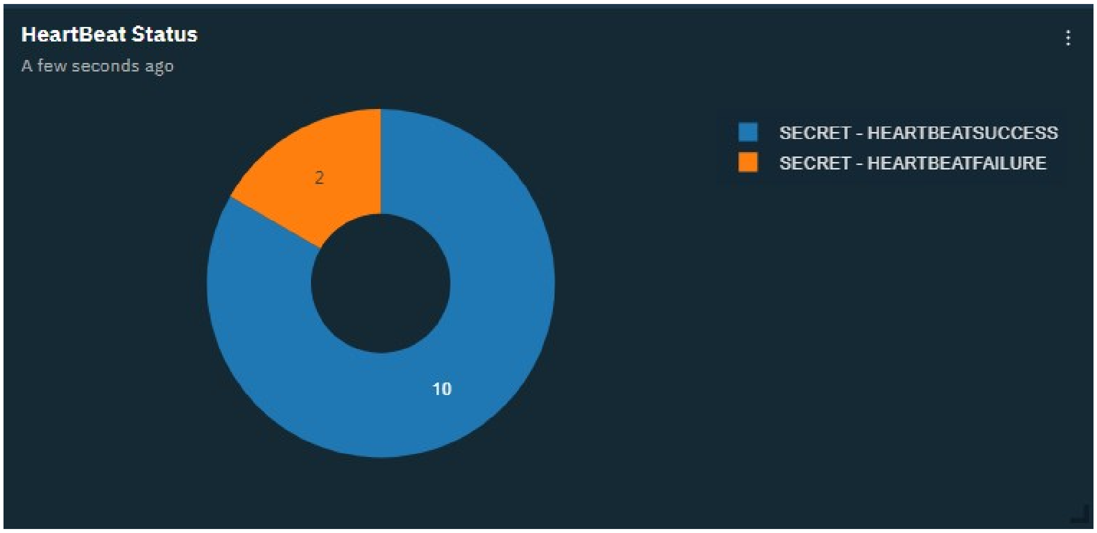
Secret Server Dashboard Widget

- Add **Log Source Name** to the **LogSource_Thycotic** Parameters.

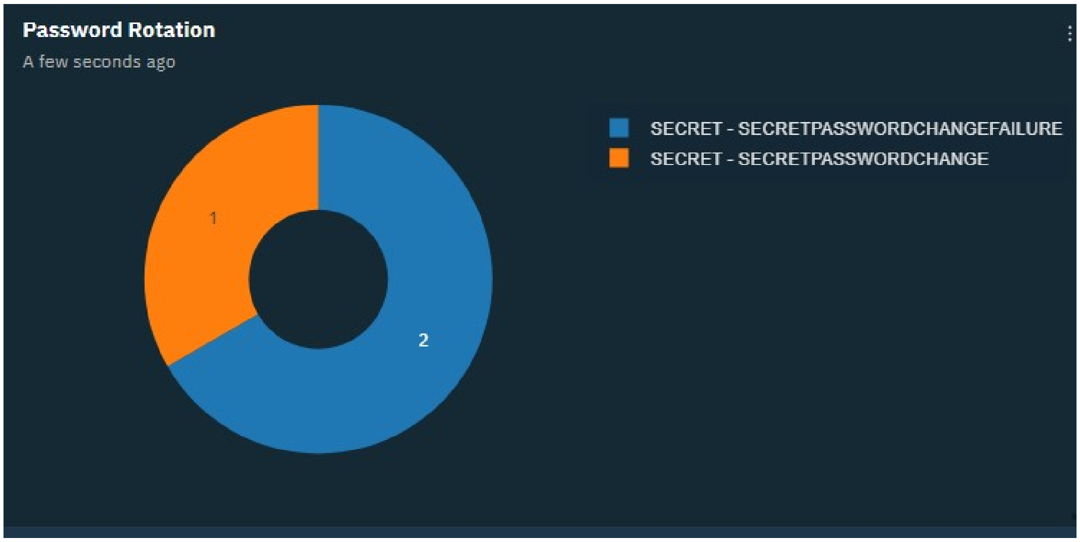


SecretServer

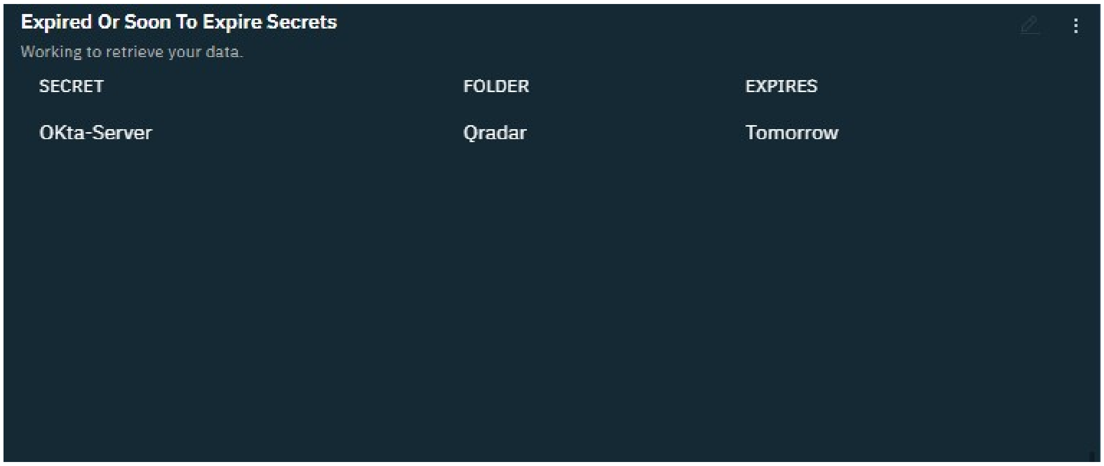
- **Heartbeat Status**
  - **Success:** The credentials in the Secret authenticated successfully with the target system.
  - **Failed:** The credentials in the Secret failed authentication with the target system.



- **Password Rotation**
  - **Success:** A Secret Password has changed.
  - **Failed:** A Secret Password has failed to change.



- **Secret expired or Soon to be expired**

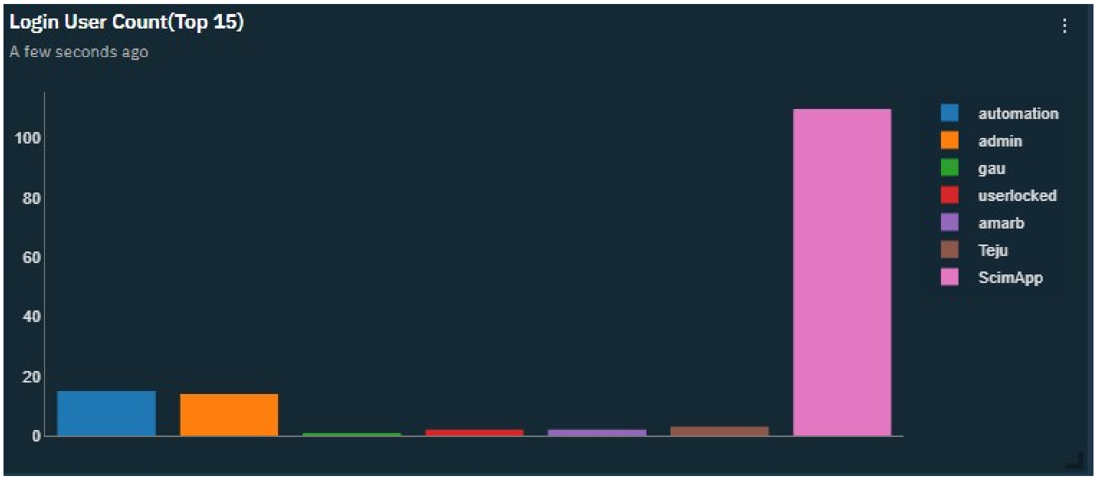


Expired Or Soon To Expire Secrets

Working to retrieve your data.

SECRET	FOLDER	EXPIRES
OKta-Server	Qradar	Tomorrow

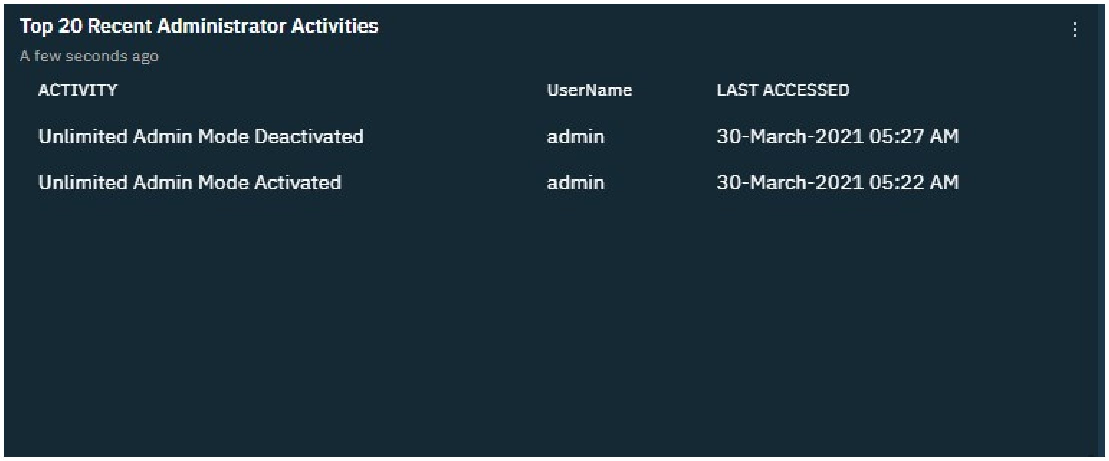
- **Top 15 login user count**



Login User Count(Top 15)

A few seconds ago

- **Top 20 Recent Administrator Activities**

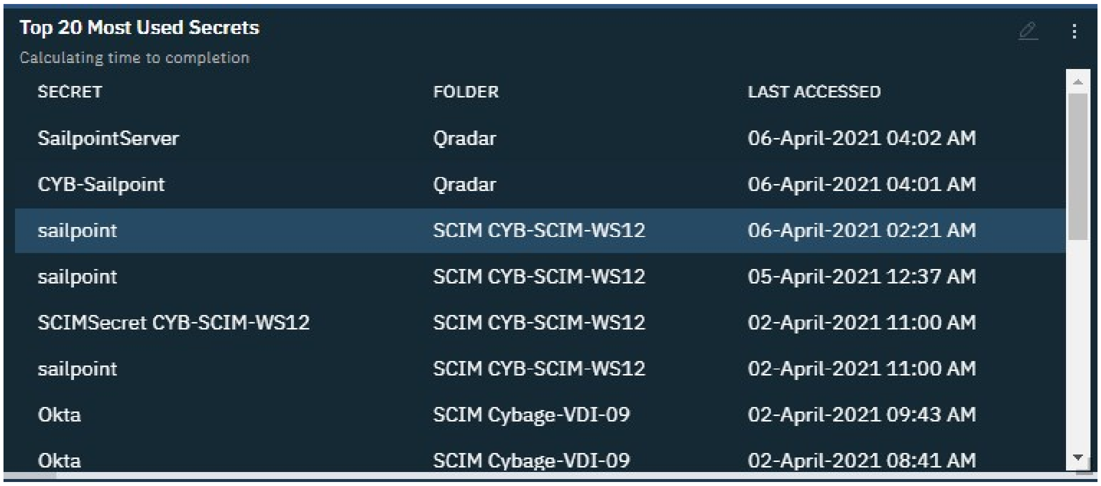


Top 20 Recent Administrator Activities

A few seconds ago

ACTIVITY	UserName	LAST ACCESSED
Unlimited Admin Mode Deactivated	admin	30-March-2021 05:27 AM
Unlimited Admin Mode Activated	admin	30-March-2021 05:22 AM

- **Top 20 Most Used Secrets**

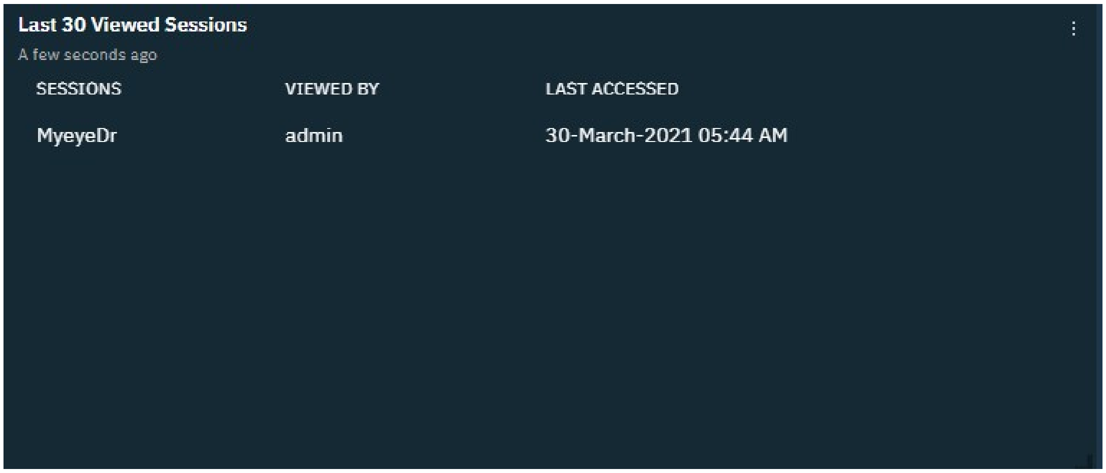


Top 20 Most Used Secrets

Calculating time to completion

SECRET	FOLDER	LAST ACCESSED
SailpointServer	Qradar	06-April-2021 04:02 AM
CYB-Sailpoint	Qradar	06-April-2021 04:01 AM
sailpoint	SCIM CYB-SCIM-WS12	06-April-2021 02:21 AM
sailpoint	SCIM CYB-SCIM-WS12	05-April-2021 12:37 AM
SCIMSecret CYB-SCIM-WS12	SCIM CYB-SCIM-WS12	02-April-2021 11:00 AM
sailpoint	SCIM CYB-SCIM-WS12	02-April-2021 11:00 AM
Okta	SCIM Cybage-VDI-09	02-April-2021 09:43 AM
Okta	SCIM Cybage-VDI-09	02-April-2021 08:41 AM

- **Last 30 Viewed Sessions**

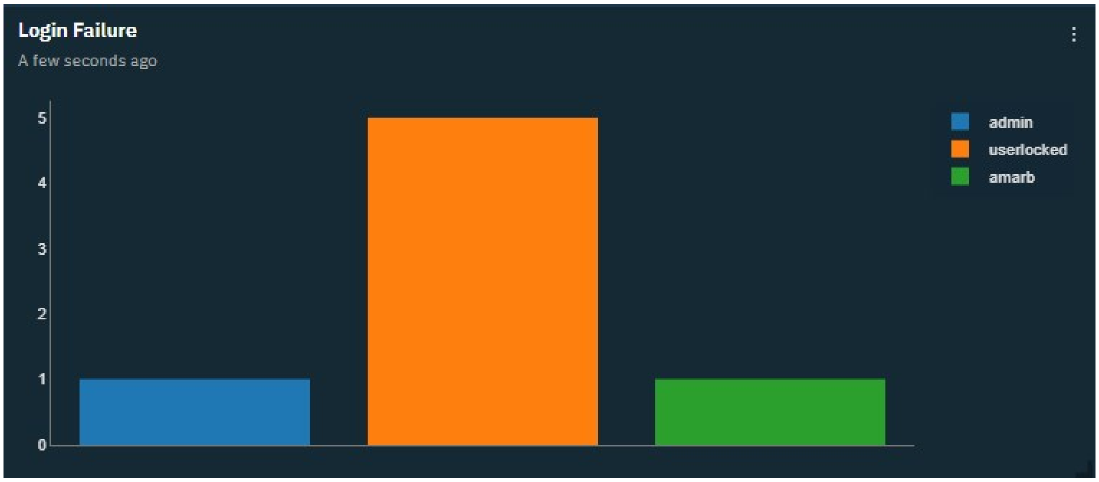


Last 30 Viewed Sessions

A few seconds ago

SESSIONS	VIEWED BY	LAST ACCESSED
MyeyeDr	admin	30-March-2021 05:44 AM

- **Last 30 Login Failure Users**



Support

Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Thycotic customers have access to support by phone and email. You also can open a case in Thycotic's support ticketing system, which promotes follow-through to issue resolution.

> **Note**: Please see our Support Services Guide for details about our support policy. This page provides a high-level summary of portions of that guide.

Use the means you prefer, except for Severity 1 issues—for those, always use phone support.

Severity 1 means a critical problem that has caused *complete loss of service* and work cannot reasonably continue at your worksite.

Obtaining a Support PIN

To obtain support by email or phone, first log in to the Support Portal to obtain a PIN. The PIN validates that your license includes support, and you must provide the PIN in your email or when you call. The PIN also makes it easier for the person helping you to locate your customer records and give you better support.

- Visit the Support Portal Login Page using the credentials you received when you became a customer.
- After logging in, you will be on the main page. Click on the large blue bar labeled PIN to obtain a PIN number.

Support by Phone

Thycotic delivers support by phone worldwide. Select the applicable number from this list:

Region	Country	Support Number
AMERICAS	all	+1 202 991 0540
EMEA	UK	+44 20 3880 0017
	Germany	+49 69 6677 37597
APAC	Australia	+61 3 8595 5827
	Philippines	+63 2 231 3885
	New Zealand	+64 9-887 4015
	Singapore	+65 3157 0602

Support by Email

Send your email to support@thycotic.com **with the PIN number as part of the subject line** of your email, for example:

- PIN 345 Workflow Stopped Unexpectedly

Include this information:

1. company name
2. contact name
3. contact phone number
4. product name

5. details of the issue

You must send your email using an email address already noted in your account with Thycotic.

- Sending a support request from an email address not on file may delay our response.

Support Ticketing

As an alternative to support by email or phone, you can open a support ticket and track your issue to resolution.

- Visit the Support Portal Login Page using the credentials you received when you became a customer.
- After logging in, you will be on the main page. Click the **Cases** tab, then **Create a Case**.
- Follow the instructions to complete your case.

IBM Verify Gateway for RADIUS Server

Note: Integration details coming soon

This document discusses integrating IBM Verify Gateway for Remote Access Dial-In User (RADIUS) Server with Thycotic Verify Privilege Vault.

Thycotic IT security and password management solutions empower companies to remove the complexities of proper access control and management of privileged accounts. An Inc. 5000 company, Thycotic is trusted by more than 3,000 organizations worldwide—including Fortune 500 members, enterprises, government agencies, technology firms, universities, non-profits, and managed service providers. To learn more, please visit thycotic.com.

Verify Privilege Vault

Thycotic Verify Privilege Vault (SS) is an on-premises Web-based password vault used throughout the world to help organizations properly manage privileged account passwords. SS allows users to control access and automate password changes for a variety of enterprise resources, such as servers, databases, network devices, and applications. SS features auditing throughout the application and role-based access control (RBAC) on all its information and features. Organizations can easily deploy SS to ensure security, reduce labor costs, adopt password best practices, and satisfy audit requirements.

IBM Verify Gateway for RADIUS Server

IBM Verify Gateway for RADIUS component includes a RADIUS interface where VPNs and other RADIUS clients can request authentication making use of the multi-factor authentication mechanisms from the Cloud.

IBM Verify Gateway for RADIUS documentation and installation can be found in the IBMCloud.

IBM Red Hat

- OpenShift

OpenShift Deployment

The integration provided by Thycotic for managing OpenShift Secrets is a Mutating Admissions Webhook. The webhook functions by intercepting Kubernetes Secrets calls that feature the webhook's annotation and translating these in to requests for Secrets from Thycotic Secret Server.

This guide is designed to walk you through how the integration fits together with all its constituent parts, and gives some example templates (in OpenShift YAML), designed to get you up and running with the integration quickly and effectively.

As a reference, the integration itself is available open source from here, hence you're able to modify it to suit the specific needs of your organization:

https://github.com/thycotic/tss-k8s

https://github.com/thycotic/dsv-k8s

The integration also backs off to the Thycotic golang integration components, which are also available open source:

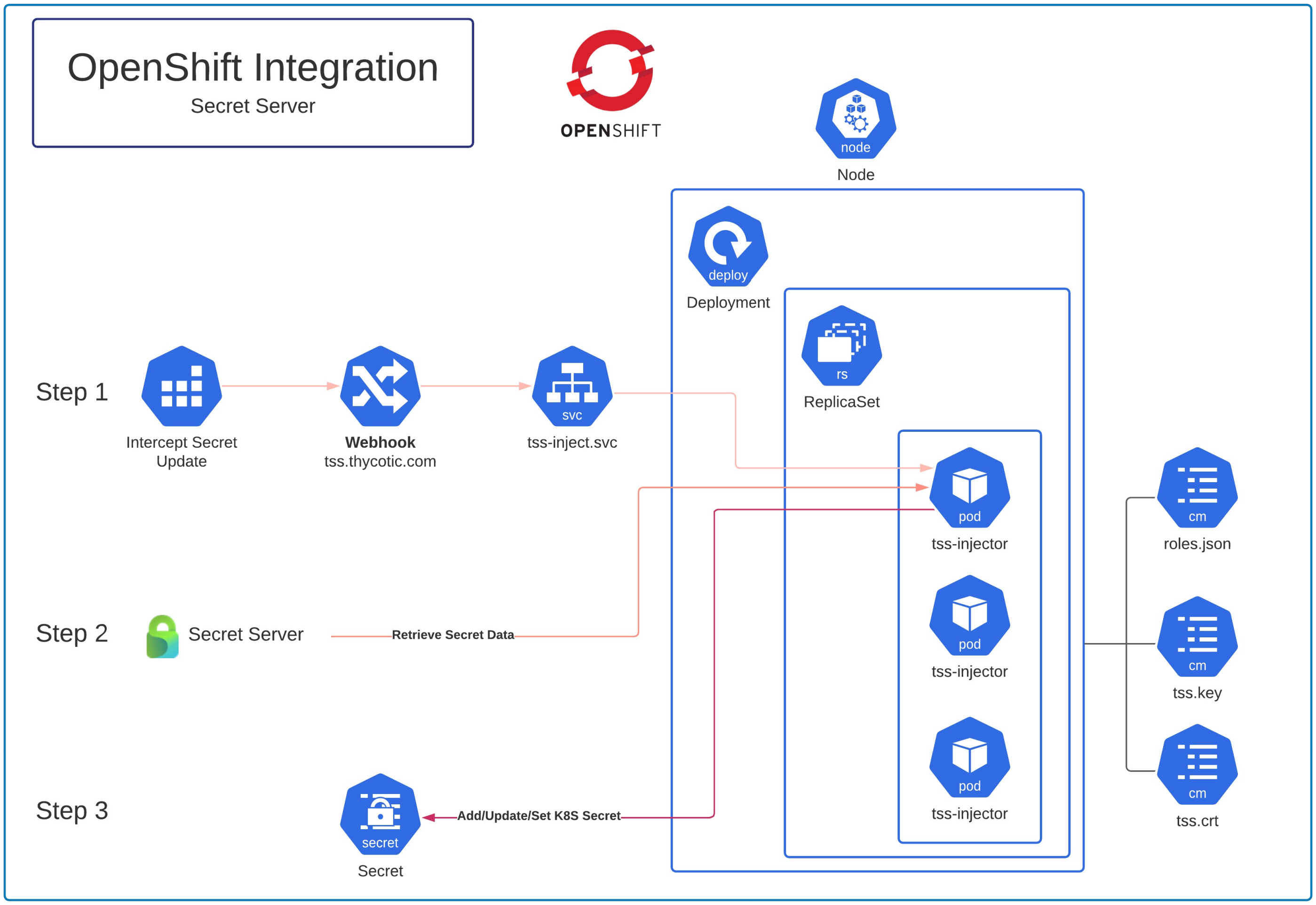
https://github.com/thycotic/tss-sdk-go

https://github.com/thycotic/dsv-sdk-go

> **Note**: This guide is tailored towards OpenShift deployment, however the method is also generally supported and cross-compatible on Kubernetes.

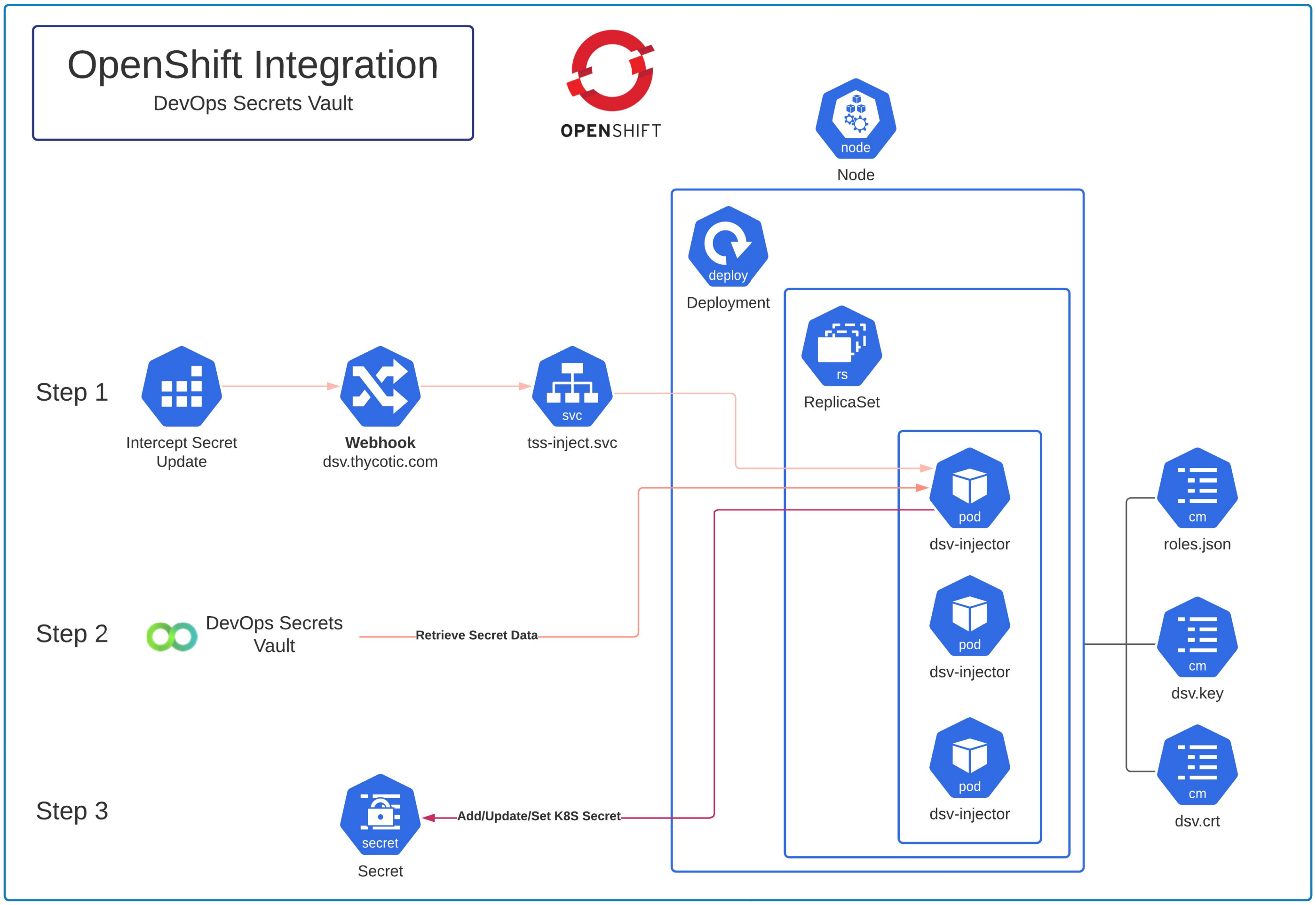
Architecture

Secret Server



The diagram above represents the visual version of the architecture and its components.

DSV



The diagram above represents the visual version of the architecture and its components.

Integration with Secret Server

The integration is a Mutating Admissions Webhook that intercepts requests for OpenShift Secrets using a specialized annotation. The request is then updated with data from Secret Server and passed in to the OpenShift Secrets vault. This ensures that credentials in the OpenShift Secrets Vault are aligned with the values as managed by Secret Server, hence password changes can occur on the Secret Server side and be able to be reflected in the OpenShift Secrets Vault.

The deployment is designed to be easily integrated with existing deployment environments, as Secret requests only need to have the various Annotations added to them in order for the Secret Server based workflow to be enacted.

> **Note**: There are numerous different ways of configuring OpenShift for different operating environments. Hence this guide, although intending to give a solid baseline idea for deployment of the integration, will not be the sole, authoritative way in which the integration can function or be deployed. All examples below use the default namespace which, alongside some other components, will likely need to be modified to ensure suitability with your organization's OpenShift environment.

The Webhook

The webhook is at the front end of the integration and intercepts the requests for Secrets that are inbound to the OpenShift Secrets store, when the requests are given the appropriate annotation. Below is a basic configuration YAML for deploying the webhook in your OpenShift instance.

Example Webhook YAML

```
---
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingWebhookConfiguration
metadata:
  name: tss-injector
  labels:
    app: tss
webhooks:
  - name: tss.thycotic.com
    rules:
      - apiGroups: ["*"]
        apiVersions: ["*"]
        operations: ["CREATE", "UPDATE"]
        resources: ["secrets"]
    clientConfig:
      service:
        namespace: default
        name: tss-injector
        path: "/inject"
        port: 8543
      caBundle: ""
    admissionReviewVersions: ["v1","v1beta1"]
    sideEffects: None
    timeoutSeconds: 5
```

The Webhook Certificate

Each of the pods in the deployment are in possession (via ConfigMaps) of a certificate that they present in order to identify themselves to the OpenShift instance. The caBundle value in the webhook must be the base64 encoded version of the public certificate (crt) that the pods are presenting.

The Service

As per standard OpenShift configuration, a load balanced service allows orchestrated applications to be handled effectively from an internal OpenShift networking standpoint. Hence, we want this service to direct all requests to the appropriate deployment/pods when an annotated Secret request comes in.

Below is an example of the service and how it could look against the default namespace:

Example Service YAML

```
---
```

```
apiVersion: v1
kind: Service
metadata:
  name: tss-injector
  namespace: default
  labels:
    app: tss-injector
spec:
  ports:
    - port: 8543
      targetPort: 18543
  selector:
    app: tss-injector
  type: LoadBalancer
```

ConfigMaps

Roles Configuration

The roles.json file is accessed by the pods in the deployment through a ConfigMap. The file gives the pods information about where Secret Server is located, and how they should authenticate with it (via a set of credentials).

Secret Server Cloud Example

```
{
    "default": {
        "credentials": {
            "username": "username",
            "password": "password"
        },
        "serverURL": "https://mytenant.secretsvaultcloud.com"
    }
}
```

Multiple roles and vaults can also be used.

> **Note**: In the absence of a role being explicitly specified, the default role will be used.

Multiple Vaults Example

```
{
    "alternaterole": {
        "credentials": {
            "username": "username",
            "password": "password"
        },
        "serverURL": "https://myfirsttenant.secretsvaultcloud.com"
    },
    "default": {
        "credentials": {
            "username": "username",
            "password": "password"
        },
       "serverURL": "https://mysecondtenant.secretsvaultcloud.com"
    }
}
```

Certificate

Two ConfigMaps are required for the certificate. One to hold the public certificate (.crt), and one for the private key associated therewith (.key).

IMPORTANT The public certificate must have a CN (Command Name) of deploymentname.namespace.svc. To fit directly in to the examples given here, a CN of tss-injector.default.svc must be present on the certificate.

Examples of these config maps (tss-crt and tss-key) are in the Sample Deployment YAML under volumes.

Container Mapping: These items are mapped in to the injector container through the ENTRYPOINT in the Dockerfile, which is, in its most basic form:

```
FROM tss-injector:latest
```

```
ARG cert_file
ARG key_file
ARG roles_file
COPY ${cert_file} ./tss.pem
COPY --chown=tss ${key_file} ./tss.key
COPY ${roles_file} ./roles.json
ENTRYPOINT ["tss-injector-svc", "-cert", "tss.pem", "-key", "tss.key", "-roles", "roles.json" ]
```

> **Note**: The tss-injector image on Docker Hub does not include these references, however the Docker Hub image reference (thycotictc/openshift:latest) in the Deployment below already includes the above configuration.

The Deployment

The final component of the integration is the deployment and the pods associated with it. Each pod includes a single running container that is the injector application, which is designed to go out to the target vaulting platform and retrieve the intended Secret values.

Note that the deployment example below includes all of the configuration input as detailed in the sections above.

Example Deployment YAML

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: tss-injector
  namespace: default
  labels:
    app: tss-injector
spec:
  replicas: 5
  selector:
    matchLabels:
      app: tss-injector
  template:
    metadata:
      labels:
        app: tss-injector
      namespace: default
    spec:
      containers:
        - image: thycotictc/openshift:latest
          name: tss-injector
          command: ["tss-injector-svc", "-cert", "tss.crt", "-key", "tss.key", "-roles", "roles.json" ]
          workingDir: "/config"
          resources:
            requests:
              memory: "512Mi"
              cpu: "250m"
            limits:
              memory: "2048Mi"
              cpu: "1000m"
          ports:
            - containerPort: 18543
              name: tss
          volumeMounts:
          - name: config-volume
            mountPath: /config
      volumes:
        - name: config-volume
          projected:
            sources:
              - configMap:
                  name: tss-config
                  items:
                  - key: roles.json
                    path: roles.json
              - configMap:
                  name: tss-key
                  items:
                  - key: tss.key
                    path: tss.key
              - configMap:
                  name: tss-cert
                  items:
                  - key: tss.crt
                    path: tss.crt
```

The Requests

Annotations

The four annotations that affect the behavior of the webhook are:

```
const(
  roleAnnotation   = "tss.thycotic.com/role"
  setAnnotation    = "tss.thycotic.com/set-secret"
  addNotation      = "tss.thycotic.com/add-to-secret"
  updateAnnotation = "tss.thycotic.com/update-secret"
)
```

- roleAnnotation identifies the role that should be used, as per the roles.json entry
- addAnnotation adds missing fields without overwriting or removing existing fields.
- updateAnnotation adds and overwrites existing fields but does not remove fields.
- setAnnotation overwrites fields and removes fields that do not exist in the TSS Secret.

> **Note**: Only one of these should be specified on any given k8s Secret, however, if more than one are defined then the order of precedence is setAnnotation then addAnnotation then updateAnnotation.

Secret Examples

In addition to the annotation, the secretID value will also need to be provided. This corresponds to the value of the target Secret within Secret Server from which data needs to be retrieved.

> **Note**: The data fields on the request itself are generally ignored, depending on the annotation used.

SET

```
---
apiVersion: v1
kind: Secret
metadata:
  name: my-secret-name
  annotations:
    tss.thycotic.com/set-secret: "10"
type: Opaque
data:
  data: dW5tb2RpZmllZC11c2VybmFtZQ==
```

ADD

```
---
apiVersion: v1
kind: Secret
metadata:
  name: user-domain
  annotations:
    tss.thycotic.com/add-to-secret: "10"
type: Opaque
data:
  data: dW5tb2RpZm
```

Integration with DSV

The integration is a Mutating Admissions Webhook that intercepts requests for OpenShift Secrets using a specialized annotation. The request is then updated with data from DevOps Secrets Vault and passed in to the OpenShift Secrets vault. This ensures that credentials in the OpenShift Secrets Vault are aligned with the values as managed by DevOps Secrets Vault, hence password changes can occur on the DevOps Secrets Vault side and be able to be reflected in the OpenShift Secrets Vault.

The deployment is designed to be easily integrated in to existing deployment environments, as Secret requests only need to have the various Annotations added to them in order for the DevOps Secrets Vault based workflow to be enacted.

> **Note**: There are numerous different ways of configuring OpenShift for different operating environments. Hence this guide, although intending to give a solid baseline idea for deployment of the integration, will not be the sole, authoritative way in which the integration can function or be deployed. All examples below use the default namespace which, alongside some other components, will likely need to be modified to ensure suitability with your organization's OpenShift environment.

The Webhook

The webhook is at the front end of the integration and intercepts the requests for Secrets that are inbound to the OpenShift Secrets store, when the requests are given the appropriate annotation. Below is a basic configuration YAML for deploying the webhook in your OpenShift instance.

Example Webhook YAML

```
---
apiVersion: admissionregistration.k8s.io/v1
kind: MutatingWebhookConfiguration
metadata:
  name: dsv-injector
  labels:
    app: dsv
webhooks:
  - name: dsv.thycotic.com
    rules:
      - apiGroups: ["*"]
        apiVersions: ["*"]
        operations: ["CREATE", "UPDATE"]
        resources: ["secrets"]
    clientConfig:
      service:
        namespace: default
        name: dsv-injector
        path: "/inject"
        port: 8543
      caBundle: ""
    admissionReviewVersions: ["v1","v1beta1"]
    sideEffects: None
    timeoutSeconds: 5
```

The Webhook Certificate

Each of the pods in the deployment are in possession (via ConfigMaps) of a certificate that they present in order to identify themselves to the OpenShift instance. The caBundle value in the webhook must be the base64 encoded version of the public certificate (crt) that the pods are presenting.

The Service

As per standard OpenShift configuration, a load balanced service allows orchestrated applications to be handled effectively from an internal OpenShift networking standpoint. Hence, we want this service to direct all requests to the appropriate deployment/pods when an annotated Secret request comes in.

Below is an example of the service and how it could look against the default namespace:

Example Service YAML

```
---
```

```
apiVersion: v1
kind: Service
metadata:
  name: dsv-injector
  namespace: default
  labels:
    app: dsv-injector
spec:
  ports:
    - port: 8543
      targetPort: 18543
  selector:
    app: dsv-injector
  type: LoadBalancer
```

ConfigMaps

Roles Configuration

The roles.json file is accessed by the pods in the deployment through a ConfigMap. The file gives the pods information about where DevOps Secrets Vault is located, and how they should authenticate with it (via a set of credentials).

DevOps Secrets Vault Example

```
{
  "default": {
    "credentials": {
      "clientid": "<ClientID>",
      "clientsecret": "<ClientSecret>"
    },
     //  "TLD":"eu", Optional for non-US instances
    "tenant": "tenantname"
  }
}
```

Multiple Vaults Example

Multiple roles and vaults can also be used.

> **Note**: In the absence of a role being explicitly specified, the default role will be used.

```
{
  "alternaterole": {
   "credentials": {
      "clientid": "<ClientID>",
      "clientsecret": "<ClientSecret>"
    },
      "tenant": "tenantname"
  },
  "default": {
    "credentials": {
      "clientid": "<ClientID>",
      "clientsecret": "<ClientSecret>"
    },
     "tenant": "anothertenantname"
  }
}
```

Certificate

Two ConfigMaps are required for the certificate. One to hold the public certificate (.crt), and one for the private key associated therewith (.key).

> **IMPORTANT**: The public certificate must have a CN (Command Name) of deploymentname.namespace.svc. To fit directly in to the examples given here, a CN of dsv-injector.default.svc must be present on the certificate.

Examples of these config maps (dsv-crt and dsv-key) are in the Sample Deployment YAML under volumes.

Container Mapping

These items are mapped in to the injector container through the ENTRYPOINT in the Dockerfile, which is, in its most basic form:

```
FROM dsv-injector:latest
ARG cert_file
ARG key_file
ARG roles_file
COPY ${cert_file} ./dsv.pem
COPY --chown=dsv ${key_file} ./dsv.key
COPY ${roles_file} ./roles.json
ENTRYPOINT ["dsv-injector-svc", "-cert", "dsv.pem", "-key", "dsv.key", "-roles", "roles.json" ]
```

> **Note**: The dsv-injector image on Docker Hub does not include these references, however the Docker Hub image reference (thycotictc/openshiftdsv:latest) in the Deployment below already includes the above configuration.

The Deployment

The final component of the integration is the deployment and the pods associated with it. Each pod includes a single running container that is the injector application, which is designed to go out to the target vaulting platform and retrieve the intended Secret values.

Note that the deployment example below includes all of the configuration input as detailed in the sections above.

Example Deployment YAML

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dsv-injector
  namespace: default
  labels:
    app: dsv-injector
spec:
  replicas: 5
  selector:
    matchLabels:
      app: dsv-injector
  template:
    metadata:
      labels:
        app: dsv-injector
      namespace: default
    spec:
      containers:
        - image: thycotictc/openshiftdsv:latest
          name: dsv-injector
          command: ["dsv-injector-svc", "-cert", "dsv.crt", "-key", "dsv.key", "-roles", "roles.json" ]
          workingDir: "/config"
          resources:
            requests:
              memory: "512Mi"
              cpu: "250m"
            limits:
              memory: "2048Mi"
              cpu: "1000m"
          ports:
            - containerPort: 18543
              name: dsv
          volumeMounts:
          - name: config-volume
            mountPath: /config
      volumes:
        - name: config-volume
          projected:
            sources:
              - configMap:
                  name: dsv-config
                  items:
                  - key: roles.json
                    path: roles.json
              - configMap:
                  name: dsv-key
```

```
         items:
         - key: dsv.key
           path: dsv.key
       - configMap:
          name: dsv-cert
         items:
         - key: dsv.crt
           path: dsv.crt
```

The Requests

Annotations

The four annotations that affect the behavior of the webhook are:

```
const(
  roleAnnotation   = "dsv.thycotic.com/role"
  setAnnotation    = "dsv.thycotic.com/set-secret"
  addNotation      = "dsv.thycotic.com/add-to-secret"
  updateAnnotation = "dsv.thycotic.com/update-secret"
)
```

- roleAnnotation identifies the role that should be used, as per the roles.json entry
- addAnnotation adds missing fields without overwriting or removing existing fields.
- updateAnnotation adds and overwrites existing fields but does not remove fields.
- setAnnotation overwrites fields and removes fields that do not exist in the DSV Secret.

> **Note**: Only one of these should be specified on any given k8s Secret, however, if more than one are defined then the order of precedence is setAnnotation then addAnnotation then updateAnnotation.

Secret Examples

In addition to the annotation, the path value that leads to the Secret will also need to be provided. This corresponds to the value of the target Secret within DevOps Secrets Vault from which data needs to be retrieved.

> **Note**: The data fields on the request itself are generally ignored, depending on the annotation used.

SET

```
---
apiVersion: v1
kind: Secret
metadata:
  name: example-secret
  annotations:
    dsv.thycotic.com/set-secret: /folderpath/secretname
type: Opaque
data:
  username: dW5tb2RpZmllZC11c2VybmFtZQ==
  domain: dW5tb2RpZmllZC1kb21haW4=
```

ADD

```
---
apiVersion: v1
kind: Secret
metadata:
  name: example-secret
  annotations:
    dsv.thycotic.com/update-secret: /folderpath/secretname
type: Opaque
data:
  data: dW5tb2RpZmllZC11c2VybmFtZQ==
```

Support

Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Thycotic customers have access to support by phone and email. You also can open a case in Thycotic's support ticketing system, which promotes follow-through to issue resolution.

> **Note**: Please see our Support Services Guide for details about our support policy. This page provides a high-level summary of portions of that guide.

Use the means you prefer, except for Severity 1 issues—for those, always use phone support.

Severity 1 means a critical problem that has caused *complete loss of service* and work cannot reasonably continue at your worksite.

Obtaining a Support PIN

To obtain support by email or phone, first log in to the Support Portal to obtain a PIN. The PIN validates that your license includes support, and you must provide the PIN in your email or when you call. The PIN also makes it easier for the person helping you to locate your customer records and give you better support.

- Visit the Support Portal Login Page using the credentials you received when you became a customer.
- After logging in, you will be on the main page. Click on the large blue bar labeled PIN to obtain a PIN number.

Support by Phone

Thycotic delivers support by phone worldwide. Select the applicable number from this list:

Region	Country	Support Number
AMERICAS	all	+1 202 991 0540
EMEA	UK	+44 20 3880 0017
	Germany	+49 69 6677 37597
APAC	Australia	+61 3 8595 5827
	Philippines	+63 2 231 3885
	New Zealand	+64 9-887 4015
	Singapore	+65 3157 0602

Support by Email

Send your email to support@thycotic.com **with the PIN number as part of the subject line** of your email, for example:

- PIN 345 Workflow Stopped Unexpectedly

Include this information:

1. company name
2. contact name
3. contact phone number
4. product name

5. details of the issue

You must send your email using an email address already noted in your account with Thycotic.

- Sending a support request from an email address not on file may delay our response.

Support Ticketing

As an alternative to support by email or phone, you can open a support ticket and track your issue to resolution.

- Visit the Support Portal Login Page using the credentials you received when you became a customer.
- After logging in, you will be on the main page. Click the **Cases** tab, then **Create a Case**.
- Follow the instructions to complete your case.

IBM WebSphere

Note: Integration details coming soon.

IBM WebSphere Integration with Secret Server WebSphere is a set of Java-based tools (middleware and application server) from IBM that allows customers to create and manage sophisticated business Web sites. The central WebSphere tool is the WebSphere Application Server (WAS), an application server that a customer can use to connect Web site users with Java applications or servlets.

The integration between Secret Server and WebSphere ensures that:

- Passwords are securely vaulted in Secret Server
- Users can enable Secret Server Credential Provider (SSCP) to fetch credentials from the Secret Server for JDBC data sources
- Users can configure credentials retrieval on a global level, data source level and a combination of enterprise application and data source level
- Users can configure local credentials cache so credentials fetched from the Secret Server are cached in-memory

In combination, these tools allow developers and organizations that are leveraging Websphere to move away from statically assigned passwords to a centralized, audited and dynamic password and credential schema. Credentials are retrieved "on demand" and can be regularly rotated, without interfering with the underlying running of the Websphere platform, or the applications that depend upon it.

ID Agent

Below are the following integrations that are available with ID Agent:

- Radius

Introduction

The integration between Thycotic Secret Server and Radius is created and maintained by Radius. This document provides guidance and best practice for implementing the integration. It is based on the following publicly available documentation from the vendor and testing performed by Thycotic. Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

Thycotic

Thycotic IT security and password management solutions empower companies to remove the complexities of proper access control and management of privileged accounts. An Inc. 5000 company, Thycotic is trusted by more than 3,000 organizations worldwide—including Fortune 500 members, enterprises, government agencies, technology firms, universities, non-profits, and managed service providers. To learn more, please visit thycotic.com.

Secret Server

Thycotic Secret Server (SS) is an on-premises Web-based password vault used throughout the world to help organizations properly manage privileged account passwords. SS allows users to control access and automate password changes for a variety of enterprise resources, such as servers, databases, network devices, and applications. SS features auditing throughout the application and role-based access control (RBAC) on all its information and features. Organizations can easily deploy SS to ensure security, reduce labor costs, adopt password best practices, and satisfy audit requirements.

RADIUS

RADIUS is an acronym for Remote Access Dial-In User. An instance of the RADIUS service installation to which different devices may connect for network authentication or access.RADIUS Client: RADIUS clients are network access servers—such as wireless access points, 802.1X-capable switches, virtual private network (VPN) servers, and dial-up servers. This entry area shows the name and device IP.

Getting Started with Radius

Sign up for RADIUS

Check out the On-Demand signup page at https://authanvil.com/try-it-free.

Before you begin, please download the walk-through guide and follow along.

RADIUS On-Demand

RADIUS is designed to enhance your existing login systems and can protect the many systems, including:

- Windows servers and workstations.
- Linux and Unix servers and desktops.
- Terminal and remote desktop services, including RD Web Access and RemoteApp.
- Citrix Access Gateway, XenApp, and XenDesktop.
- Web applications running on IIS and Apache.
- Virtual private networks (VPN) and SSL-VPN.
- Firewalls, routers, and switches.
- RMM solutions like Kaseya VSA and LabTech.
- PSA solutions like Connectwise and Autotask.

For further information please visit:

- https://authanvil.com/try-it-free
- https://help.authanvil.com/hc/en-us/articles/218394478
- https://help.authanvil.com/hc/en-us/articles/115001913808-How-Should-I-setup-RADIUS-
- https://help.authanvil.com/hc/en-us/articles/218924497-Testing-RADIUS-Communication

Integration Requirements

Pre-requisites

- Visual C++ -This update can be downloaded here.
- Microsoft .NET v4.6 This update can be downloaded here.
- .Net 4.6.0 or Higher must be installed on the host machine or the RADIUS installation package will not install.

Configuration

Please review the following steps below to properly configure Radius for Secret Server:

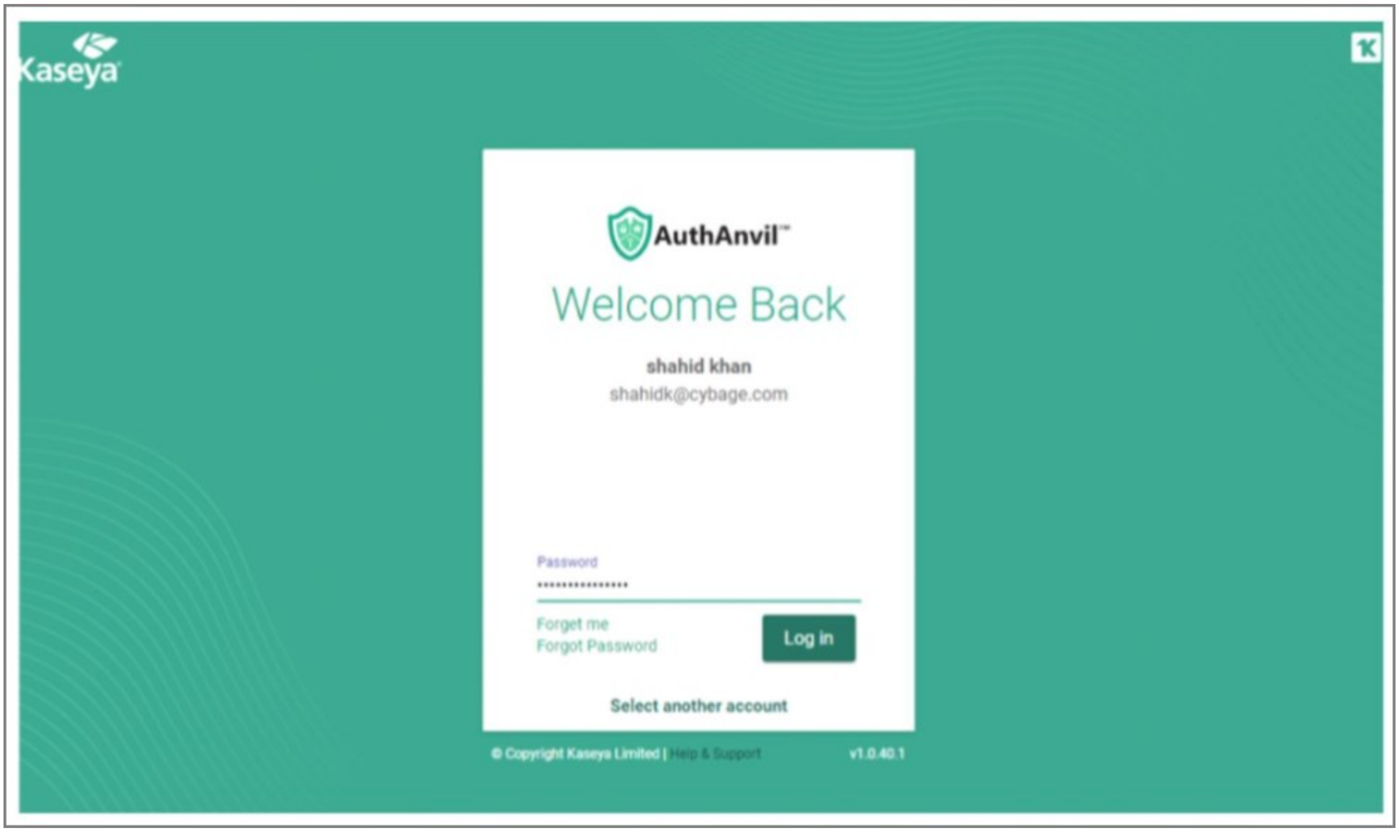
- Add a RADIUS Agent.
- Download the RADIUS Agent Installer.
- Configure RADIUS for the Secret ServerInstance.

Add a RADIUS Agent

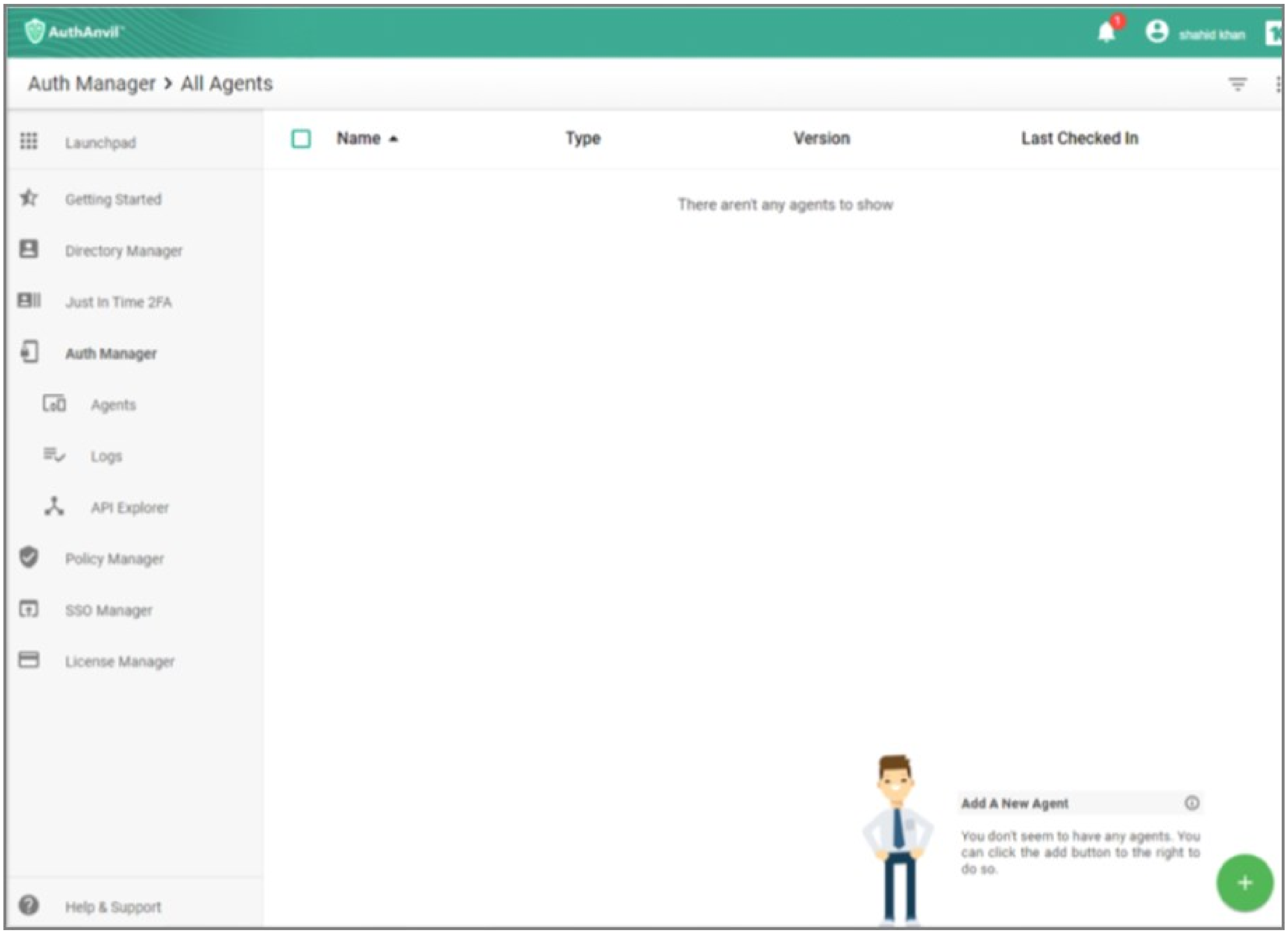
Before you start

- Ensure you have administrative access to your AuthAnvil on Demand tenant.
- Ensure access to a computer that will host the RADIUS Client.
- Ensure access to the desired VPN capable device and are familiar with the configuration.

1. Log into your instance of AuthAnvil on Demand.



2. Click **Auth Manager I plus sign** to add a new Agent.



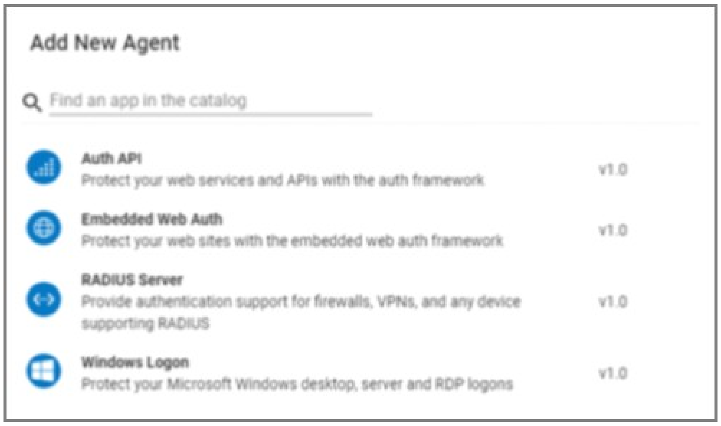
□	Name ▲	Type	Version	Last Checked In

There aren't any agents to show

Add A New Agent

You don't seem to have any agents. You can click the add button to the right to do so.

3. Select **RADIUS Server**.

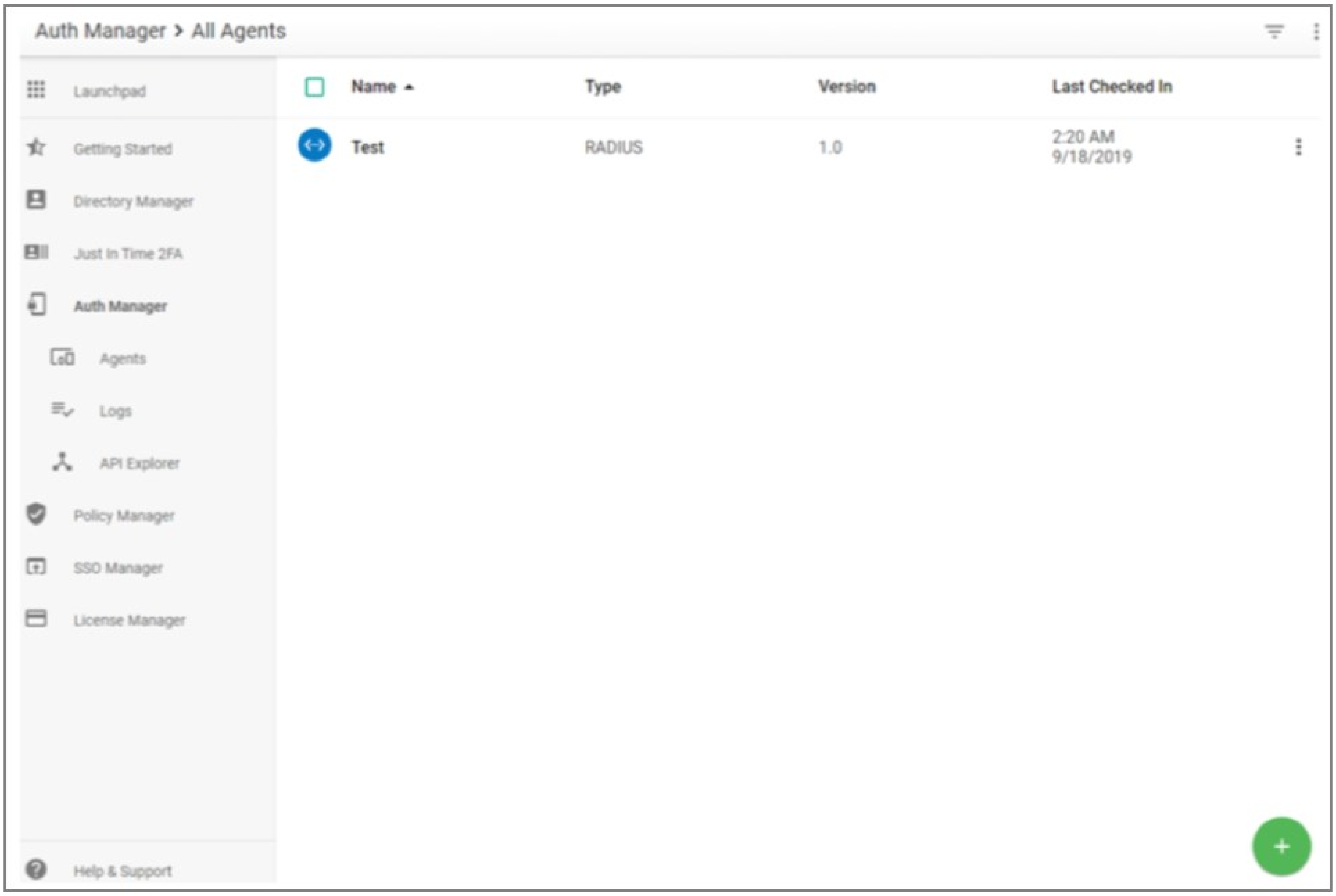


Add New Agent

Auth API Protect your web services and APIs with the auth framework	v1.0
Embedded Web Auth Protect your web sites with the embedded web auth framework	v1.0
RADIUS Server Provide authentication support for firewalls, VPNs, and any device supporting RADIUS	v1.0
Windows Logon Protect your Microsoft Windows desktop, server and RDP logons	v1.0

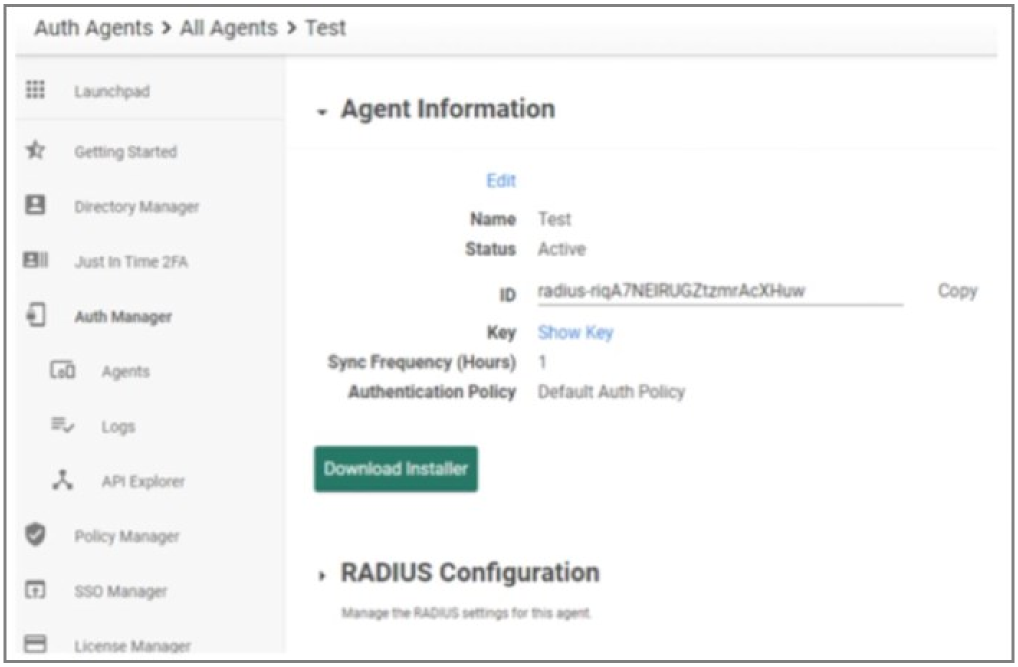
4. Enter a **name for the Agent** to identify its uniqueness from the other agents.

5. Click **Add Agent** in the lower right corner.(The content in the fields is only demonstrative and not to be used for your work.)The All Agents screen will appear with the new Agent listed.



□ Name ▲	Type	Version	Last Checked In
Test	RADIUS	1.0	2:20 AM 9/18/2019

6. Click the **name** to display the Agent Information, and note the following as you will be prompted for this when installing the RADIUS Agent service:
   - ID: The unique ID of the agent.
   - Key: The auto-generated secret value of the agent.



Agent Information

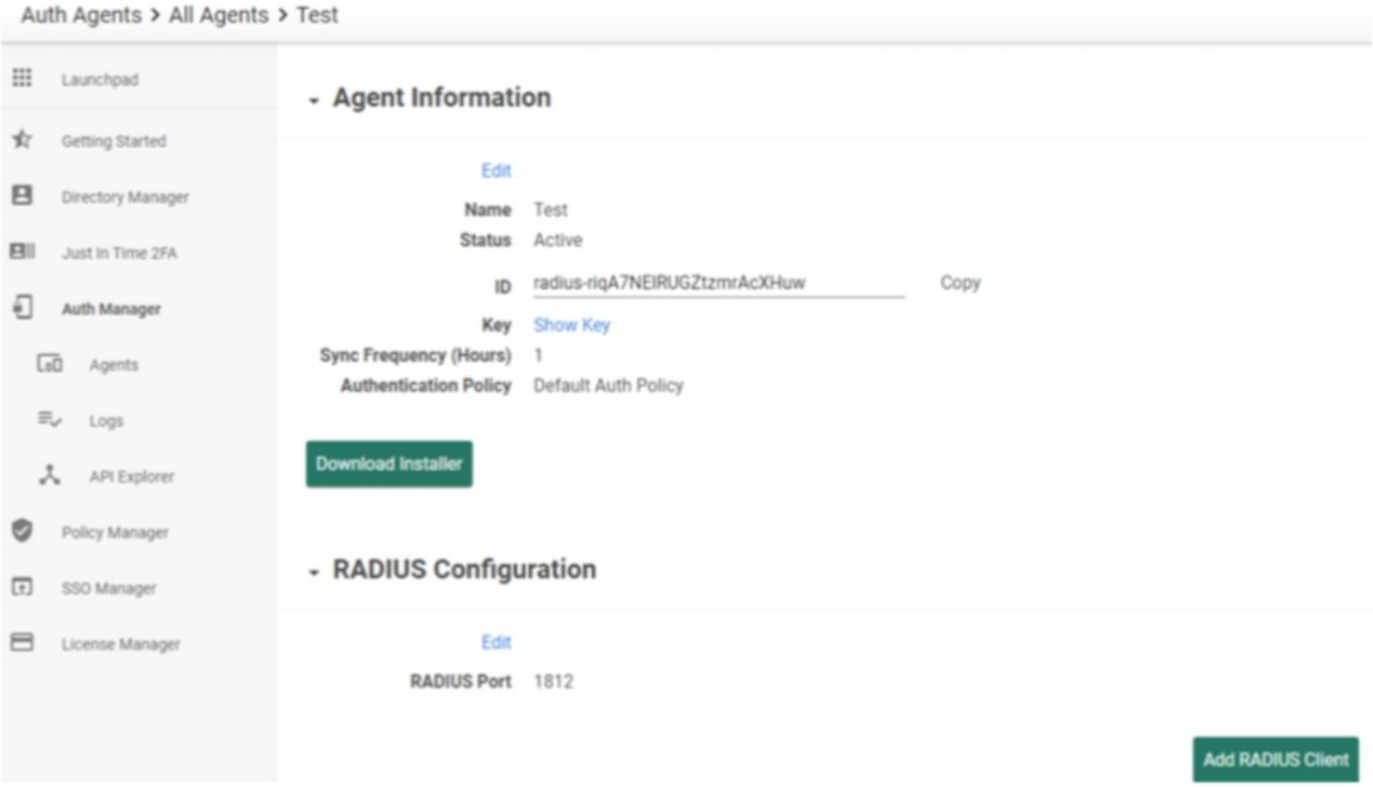
Name Test
Status Active
ID radius-riqA7NEIRUGZtzmrAcXHuw
Key Show Key
Sync Frequency (Hours) 1
Authentication Policy Default Auth Policy

RADIUS Configuration

Manage the RADIUS settings for this agent.

Note: The above information is to be used ONLY at the time of installing the service.

7. Click **RADIUS Configuration** and click **Add RADIUS Client** button or edit the port to use for communication (default port is 1812).



Agent Information

Name Test
Status Active
ID radius-riqA7NEIRUGZtzmrAcXHuw
Key Show Key
Sync Frequency (Hours) 1
Authentication Policy Default Auth Policy

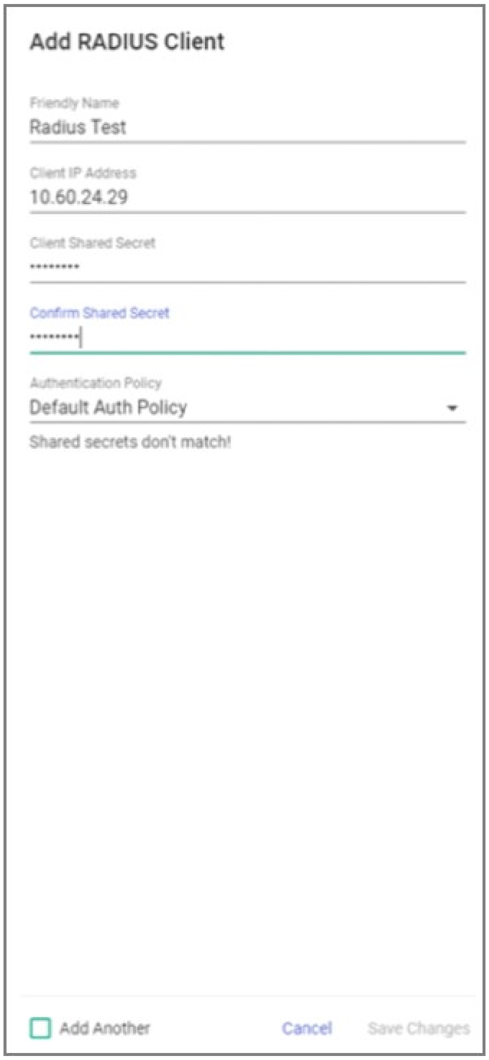
RADIUS Configuration

RADIUS Port 1812

8. In the Add RADIUS Client screen, perform the following:

- Add a **friendly name** for the Client, add the **Client IP address**, and add a **Client Shared Secret**. (The Client Shared Secret key is the password or key setup on each client added in AuthAnvil portal. The same key must be entered in the RADIUS configuration tab in Secret Server under RADIUS Shared Secret).
- Confirm Shared Secret (formerly Confirm Password) for the Client.

Note: This is the shared Secret that will be placed on the forwarding device/router to authenticate the communication.



Add RADIUS Client

Friendly Name: Radius Test

Client IP Address: 10.60.24.29

Client Shared Secret: ••••••••

Confirm Shared Secret: ••••••••

Authentication Policy: Default Auth Policy

Shared secrets don't match!

9. Click **Save Changes**.

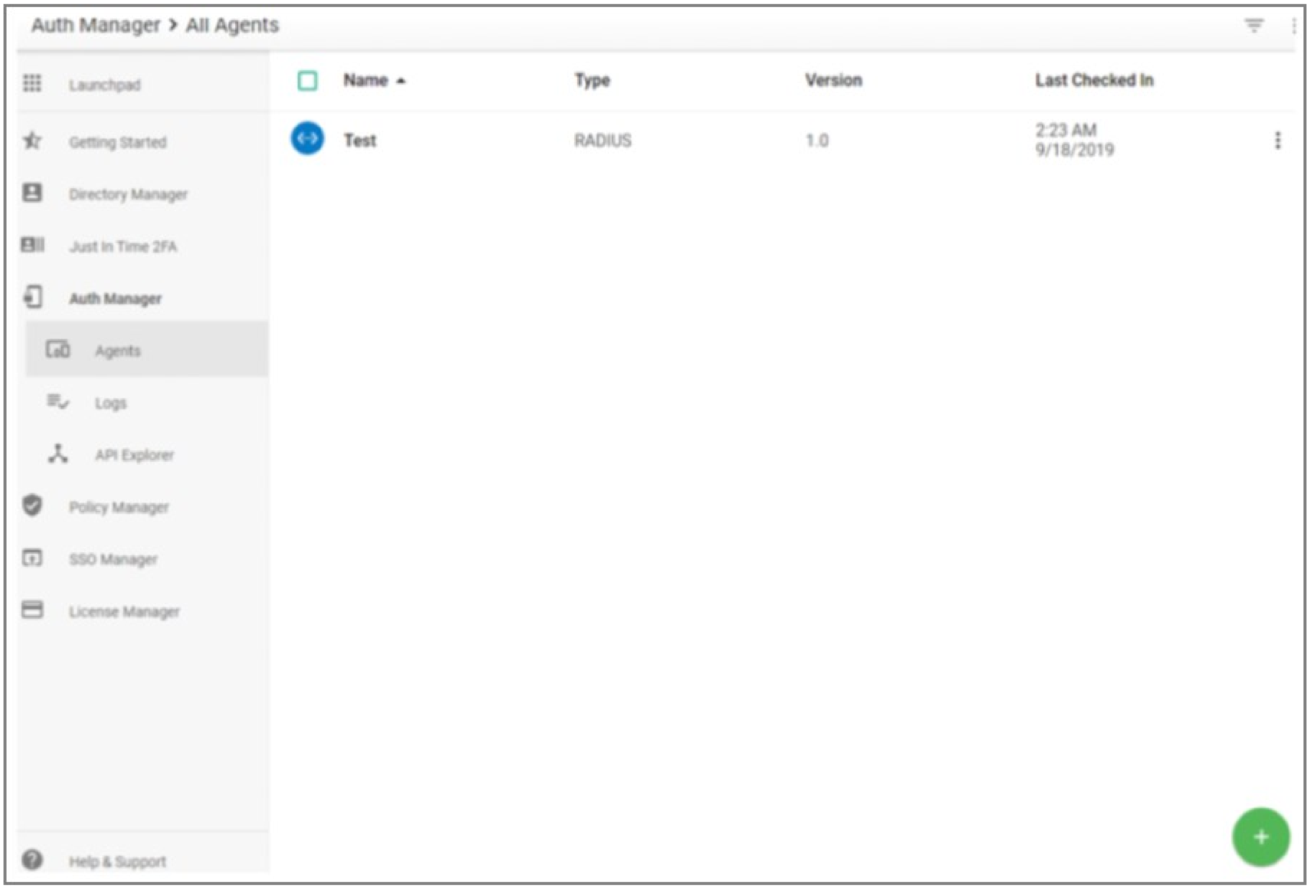
Note: To add more than one client, select the **Add Another** checkbox before selecting the Add RADIUS client button.

Download the RADIUS Agent Installer

Before you start

- Ensure you have administrative access to your AuthAnvil on Demand tenant.
- You should be performing the following on the machine that will host the RADIUS Service.

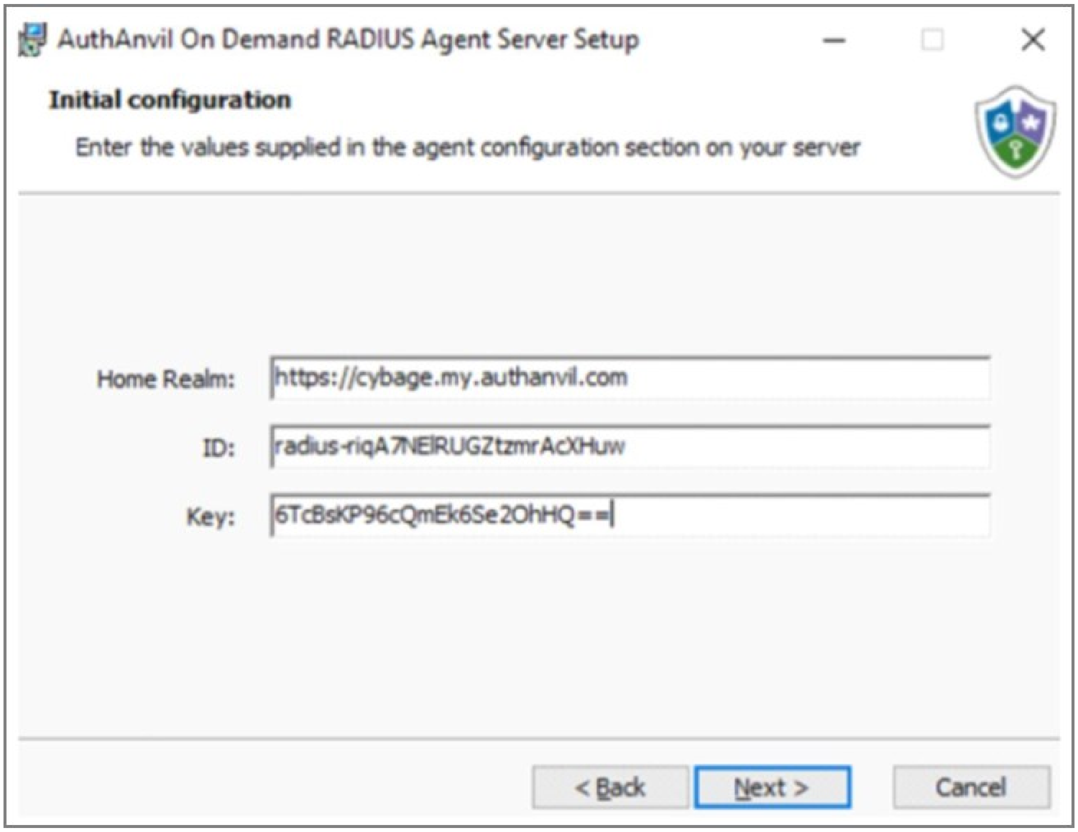
1. On the left navigation tree, select the **Auth Manager | Agents**.



Auth Manager > All Agents

□	Name ▴	Type	Version	Last Checked In
	Test	RADIUS	1.0	2:23 AM 9/18/2019

2. Download the RADIUS installer by performing one of the following options:
   - Click the ellipsis to the right of the agent and select **Download** from the dropdown list.
   - Click the **Agent name** and select the **Download Installer** button that appears.
3. Once downloaded, run through the installation wizard on the machine hosting the RADIUS service.
4. Enter in the **ID** and **Key** that were copied during the RADIUS client creation.
5. Click **Next**.



AuthAnvil On Demand RADIUS Agent Server Setup

Initial configuration

Enter the values supplied in the agent configuration section on your server

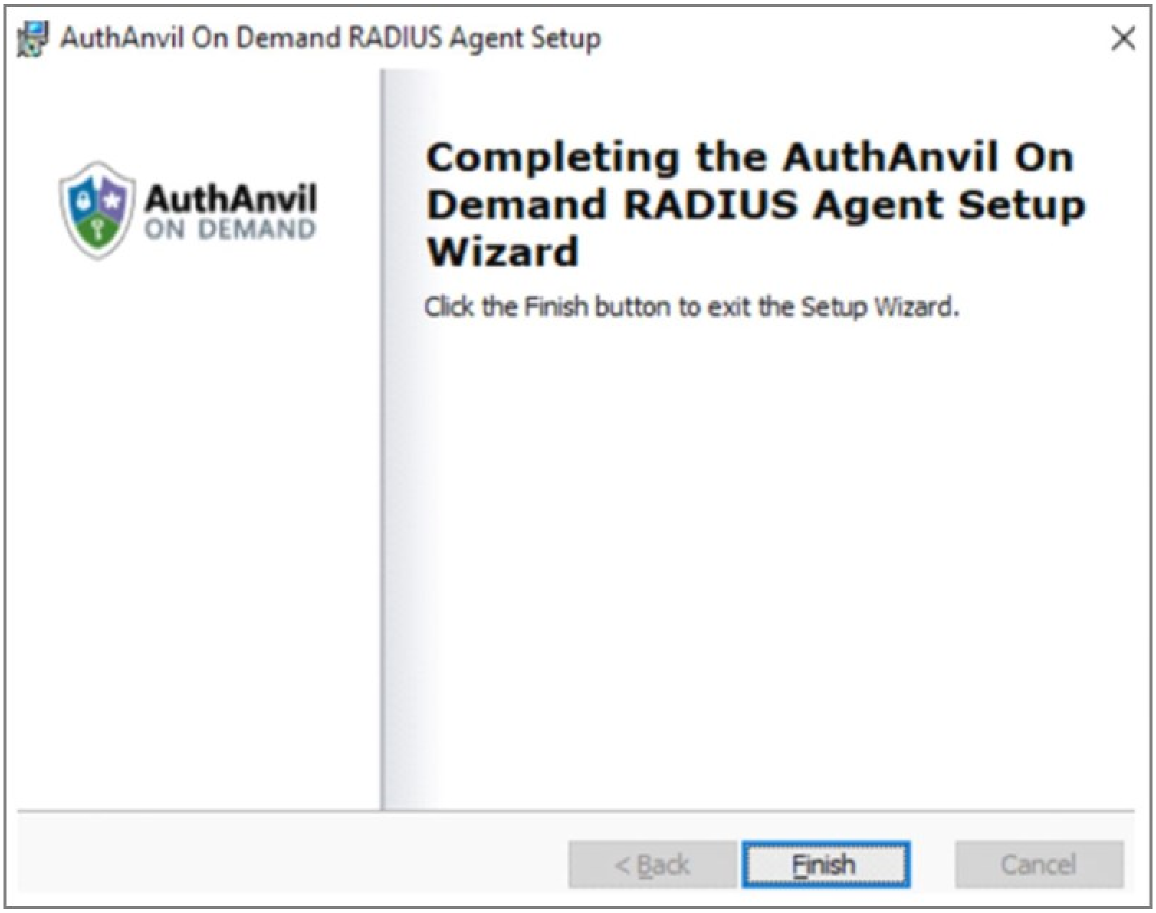
Home Realm: https://cybage.my.authanvil.com

ID: radius-riqA7NElRUGZtzmrAcXHuw

Key: 6TcBsKP96cQmEk6Se2OhHQ==

Note: Home Realm is the user account used to sign in to the AuthAnvil On-Demand portalhttps://yourcompany.my.authanvil.com.

6. Complete the install steps with the Wizard and click **Finish** when complete.



AuthAnvil On Demand RADIUS Agent Setup

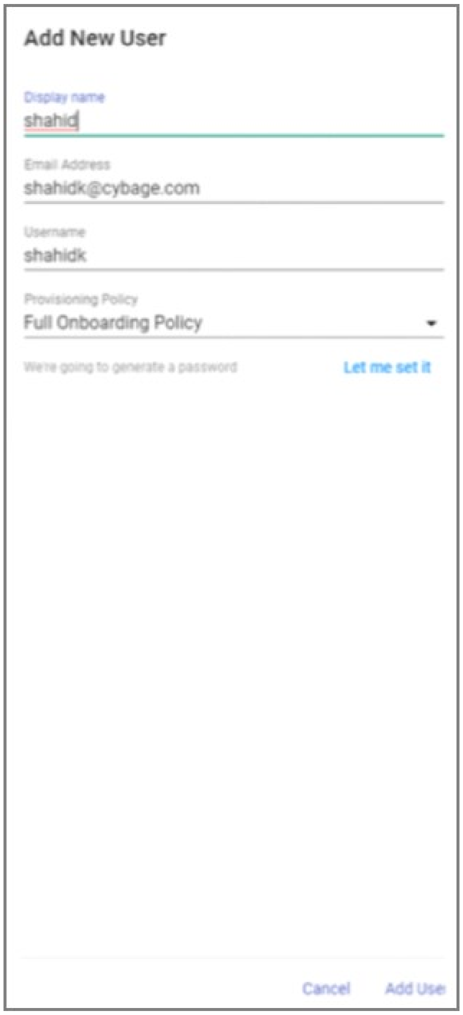
Completing the AuthAnvil On Demand RADIUS Agent Setup Wizard

Click the Finish button to exit the Setup Wizard.

Add a User (Example content)

This example shows you how to add a single new user with AuthAnvil's interactive wizard.

1. Sign in to your AuthAnvil portal.
2. In the left navigation menu, select the **Directory Manager**. The All Users window appears.
3. In the All Users window, click **green plus sign I Add a User**. The Add New User panel appears.
4. Enter the **display name, email address, and username** with your information.
   - Display name – name
   - Email – xxxx@companyname.com
   - Username – username
   - Choose a policy for the new user



Add New User

Display name
shahid

Email Address
shahidk@cybage.com

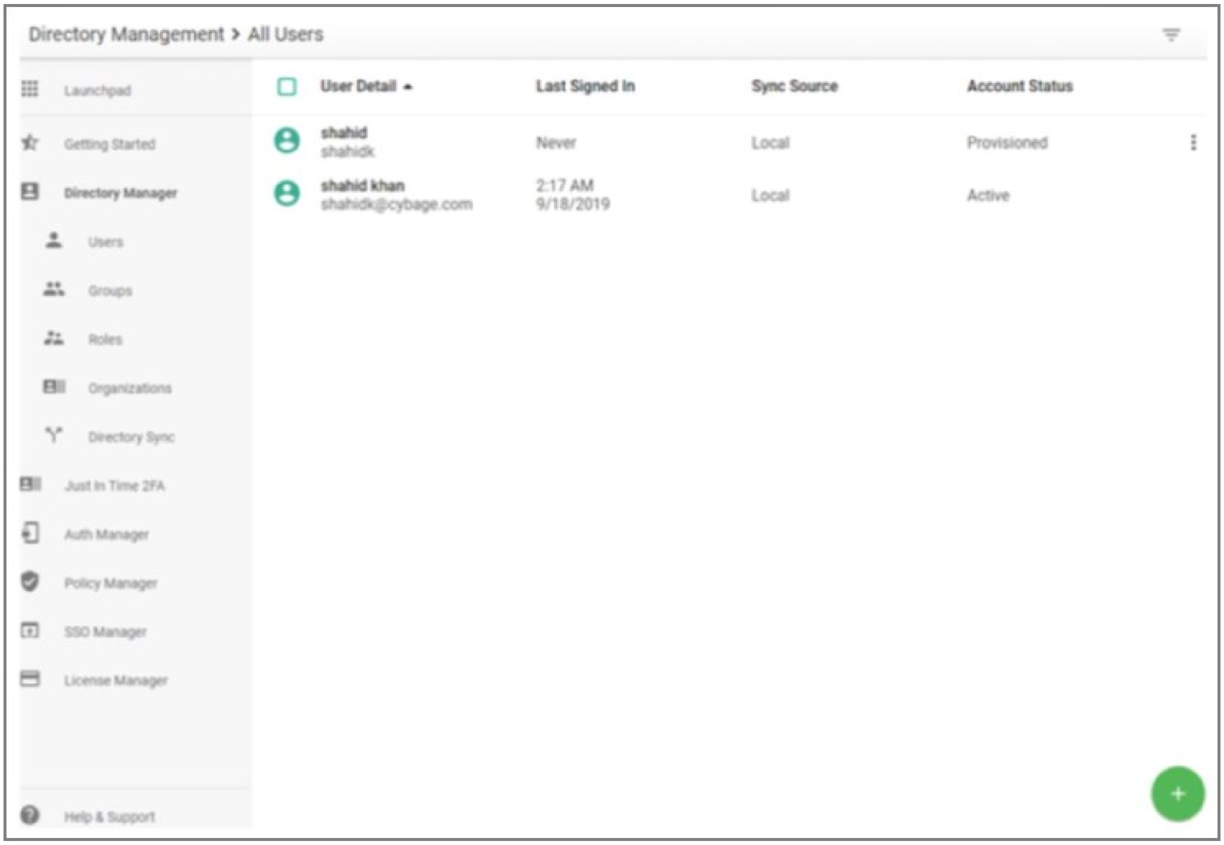
Username
shahidk

Provisioning Policy
Full Onboarding Policy

We're going to generate a password

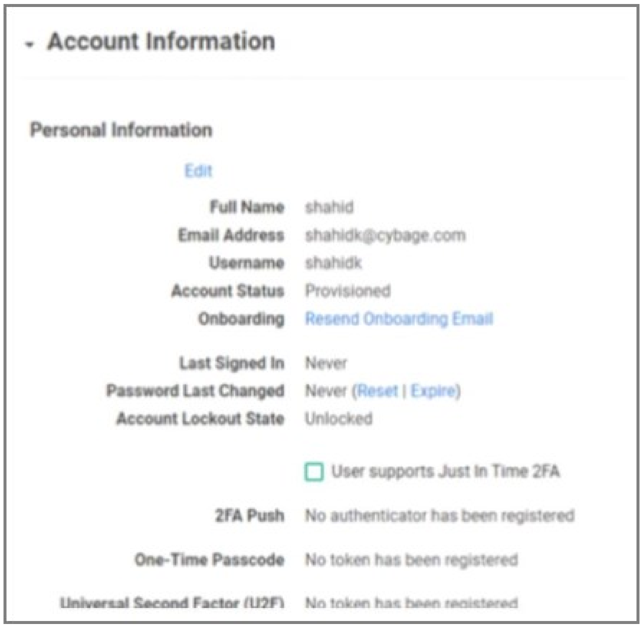
Note: The Default Policy does not send an onboarding email. The Full Onboarding Policy sends an activation email to the specified email address.

5. Click **Add User**. AuthAnvil returns to the All Users view and the new user is now on the list.



User Detail ▲	Last Signed In	Sync Source	Account Status
shahid shahidk	Never	Local	Provisioned
shahid khan shahidk@cybage.com	2:17 AM 9/18/2019	Local	Active

6. Click the **ellipsis** next to the new user. The Account Information window appears.



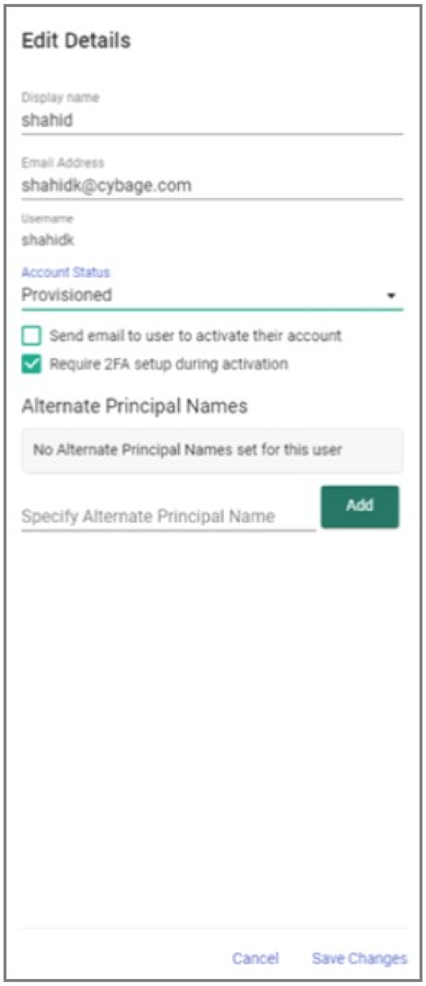
Account Information

Personal Information

Edit

Full Name	shahid
Email Address	shahidk@cybage.com
Username	shahidk
Account Status	Provisioned
Onboarding	Resend Onboarding Email
Last Signed In	Never
Password Last Changed	Never (Reset | Expire)
Account Lockout State	Unlocked
	□ User supports Just In Time 2FA
2FA Push	No authenticator has been registered
One-Time Passcode	No token has been registered
Universal Second Factor (U2F)	No token has been registered

7. Click **Edit**.



Edit Details

Display name
shahid

Email Address
shahidk@cybage.com

Username
shahidk

Account Status
Provisioned

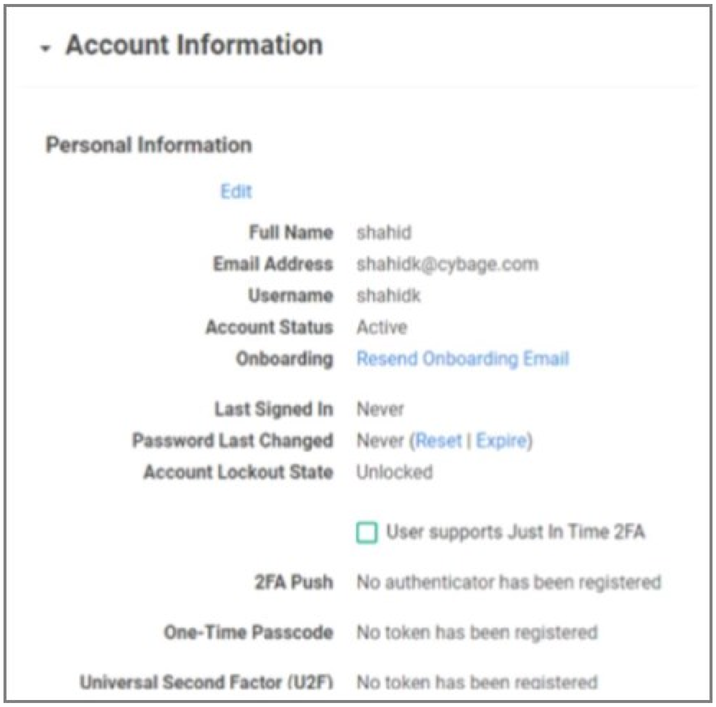
□ Send email to user to activate their account

☑ Require 2FA setup during activation

Alternate Principal Names

No Alternate Principal Names set for this user

8. Click the **Account Status** dropdown menu and change the status to **Active**.
9. Click **Save Changes**.
10. Next to Password Last Changed, click **Reset** to reset the user password.



Personal Information

Full Name	shahid
Email Address	shahidk@cybage.com
Username	shahidk
Account Status	Active
Onboarding	Resend Onboarding Email
Last Signed In	Never
Password Last Changed	Never (Reset | Expire)
Account Lockout State	Unlocked
	□ User supports Just In Time 2FA
2FA Push	No authenticator has been registered
One-Time Passcode	No token has been registered
Universal Second Factor (U2F)	No token has been registered

Configure RADIUS for the Secret Server Instance

Enable RADIUS Two-Factor Authentication in Thycotic Secret Server 10.6 version

Secret Server allows the use of RADIUS two-factor authentication on top of the normal authentication process for additional security needs.

Configure RADIUS for the Secret Server Instance

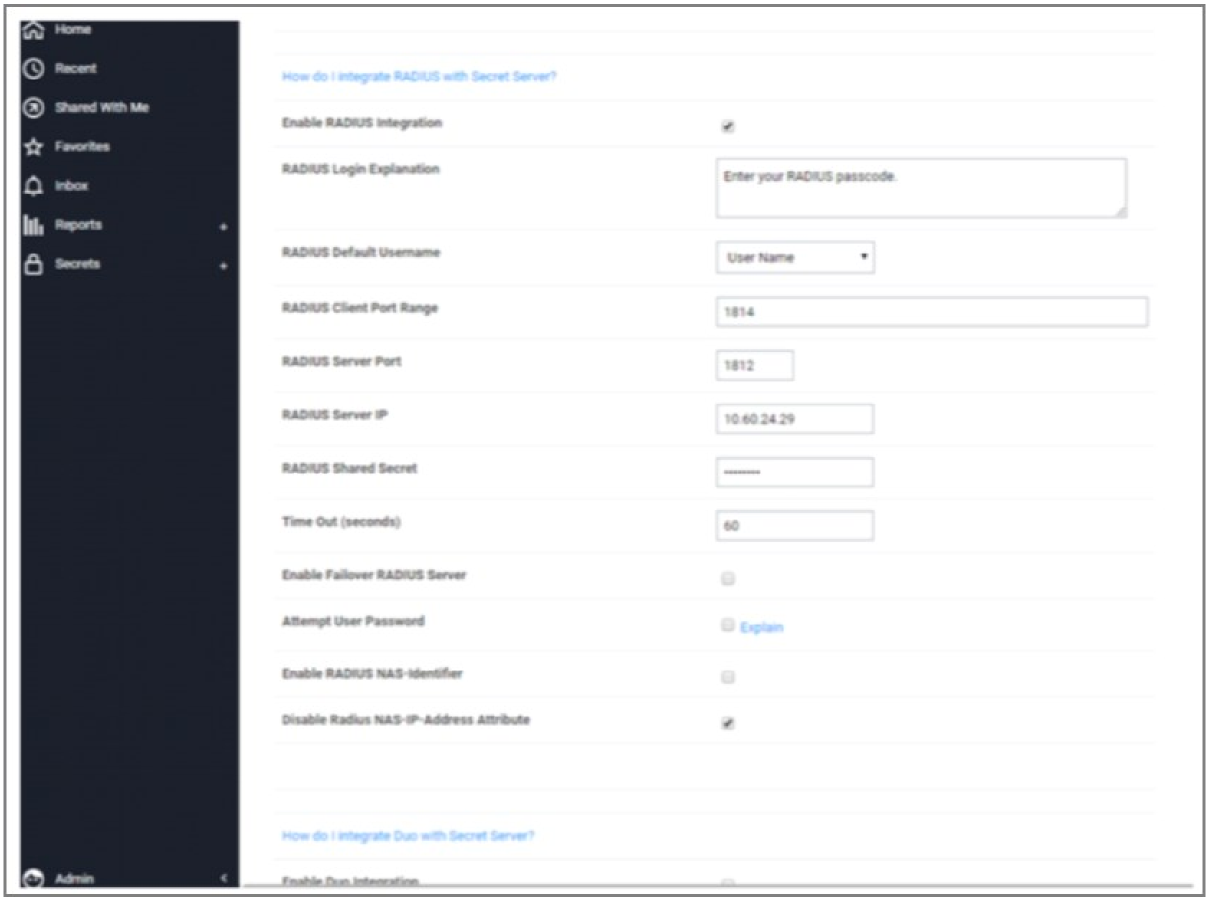
1. Sign in to an account with Administer Configuration and Administer RADIUS permissions.
2. Navigate to **Administration menu l Configuration l Login**.
3. Enable Secret Server with your RADIUS server information by going into edit mode.
   - RADIUS Server IP: IP address to your RADIUS Server.
   - RADIUS Client Port: default 1812.

   Note: If your RADIUS server runs on the same machine as your Secret Server, client and server ports must be different.

   - RADIUS Server Port(default 1812 for RSA and 1812 for AuthAnvil).
   - RADIUS Shared Secret must match the chosen RADIUS shared secret on your RADIUS Server. (Shared Secret is a RADIUS term and not related to any Secret Server secret.)
   - RADIUS Login explanation(custom message or instruction). Defaults to Please enter your RADIUS passcode.
4. Click **Save** after the entries are confirmed.

Test RADIUS settings

1. Click the **Test RADIUS Login** button.



How do I integrate RADIUS with Secret Server?

Enable RADIUS Integration	☑
RADIUS Login Explanation	Enter your RADIUS passcode.
RADIUS Default Username	User Name ▾
RADIUS Client Port Range	1814
RADIUS Server Port	1812
RADIUS Server IP	10.60.24.29
RADIUS Shared Secret	•••••••••
Time Out (seconds)	60
Enable Failover RADIUS Server	□
Attempt User Password	□ Explain
Enable RADIUS NAS-Identifier	□
Disable Radius NAS-IP-Address Attribute	☑

How do I integrate Duo with Secret Server?

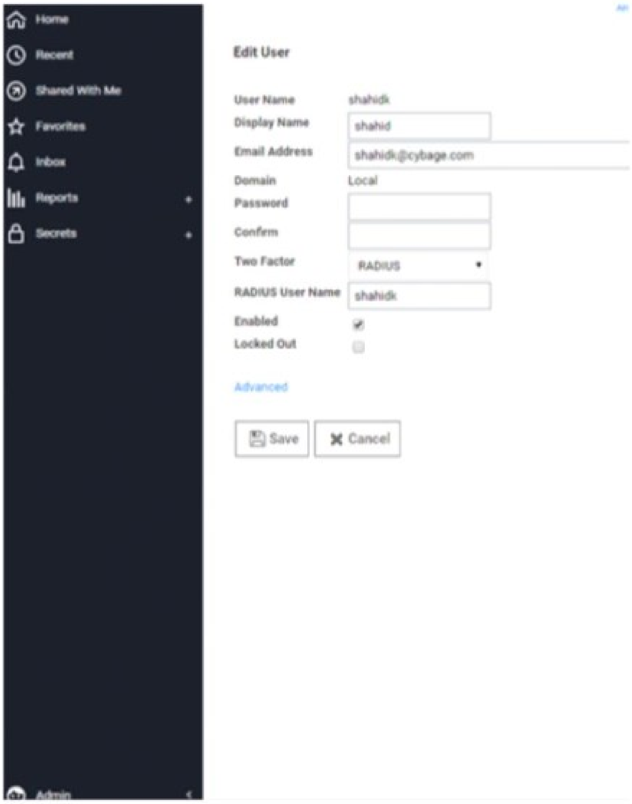
Enable Duo Integration	□

Note: After enabling RADIUS in your Secret Server instance, you must also enable **RADIUS two-factor authentication** for each user. You can enable it on a per-user basis.

2. Sign in to an account with Administer Configuration and Administer RADIUS permissions.
3. Navigate to **Administration | Users | Username** of user to enable.
4. Click the **Edit** button and check the **RADIUS Two Factor Authentication** checkbox.
5. Enter the **RADIUS username** in the text field.

   Note: Secret Server defaults this value to its username. If you wish to use this default name, it must match the username on the RADIUS server.

6. Review the settings and click **Save**.
7. Repeat 3-5 for each user.

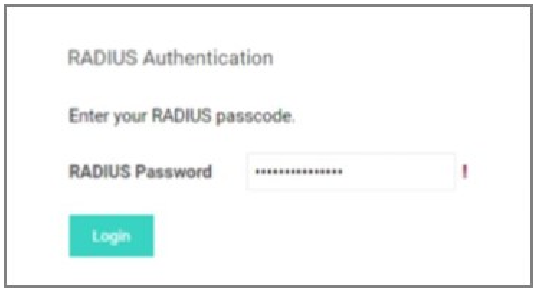


Edit User

User Name: shahidk
Display Name: shahid
Email Address: shahidk@cybage.com
Domain: Local
Password:
Confirm:
Two Factor: RADIUS
RADIUS User Name: shahidk
Enabled: ☑
Locked Out: □

Validate Authentication through RADIUS Server

1. Sign into **Secret Server**. The RADIUS Authentication screen appears.



RADIUS Authentication

Enter your RADIUS passcode.

RADIUS Password: ••••••••••••••• !

2. Enter your **RADIUS user password** created in AuthAnvil On-Demand.

You should be successfully logged into Secret Server using two-factor authentication through RADIUS.

Microsoft

Below are the following integrations that are available with Microsoft:

- Azure Sentinel

Connect your Thycotic Secret Server to Azure Sentinel

This article explains how to connect your Thycotic Secret Server appliance to Azure Sentinel. The Thycotic Secret Server data connector allows you to easily connect your Thycotic Secret Server logs with Azure Sentinel, to view dashboards, create custom alerts, and improve investigation. Integration between Thycotic and Azure Sentinel makes use of the CEF Data Connector to properly parse and display Secret Server Syslog messages.

> **NOTE**: Data will be stored in the geographic location of the workspace on which you are running Azure Sentinel.

Configure Logs and Connect Thycotic Secret Server to the Syslog Agent

Configure Thycotic Secret Server to forward Syslog messages in CEF format to your Azure workspace via the Syslog agent. If you don't have such a log forwarding server, see these instructions to get one up and running.

1. In the Azure Sentinel portal:
   1. Click Data connectors.
   2. Select Thycotic Secret Server.
   3. Open the connector page.
2. Follow the configure Secret Server instructions to configure sending syslog data to the log forwarding server.
3. Validate your connection and verify data ingestion using these instructions. It may take up to 20 minutes until your logs start to appear in Log Analytics.

Find Your Data

After a successful connection is established, the data appears in Logs, under the Azure Sentinel section, in the **CommonSecurityLog** table.

To query the Thycotic logs in Log Analytics, enter **CommonSecurityLog** at the top of the query window.

Next Steps

In this document, you learned how to connect Thycotic Secret Server to Azure Sentinel. To learn more about Azure Sentinel, see the following articles:

- Learn how to get visibility into your data, and potential threats.
- Get started detecting threats with Azure Sentinel.
- Use workbooks to monitor your data.

Okta

Below are the following integrations that are available with Okta:

- SAML
- SCIM

Okta SCIM Integration

This document explains how to connect Okta with the SCIM Connector application for Thycotic Secret Server (SS). This includes what Okta configuration settings are required for connection, known issues, and other related information.

The following topic are available:

- Configuring an Okta Endpoint to Work with the SCIM Connector
- Okta Provisioning

Resources

SCIM Connectors

For more information on the SCIM Connector, please see the below links:

- Secret Server SCIM Connector – Installation
- Secret Server SCIM Connector – Getting Started

OKTA Documentation

- SCIM: Provisioning with Okta's Lifecycle Management
- Okta Basic User Schema
- Okta Universal Director

Configuring an Okta Endpoint to Work with the SCIM Connector

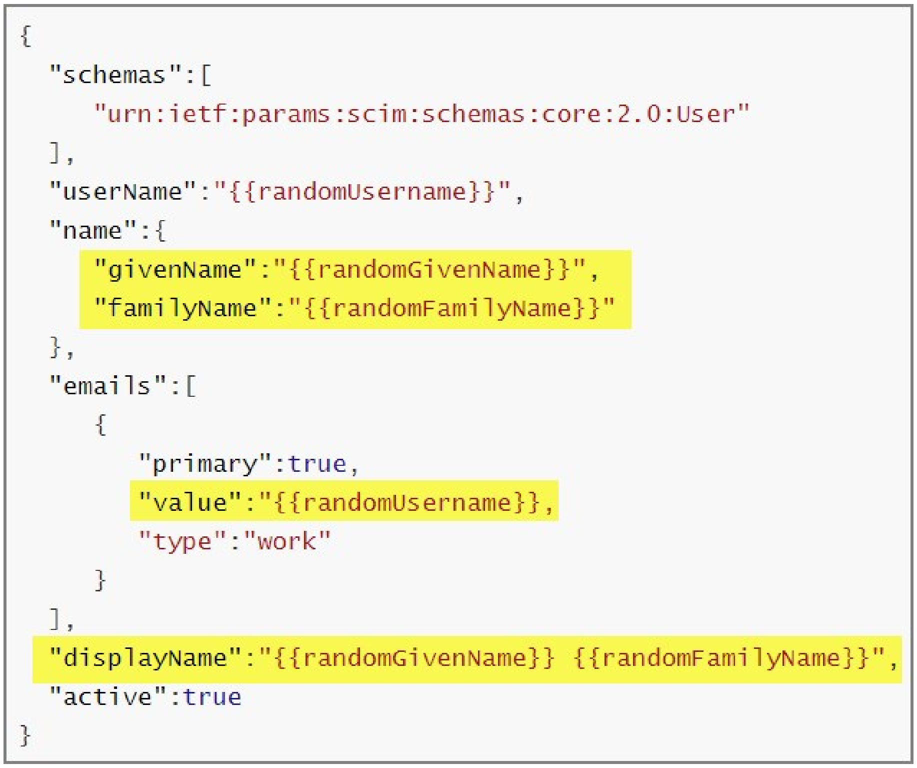
The steps in this topic are required to configure Okta for use as a SCIM EndPoint for the SS SCIM Connector application. They are in addition to making a SCIM Endpoint connection within the SCIM Connector application itself.

By default, there are two fields in OKTA that are marked as mandatory and used to identify users; these are the First Name and Last Name fields (please see the Okta Universal Directory link below for details).

However, the SS SCIM Connector application uses the primary email value to identify users instead, so if the SCIM connector uses the SCIM standard to request user values, it passes blank values for these two fields, resulting in data request or importation failure. To allow OKTA and the SCIM Connector to communicate successfully, you must change the status of these two fields (First Name and Last Name) from mandatory to optional.

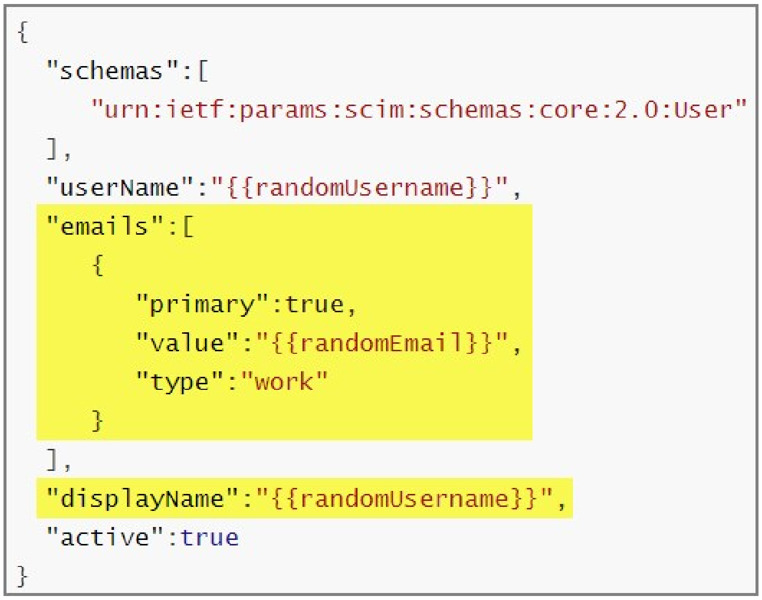
Once the fields are made optional, some additional changes are required in the code for the JSON body of the POST call (the yellow-highlighting shows the modified text):

Original JSON body for the POST call:



```
{
  "schemas":[
     "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName":"{{randomUsername}}",
  "name":{
     "givenName":"{{randomGivenName}}",
     "familyName":"{{randomFamilyName}}"
  },
  "emails":[
     {
        "primary":true,
        "value":"{{randomUsername}},
        "type":"work"
     }
  ],
  "displayName":"{{randomGivenName}} {{randomFamilyName}}",
  "active":true
}
```

Modified JSON body for the POST call:

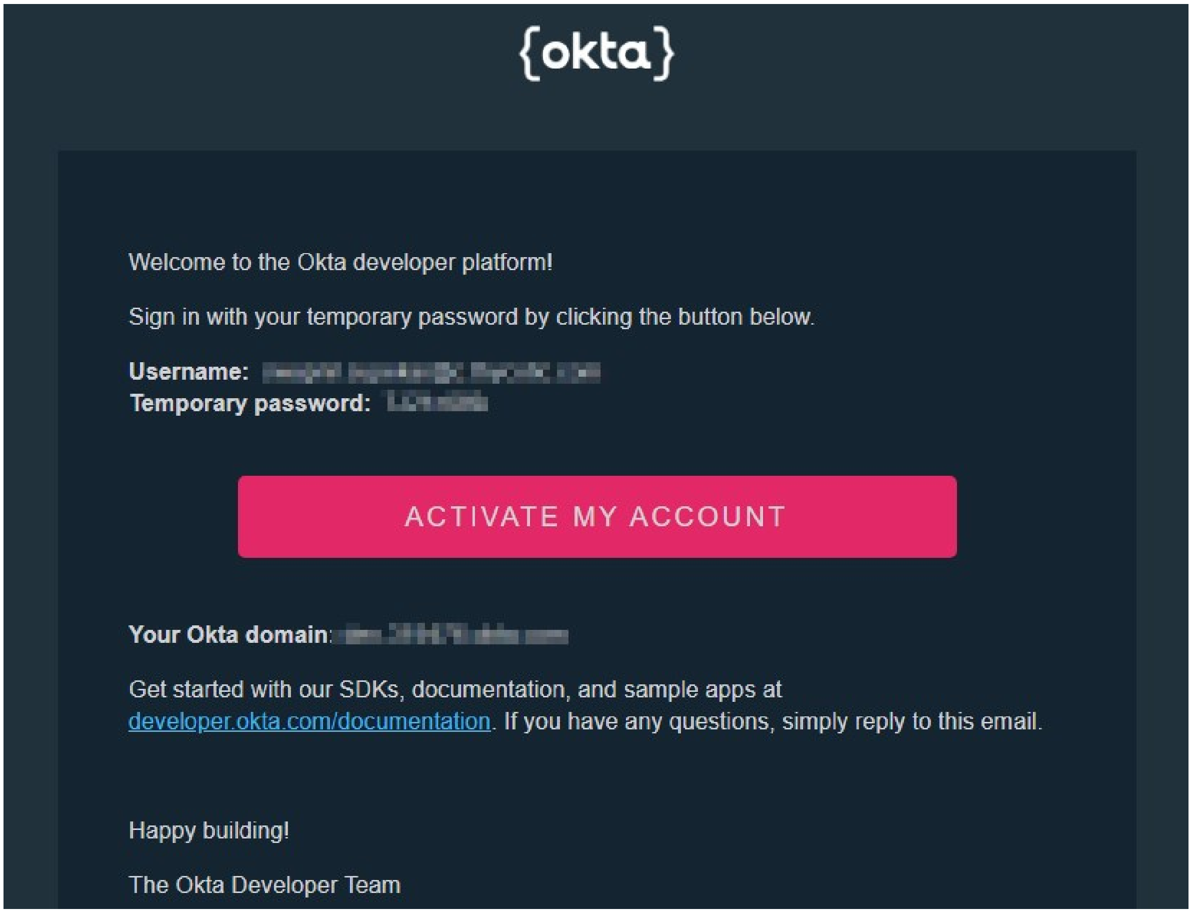


```
{
  "schemas":[
     "urn:ietf:params:scim:schemas:core:2.0:User"
  ],
  "userName":"{{randomUsername}}",
  "emails":[
     {
        "primary":true,
        "value":"{{randomEmail}}",
        "type":"work"
     }
  ],
  "displayName":"{{randomUsername}}",
  "active":true
}
```

Okta Provisioning

Connect your SCIM service with a Okta integration

1. Begin by signing up for a developer account using URL: https://developer.okta.com/signup
2. After creating the account you will receive an email, open the link to your developer account.



{okta}

Welcome to the Okta developer platform!

Sign in with your temporary password by clicking the button below.

Username: [redacted]
Temporary password: [redacted]

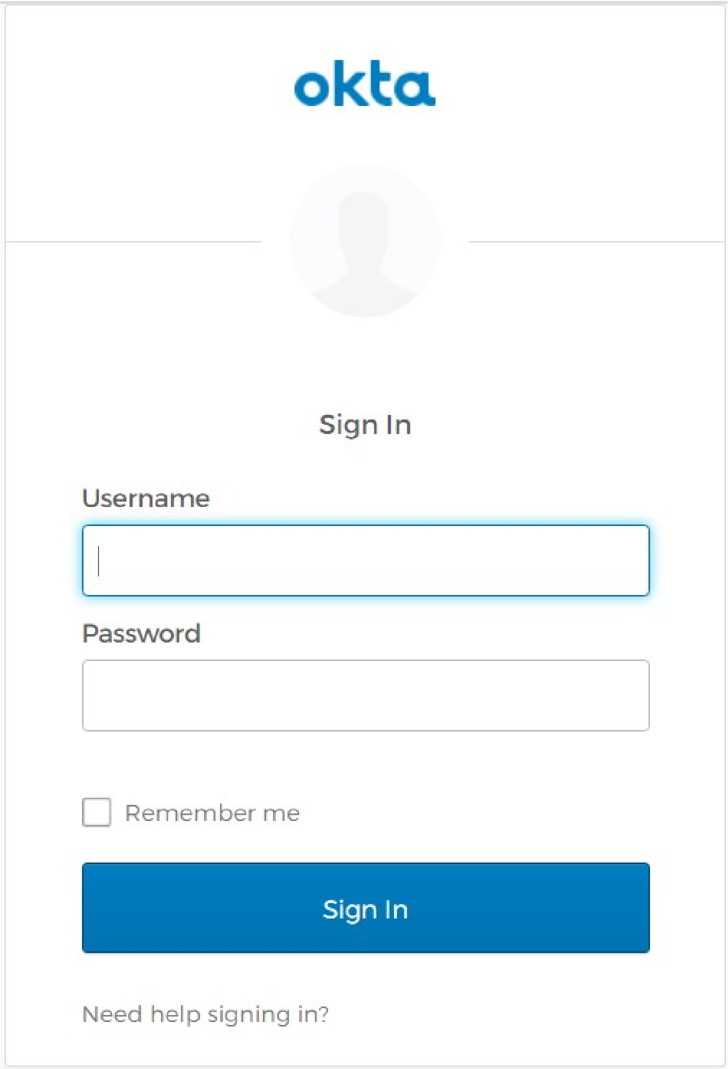
Your Okta domain: [redacted]

Get started with our SDKs, documentation, and sample apps at developer.okta.com/documentation. If you have any questions, simply reply to this email.

Happy building!

The Okta Developer Team

3. Enter the **username** and **password**.

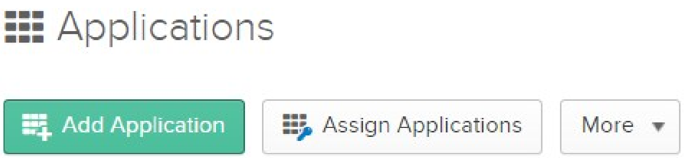


4. Navigate to the Admin Console in your Okta org by clicking **Admin**.

5. If you are in the Developer Console, click **Developer Console** and then **Classic UI** to switch over to the Admin Console in your Okta org.

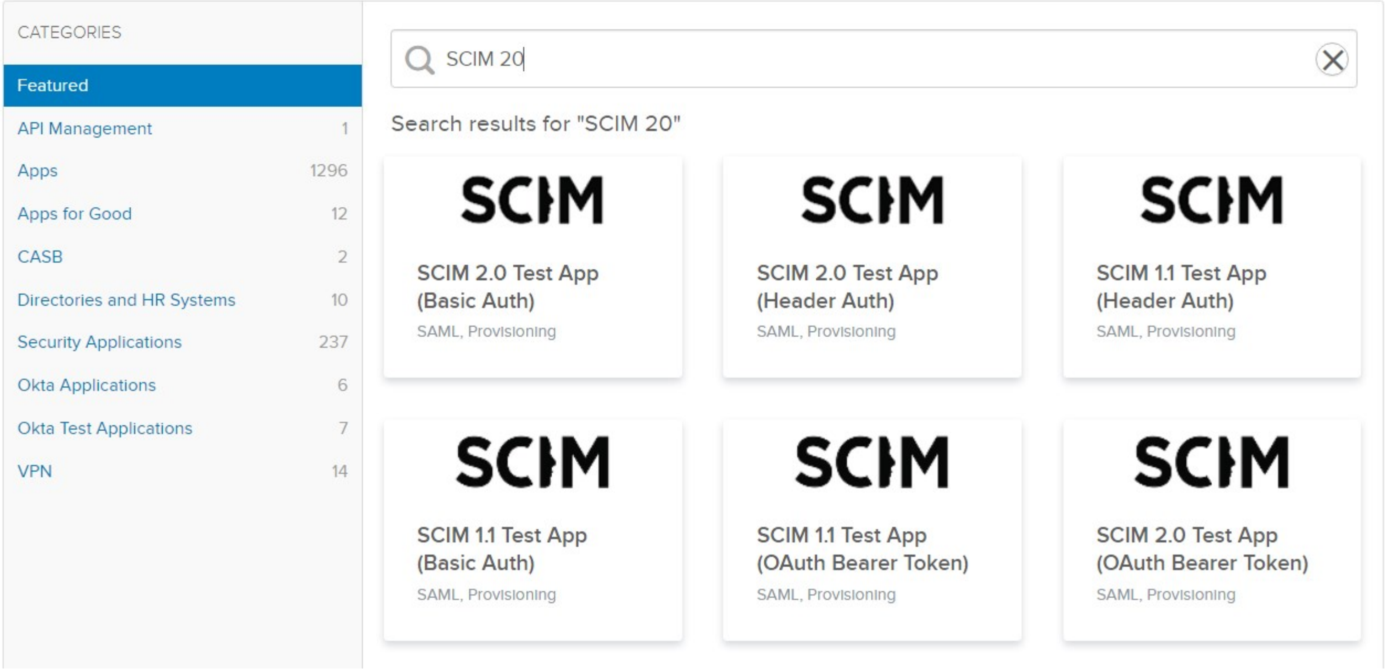
6. Click **Applications**.

7. Click **Add Application**.



8. Search for **SCIM 2.0**.

   Note: Three different SCIM template applications, will be displayed each of the three authentication methods that you can use to connect to your SCIM implementation (Basic Auth, Header Auth, or OAuth Bearer Token).

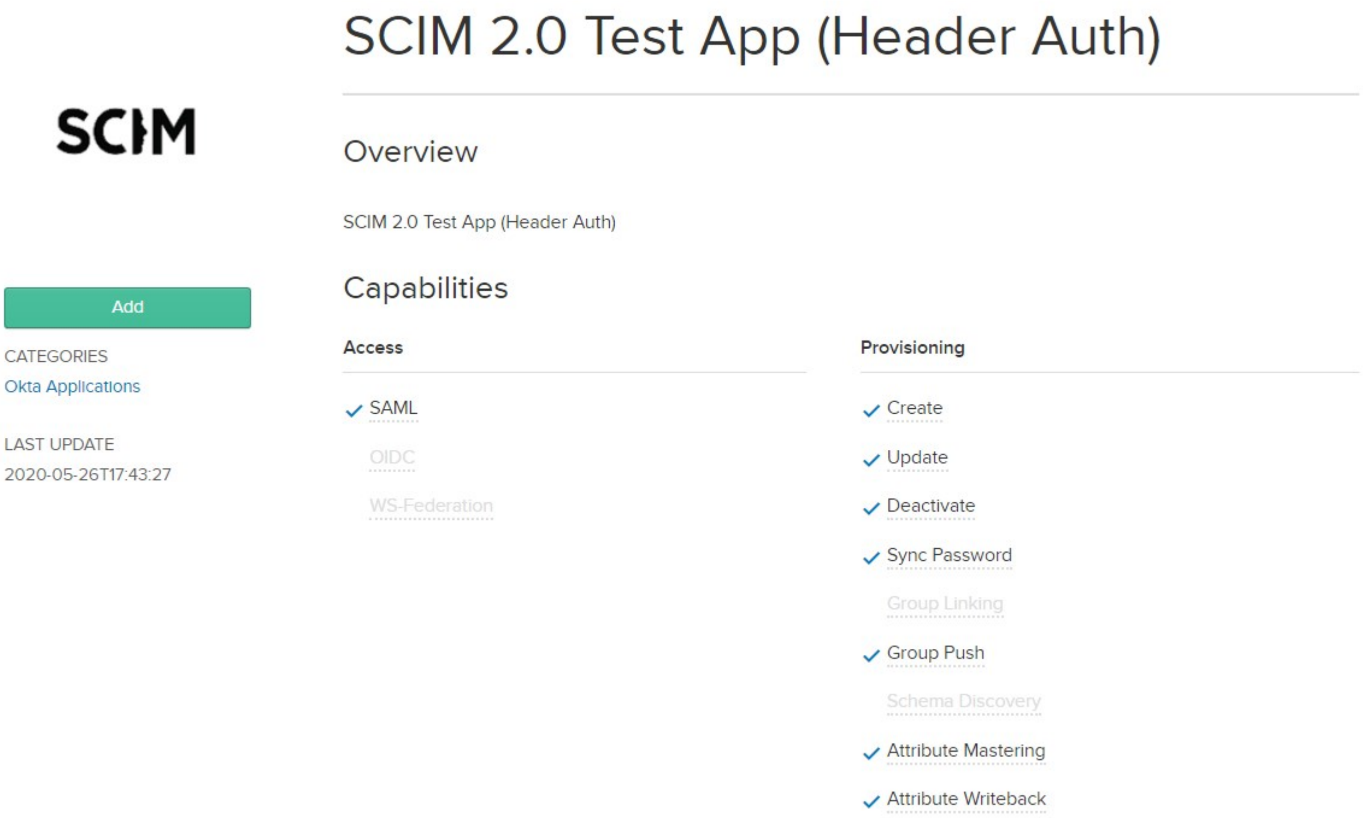


SCIM 20

Search results for "SCIM 20"

CATEGORIES
Featured
API Management 1
Apps 1296
Apps for Good 12
CASB 2
Directories and HR Systems 10
Security Applications 237
Okta Applications 6
Okta Test Applications 7
VPN 14
SCIM
SCIM 2.0 Test App (Basic Auth)
SAML, Provisioning
SCIM
SCIM 2.0 Test App (Header Auth)
SAML, Provisioning
SCIM
SCIM 1.1 Test App (Header Auth)
SAML, Provisioning
SCIM
SCIM 1.1 Test App (Basic Auth)
SAML, Provisioning
SCIM
SCIM 1.1 Test App (OAuth Bearer Token)
SAML, Provisioning
SCIM
SCIM 2.0 Test App (OAuth Bearer Token)
SAML, Provisioning

9. Select **SCIM 2.0 Test App** (Header Auth).
10. Click **Add** on the template to use.



SCIM 2.0 Test App (Header Auth)

Overview

SCIM 2.0 Test App (Header Auth)

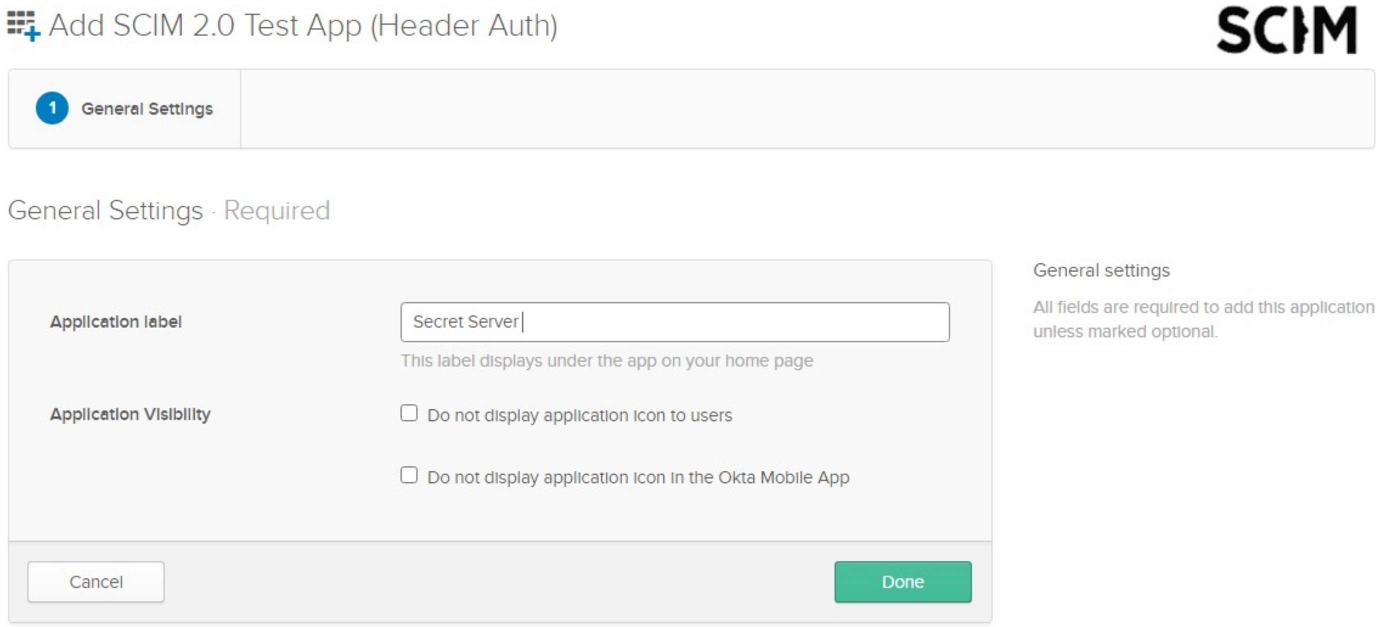
Capabilities

Access	Provisioning
✓ SAML	✓ Create
OIDC	✓ Update
WS-Federation	✓ Deactivate
	✓ Sync Password
	Group Linking
	✓ Group Push
	Schema Discovery
	✓ Attribute Mastering
	✓ Attribute Writeback

CATEGORIES
Okta Applications

LAST UPDATE
2020-05-26T17:43:27

11. On the **General Settings** page, give your integration a descriptive name and click **Done**.



Add SCIM 2.0 Test App (Header Auth)

General Settings · Required

Application label: Secret Server

This label displays under the app on your home page

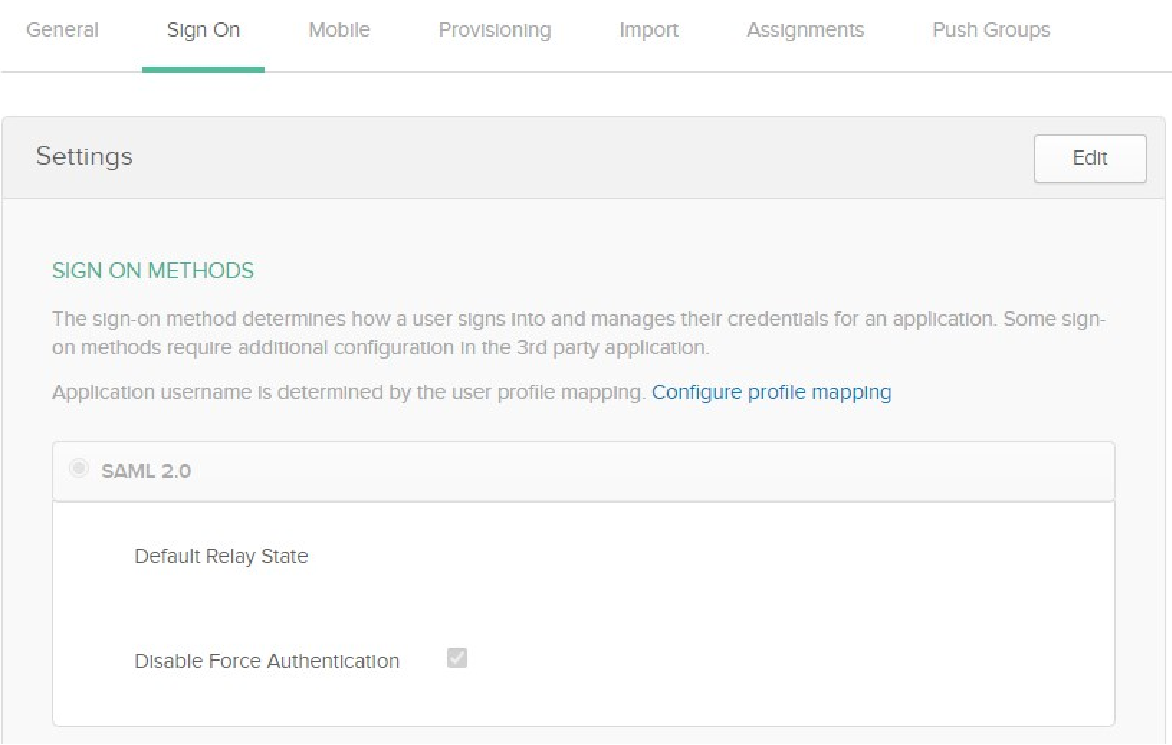
Application Visibility: □ Do not display application icon to users

□ Do not display application icon in the Okta Mobile App

General settings

All fields are required to add this application unless marked optional.

12. On the **Sign-On Options** page, verify SAML 2.0 is selected.



SIGN ON METHODS

The sign-on method determines how a user signs into and manages their credentials for an application. Some sign-on methods require additional configuration in the 3rd party application.

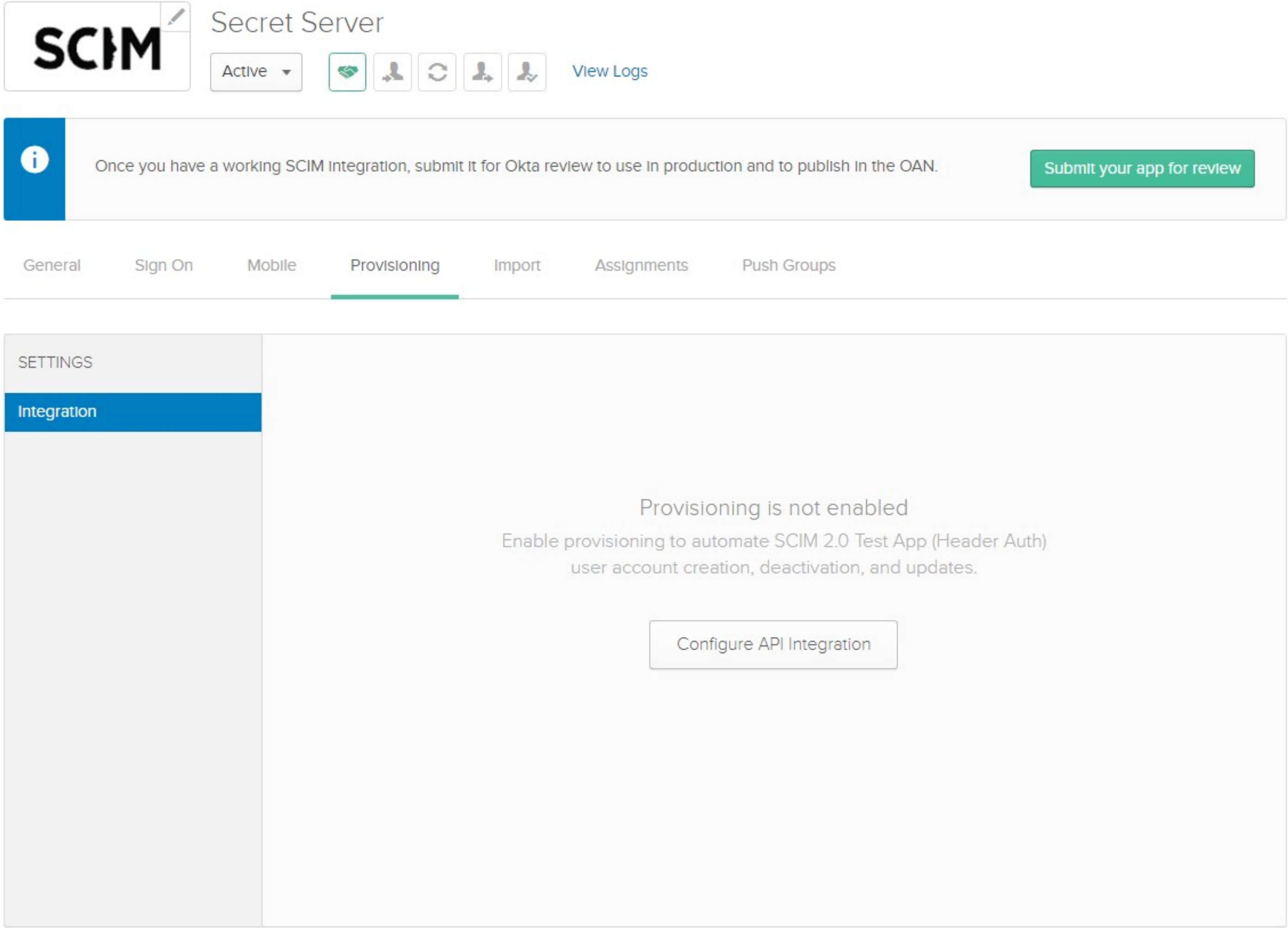
Application username is determined by the user profile mapping. Configure profile mapping

● SAML 2.0

Default Relay State

Disable Force Authentication ☑

13. Click the **Provisioning** tab, and in the main panel, click **Configure API Integration**.

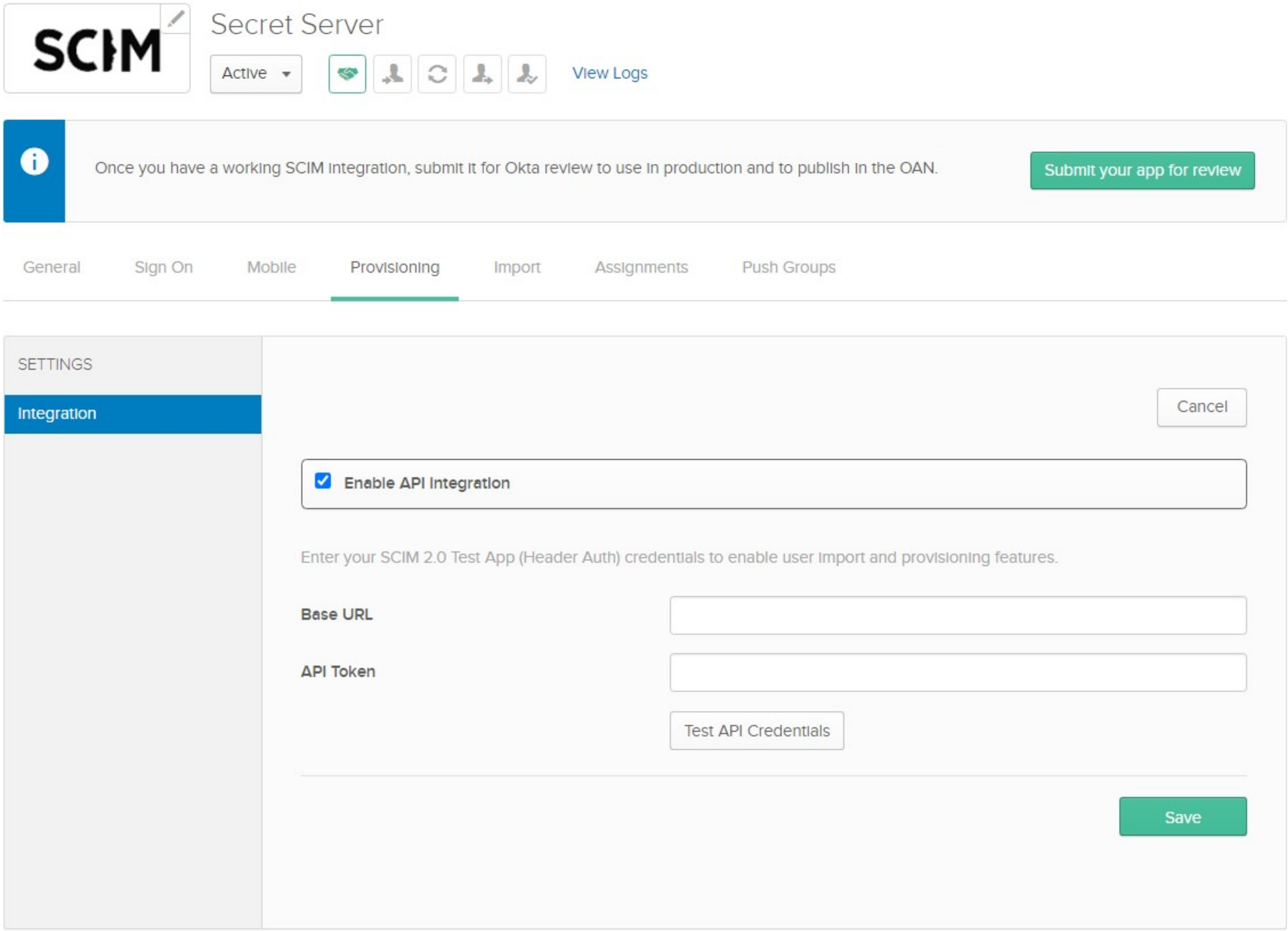


Once you have a working SCIM integration, submit it for Okta review to use in production and to publish in the OAN.

Provisioning is not enabled

Enable provisioning to automate SCIM 2.0 Test App (Header Auth) user account creation, deactivation, and updates.

14. Select the **Enable API Integration** check box.



Once you have a working SCIM integration, submit it for Okta review to use in production and to publish in the OAN.

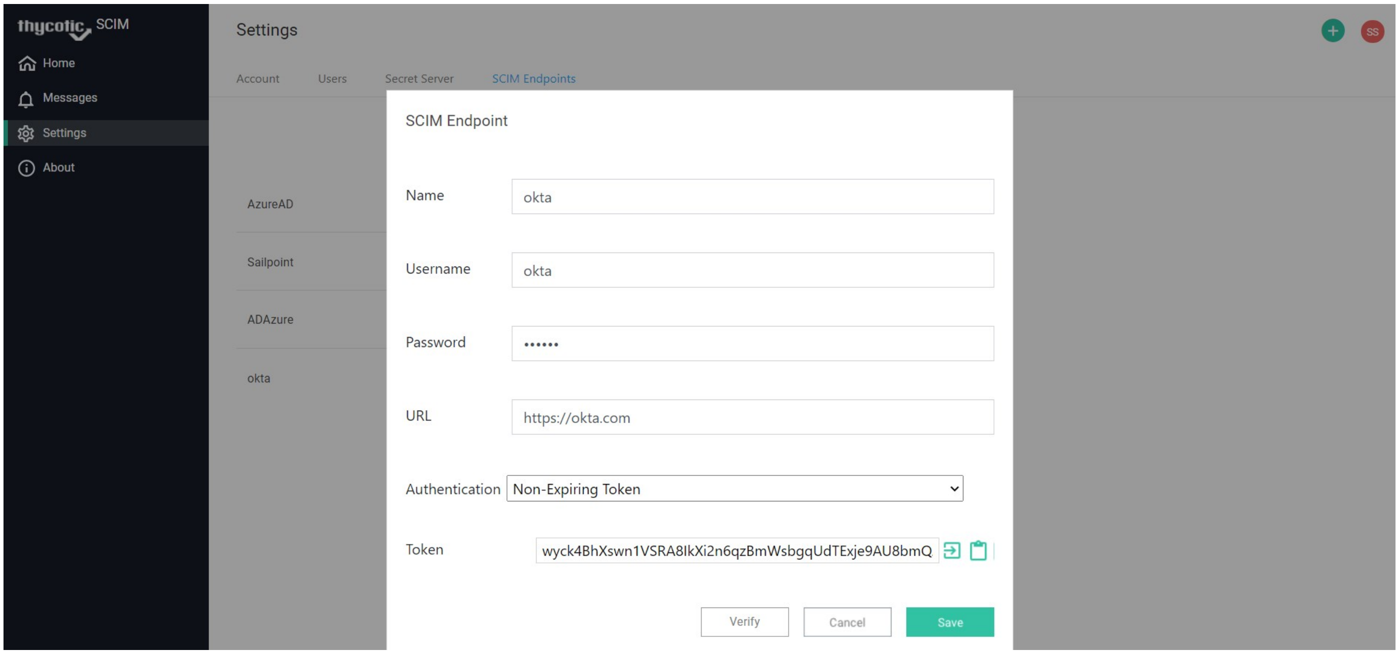
☑ Enable API Integration

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

Base URL

API Token

15. Enter the base URL and Token from the Thycotic SCIM Connector HTTP Header. To authenticate using HTTP Header, provide a bearer token to access your SCIM implementation.



SCIM Endpoint

Name: okta

Username: okta

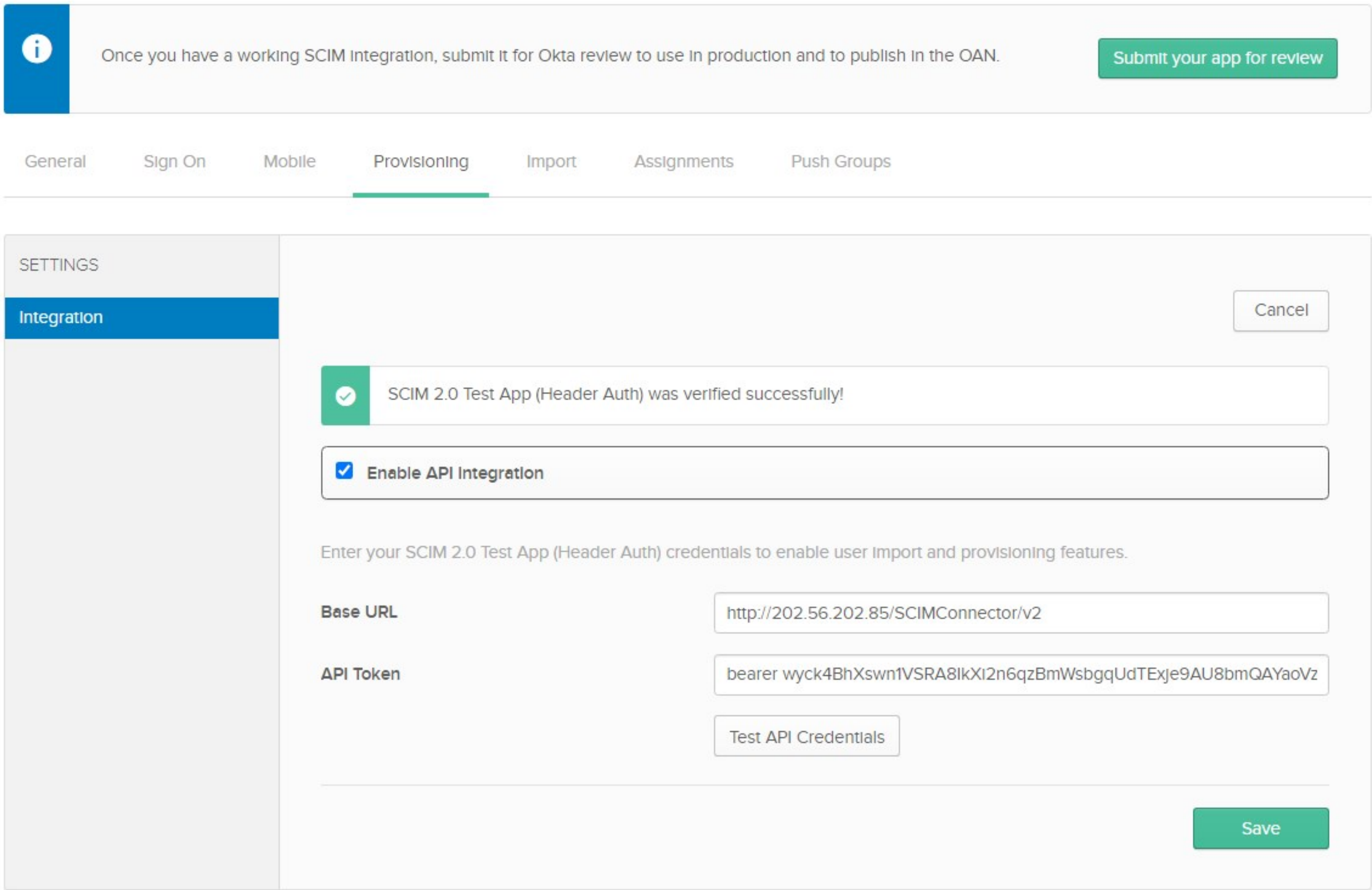
Password: ••••••

URL: https://okta.com

Authentication: Non-Expiring Token

Token: wyck4BhXswn1VSRA8lkXi2n6qzBmWsbgqUdTExje9AU8bmQ

16. Click **Test API Credentials** to test whether the Okta integration can connect to your SCIM API.


Once you have a working SCIM integration, submit it for Okta review to use in production and to publish in the OAN.

SCIM 2.0 Test App (Header Auth) was verified successfully!

☑ Enable API Integration

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

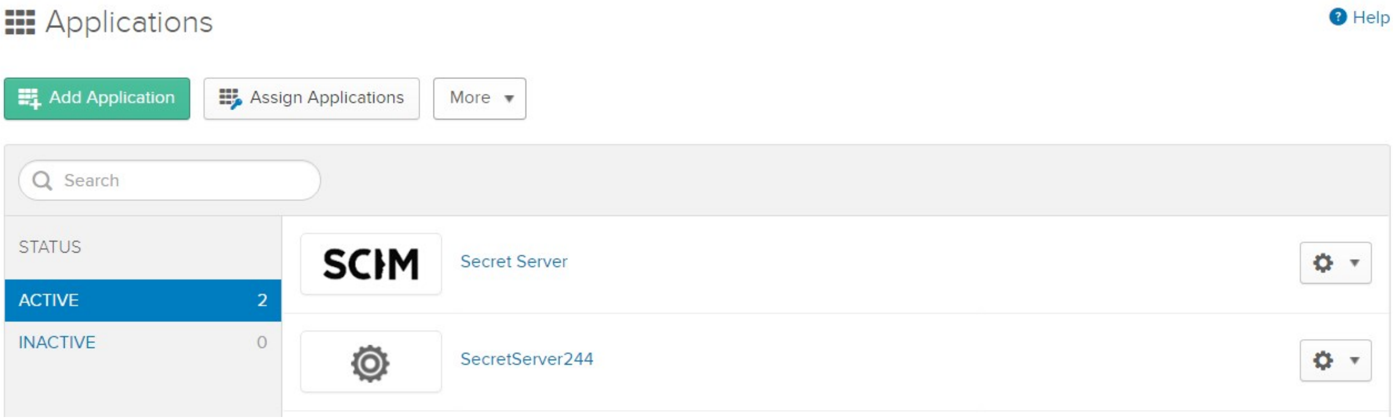
Base URL: http://202.56.202.85/SCIMConnector/v2

API Token: bearer wyck4BhXswn1VSRA8lkXi2n6qzBmWsbgqUdTExje9AU8bmQAYaoVz

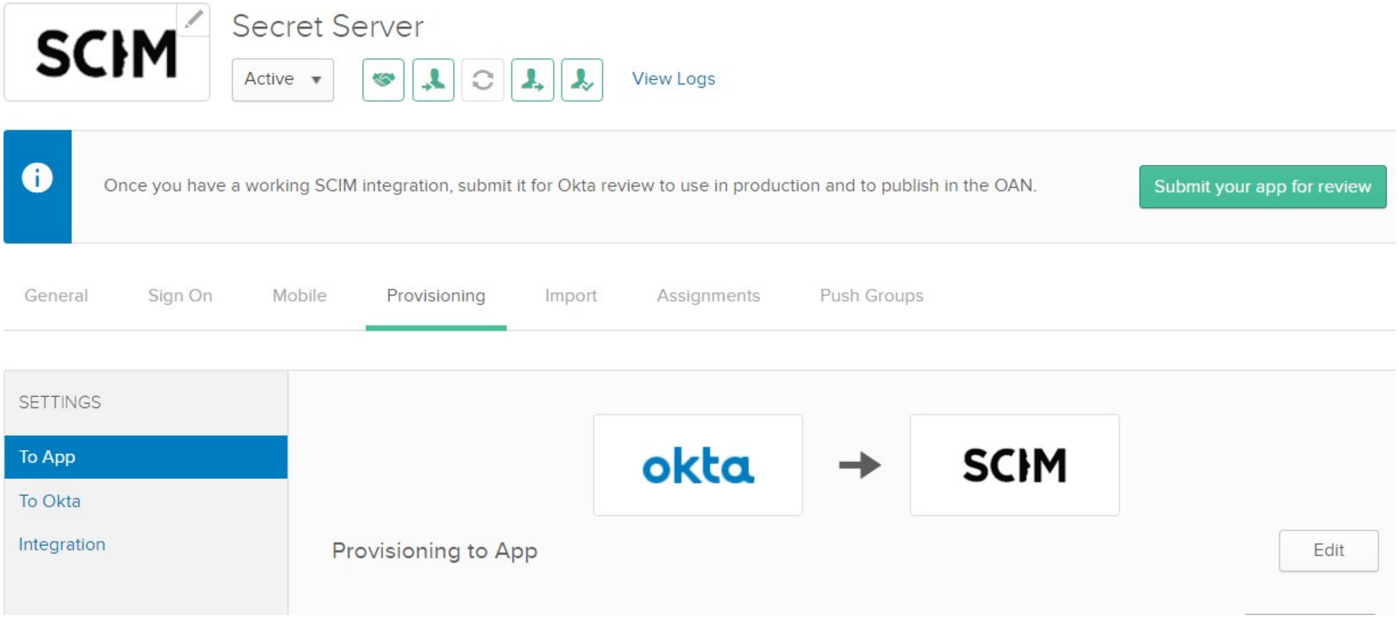
17. Click **Save** to complete the API integration.

Configure your Okta integration

1. Login to **Okta** using dev account.
2. Click on **application** | **applications**.
3. Click on the **SCIM application** created above.

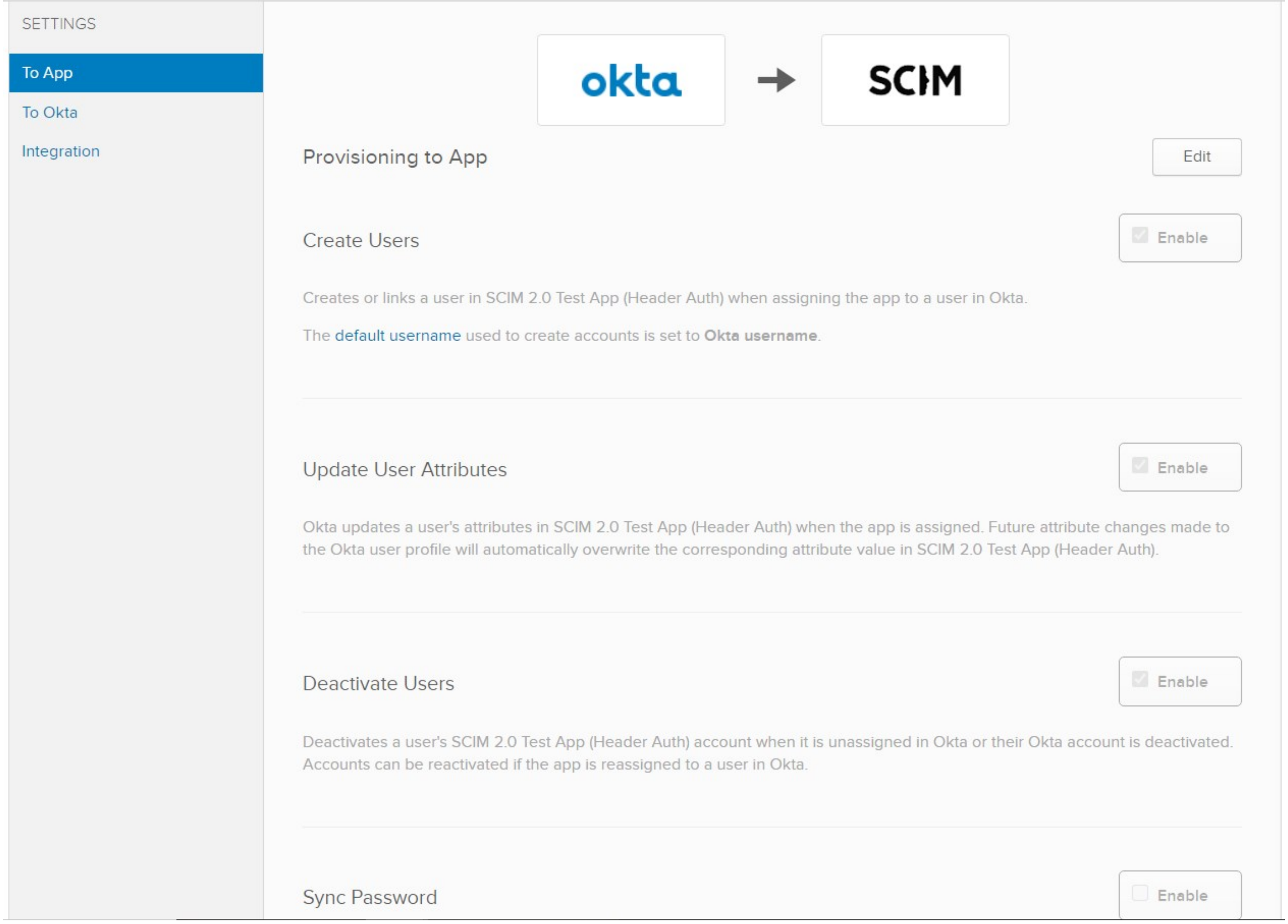


4. Click On the **Provisioning** tab of your Okta integration page, there are now three options listed in the **SETTINGS** panel:
   - **To App**
   - **To Okta**
   - **API Integration**
5. Click **App**.



Once you have a working SCIM integration, submit it for Okta review to use in production and to publish in the OAN.

6. Click **Edit** to make changes to the following sections.



Provisioning to App

Create Users

☑ Enable

Creates or links a user in SCIM 2.0 Test App (Header Auth) when assigning the app to a user in Okta.

The default username used to create accounts is set to **Okta username**.

Update User Attributes

☑ Enable

Okta updates a user's attributes in SCIM 2.0 Test App (Header Auth) when the app is assigned. Future attribute changes made to the Okta user profile will automatically overwrite the corresponding attribute value in SCIM 2.0 Test App (Header Auth).

Deactivate Users

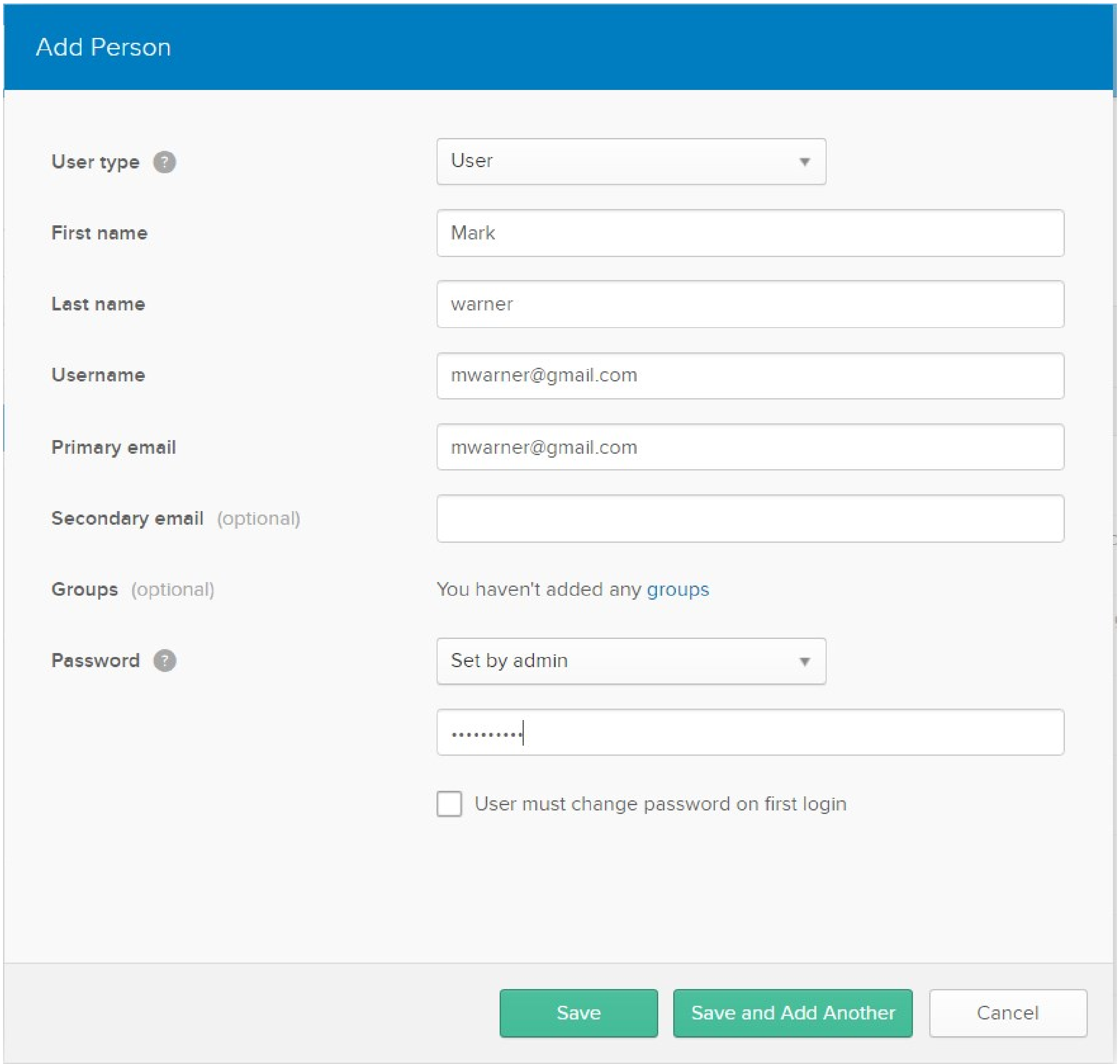
☑ Enable

Deactivates a user's SCIM 2.0 Test App (Header Auth) account when it is unassigned in Okta or their Okta account is deactivated. Accounts can be reactivated if the app is reassigned to a user in Okta.

Sync Password

☐ Enable

7. Navigate to **Directory**.
8. Click on **People**.
9. Click on **add Person** and enter the details.
10. Click on **Save**.



Add Person

User type: User

First name: Mark

Last name: warner

Username: mwarner@gmail.com

Primary email: mwarner@gmail.com

Secondary email (optional):

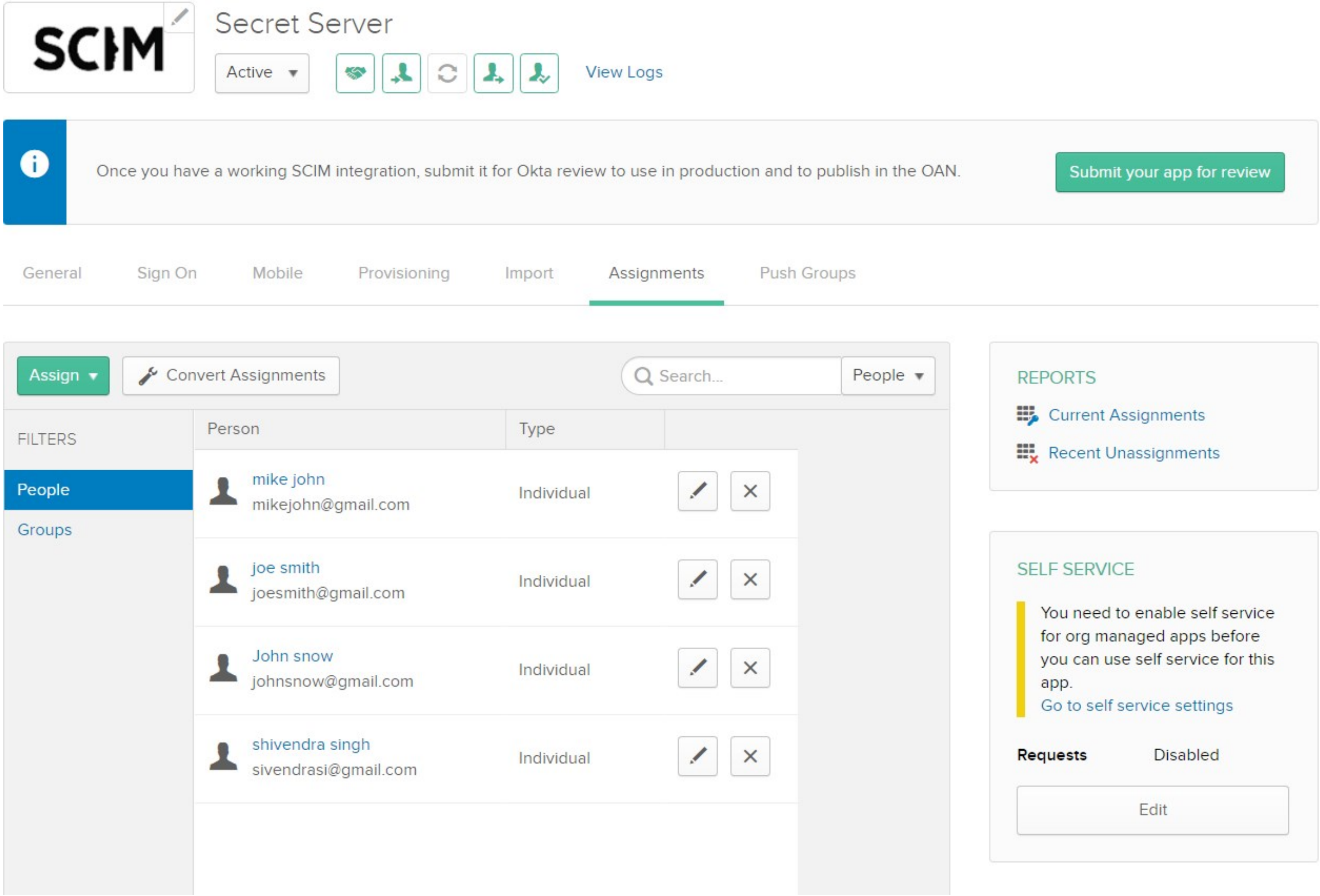
Groups (optional): You haven't added any groups

Password: Set by admin

••••••••••

□ User must change password on first login

11. Again go to **SCIM** Application and click on **Assignments**.



Secret Server

Once you have a working SCIM integration, submit it for Okta review to use in production and to publish in the OAN.

Person	Type
mike john mikejohn@gmail.com	Individual
joe smith joesmith@gmail.com	Individual
John snow johnsnow@gmail.com	Individual
shivendra singh sivendrasi@gmail.com	Individual

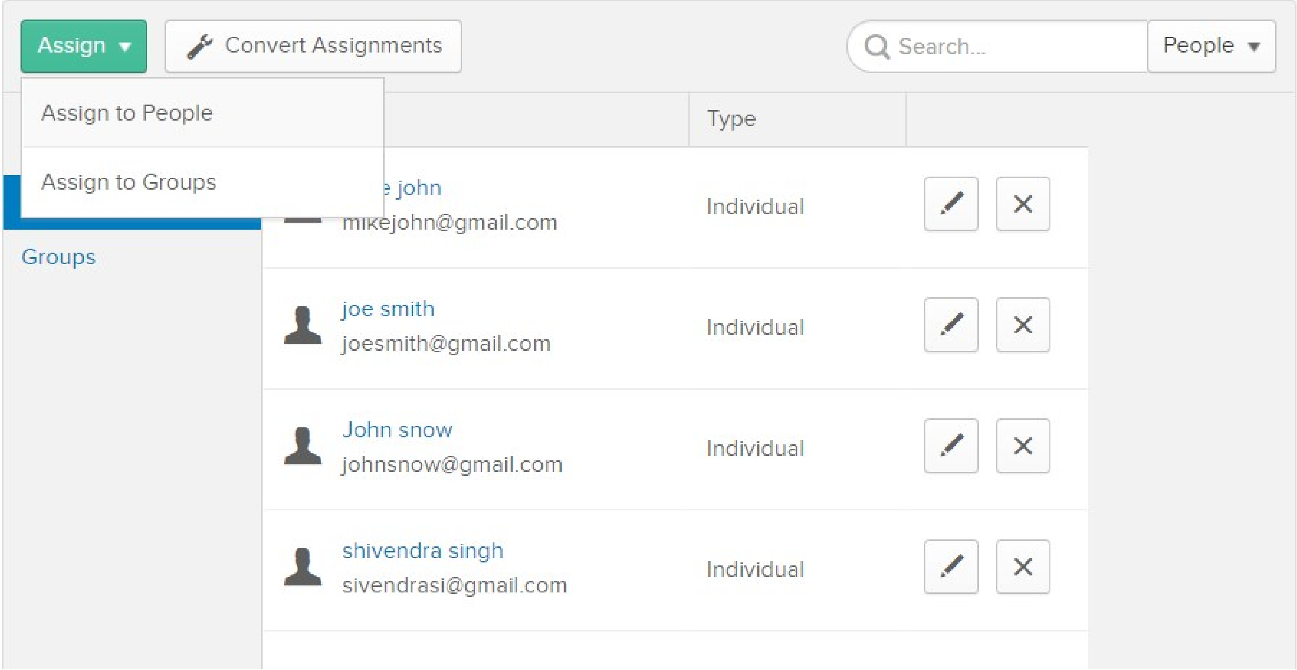
SELF SERVICE

You need to enable self service for org managed apps before you can use self service for this app.

Go to self service settings

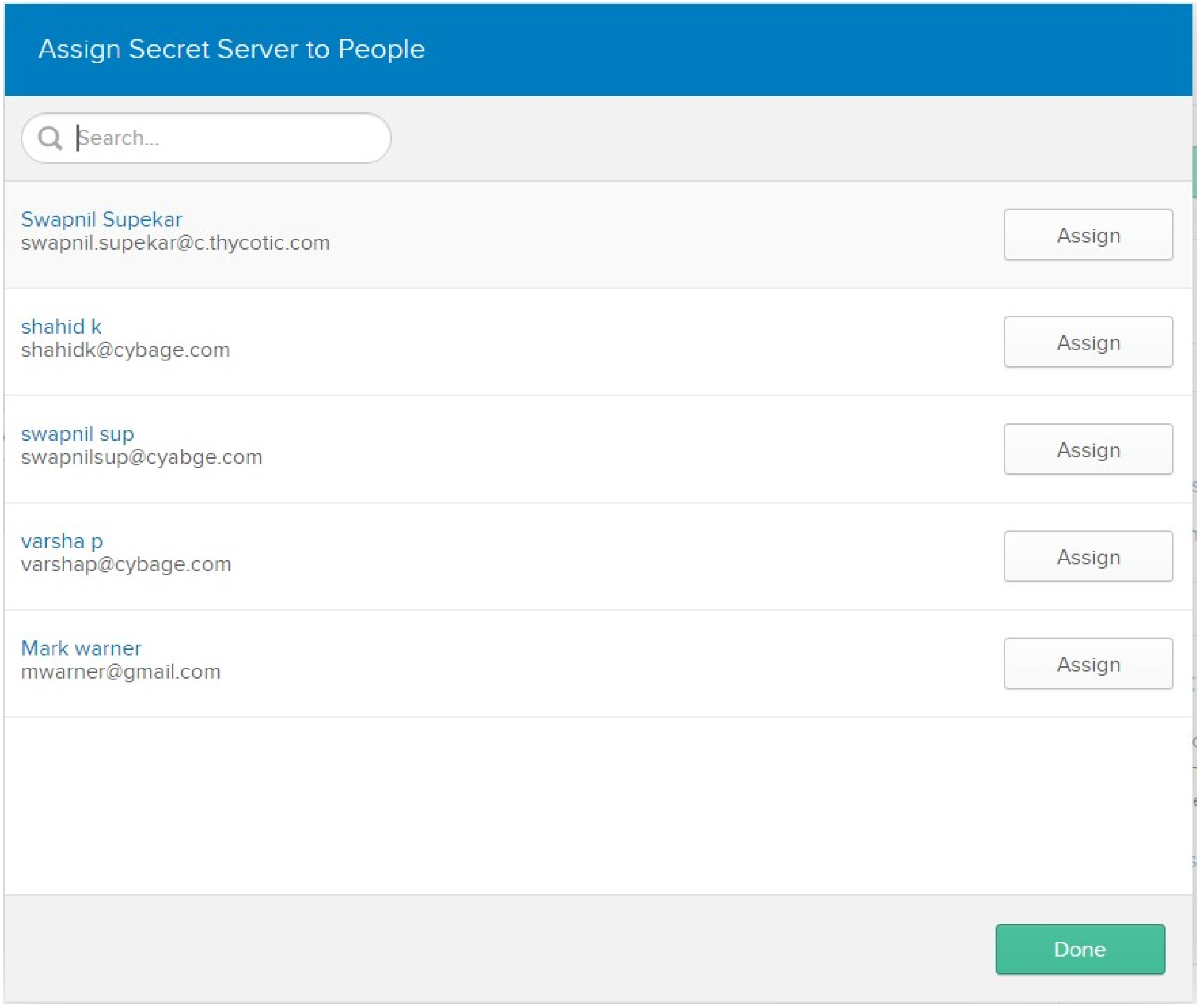
Requests Disabled

12. Click on **Assign** and select **Assign to People**.



	Type
e john mikejohn@gmail.com	Individual
joe smith joesmith@gmail.com	Individual
John snow johnsnow@gmail.com	Individual
shivendra singh sivendrasi@gmail.com	Individual

13. Select the user which we want to sync to Secret Server and select **Assign** and then click on **Done**.



Assign Secret Server to People

Search...

Swapnil Supekar
swapnil.supekar@c.thycotic.com

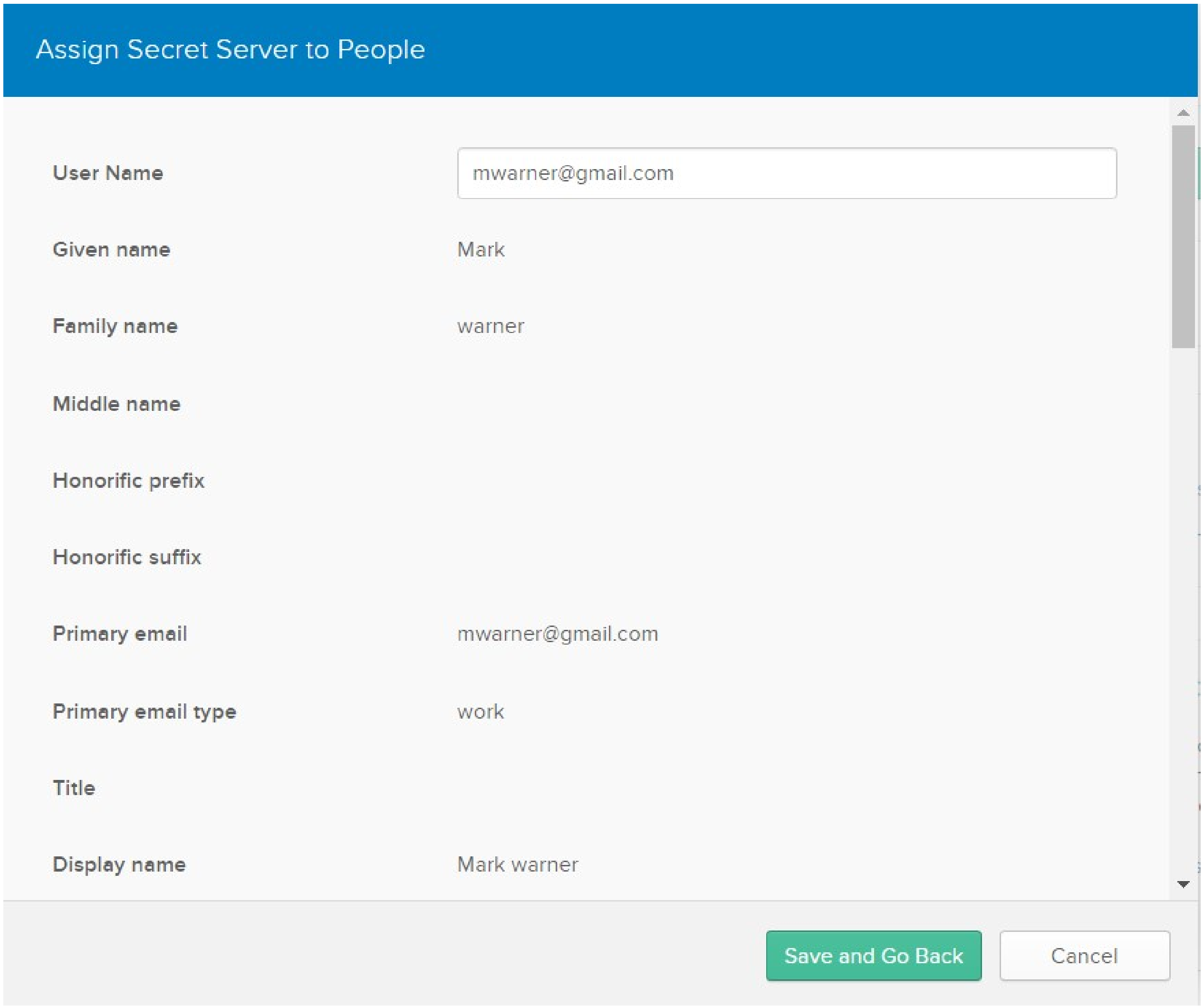
shahid k
shahidk@cybage.com

swapnil sup
swapnilsup@cyabge.com

varsha p
varshap@cybage.com

Mark warner
mwarner@gmail.com

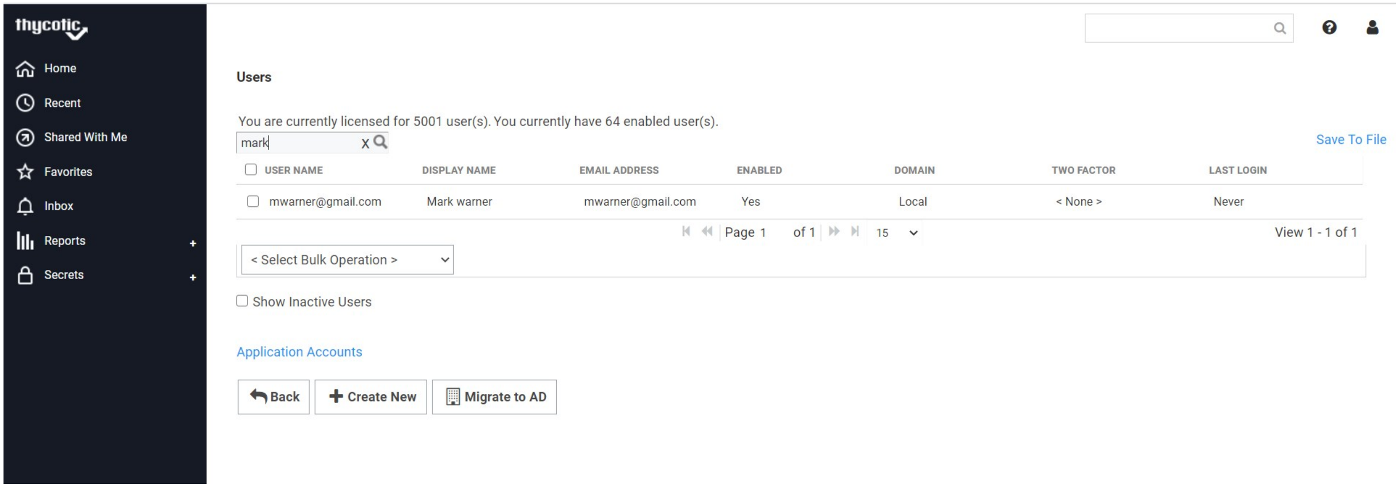
14. Click on **Save and GoBack**.



Assign Secret Server to People

User Name	mwarner@gmail.com
Given name	Mark
Family name	warner
Middle name	
Honorific prefix	
Honorific suffix	
Primary email	mwarner@gmail.com
Primary email type	work
Title	
Display name	Mark warner

15. User will be sync to Secret Server.
16. Login to **SecretServer l click on Admin l Users**.
17. Search for the user created in Okta in Secret Server



Users

You are currently licensed for 5001 user(s). You currently have 64 enabled user(s).

mark

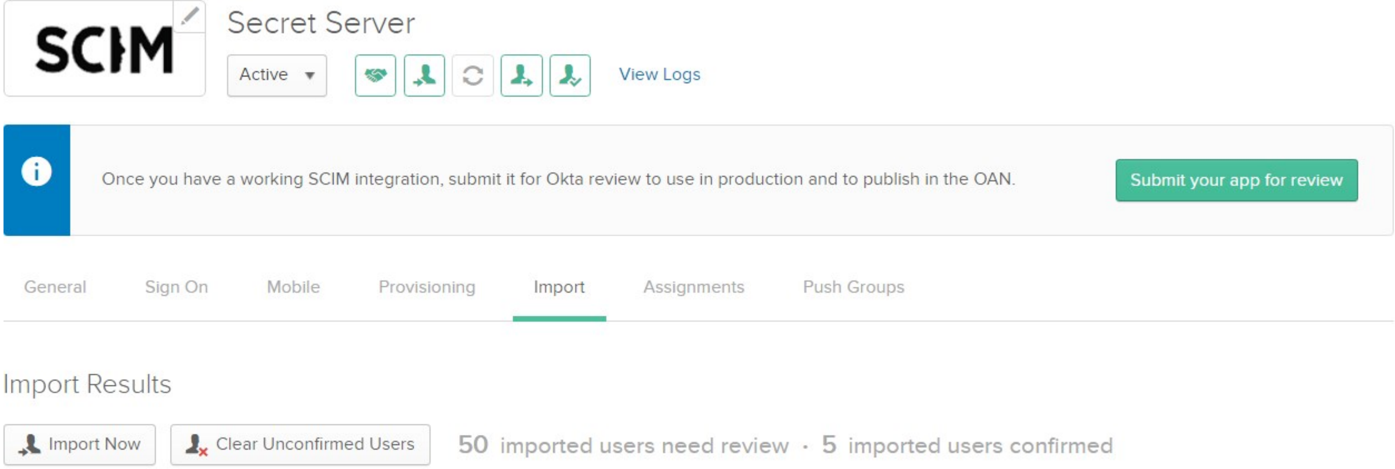
□ USER NAME	DISPLAY NAME	EMAIL ADDRESS	ENABLED	DOMAIN	TWO FACTOR	LAST LOGIN
□ mwarner@gmail.com	Mark warner	mwarner@gmail.com	Yes	Local	< None >	Never

Page 1 of 1 15 View 1 - 1 of 1

< Select Bulk Operation >

□ Show Inactive Users

18. To import the user and groups from Secret Server click on the **Import tab**.

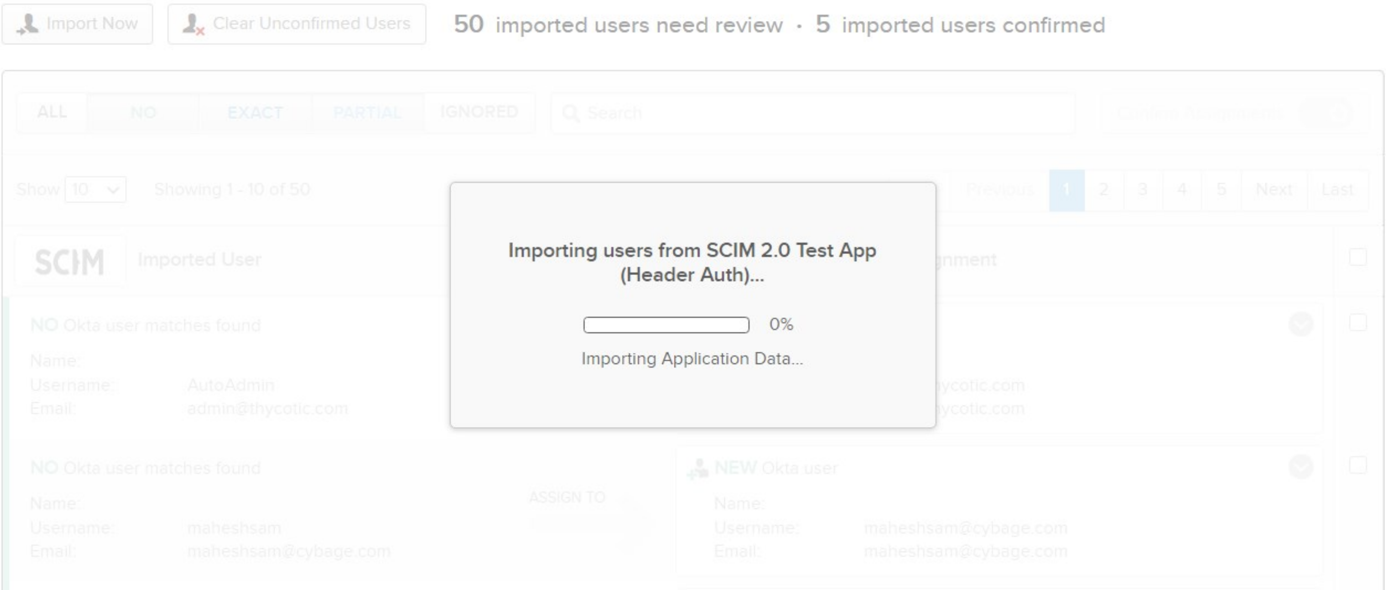


Once you have a working SCIM integration, submit it for Okta review to use in production and to publish in the OAN.

Import Results

50 imported users need review · 5 imported users confirmed

19. Click on **Import Now**.



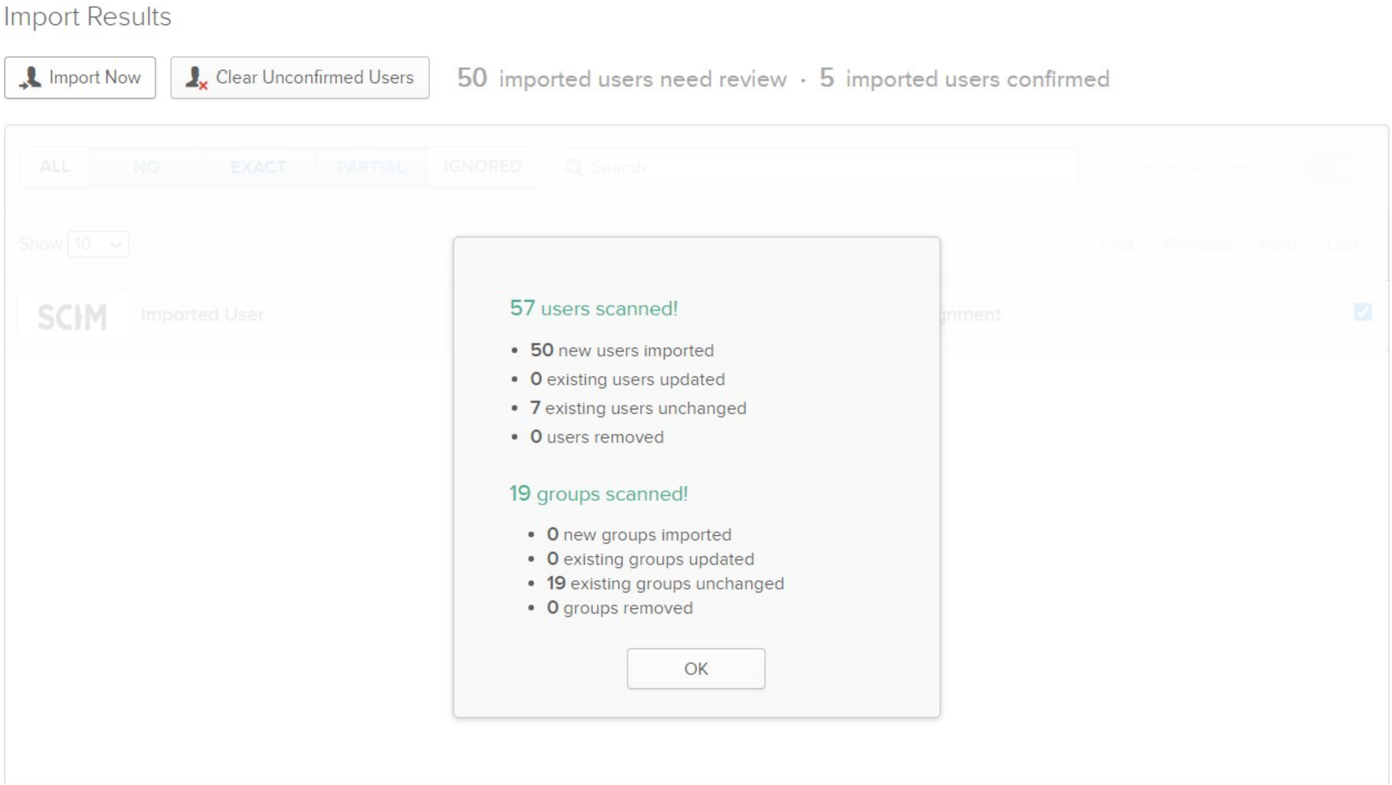
50 imported users need review · 5 imported users confirmed

Importing users from SCIM 2.0 Test App (Header Auth)...

0%

Importing Application Data...

20. After completion of process users from Secret Server will be displayed.



Import Results

50 imported users need review · 5 imported users confirmed

57 users scanned!

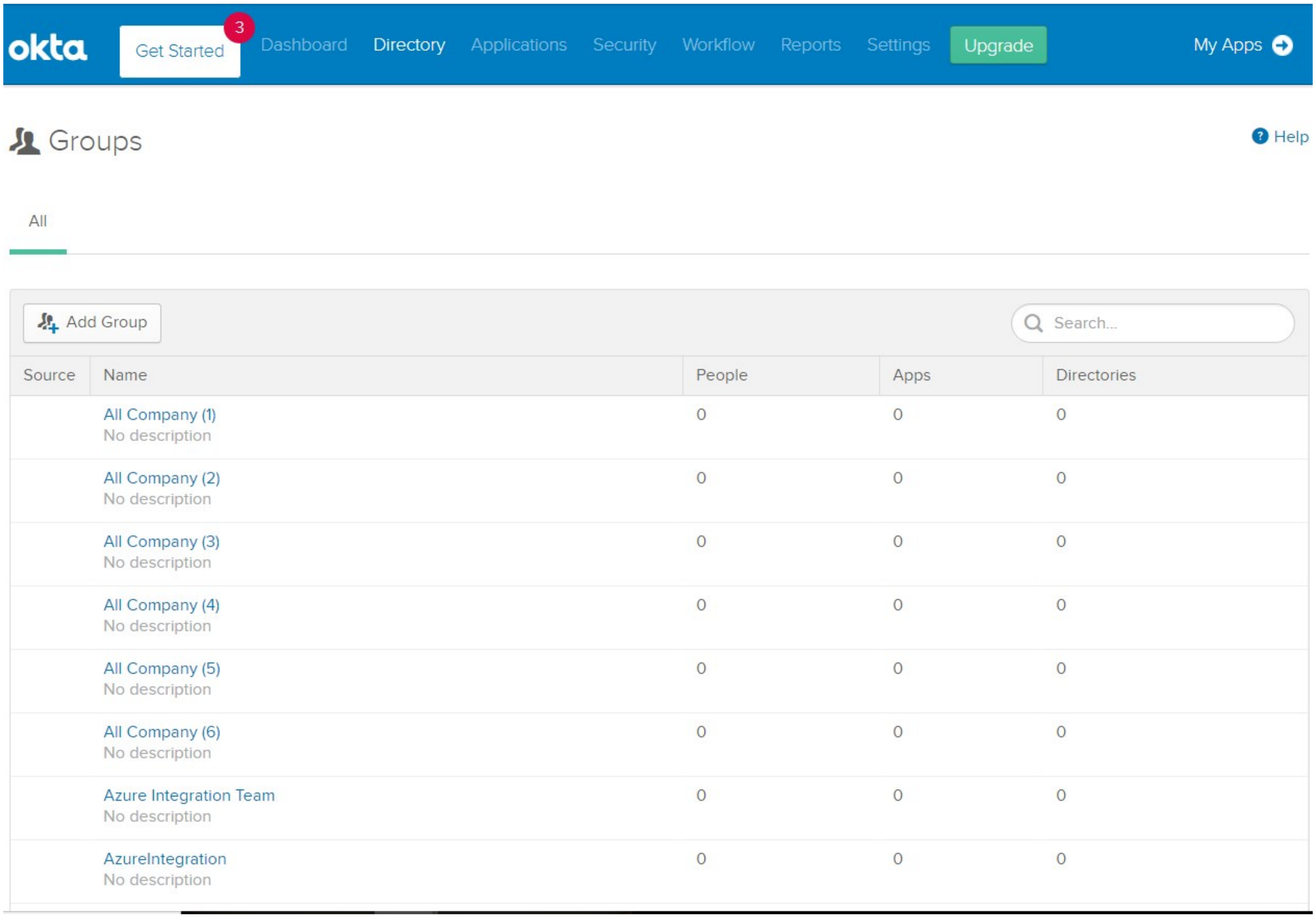
- 50 new users imported
- 0 existing users updated
- 7 existing users unchanged
- 0 users removed

19 groups scanned!

- 0 new groups imported
- 0 existing groups updated
- 19 existing groups unchanged
- 0 groups removed

21. Click on **Directory I Groups**.

22. SecretServer Groups will be displayed.



Groups

All

Source	Name	People	Apps	Directories
	All Company (1) No description	0	0	0
	All Company (2) No description	0	0	0
	All Company (3) No description	0	0	0
	All Company (4) No description	0	0	0
	All Company (5) No description	0	0	0
	All Company (6) No description	0	0	0
	Azure Integration Team No description	0	0	0
	AzureIntegration No description	0	0	0

Okta for SAML Integration

The integration between Thycotic Secret Server and Okta is created and maintained by Okta. This document provides guidance and best practice for implementing the integration. It is based on the following publicly available documentation from the vendor and testing performed by Thycotic. Integrations are supported to the extent of the third-party product procedures documented for this integration. Please contact the third-party for any customized setup of the integrated product.

This document is to serve as the configuration document for integrating Okta for SAML and Secret Server.

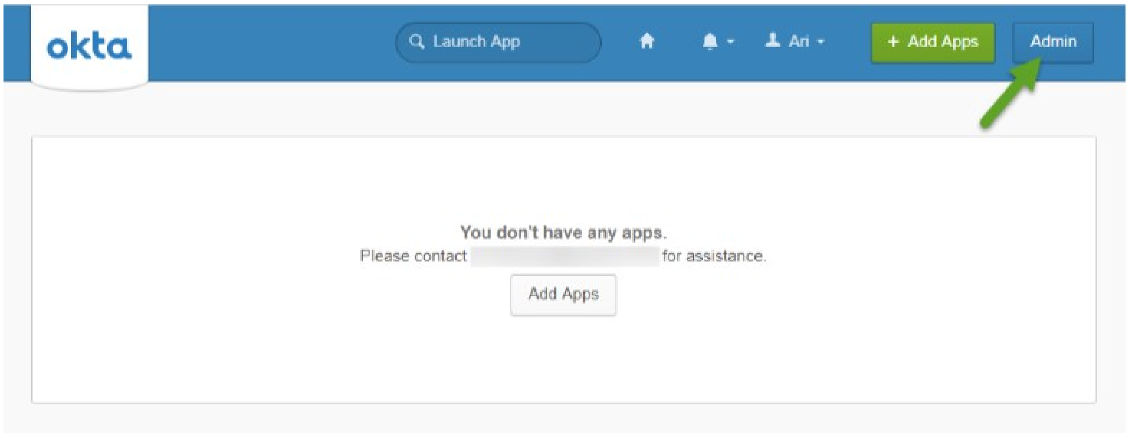
Secret Server acts as a SAML Service Provider that can communicate with Okta for authentication. This allows Secret Server the use of SAML Identity Provider (IdP) authentication instead of the normal authentication process for single sign-on (SSO). To do this, Secret Server acts as a SAML Service Provider (SP) that can communicate with any configured SAML IdP, including Okta.

Getting Started with Okta for SAML

As an Okta Administrator you can create new apps to setup integration. As part of the getting started steps the Setting up Secret Server as a new App needs to be completed first.

Setting up Secret Server as a new App

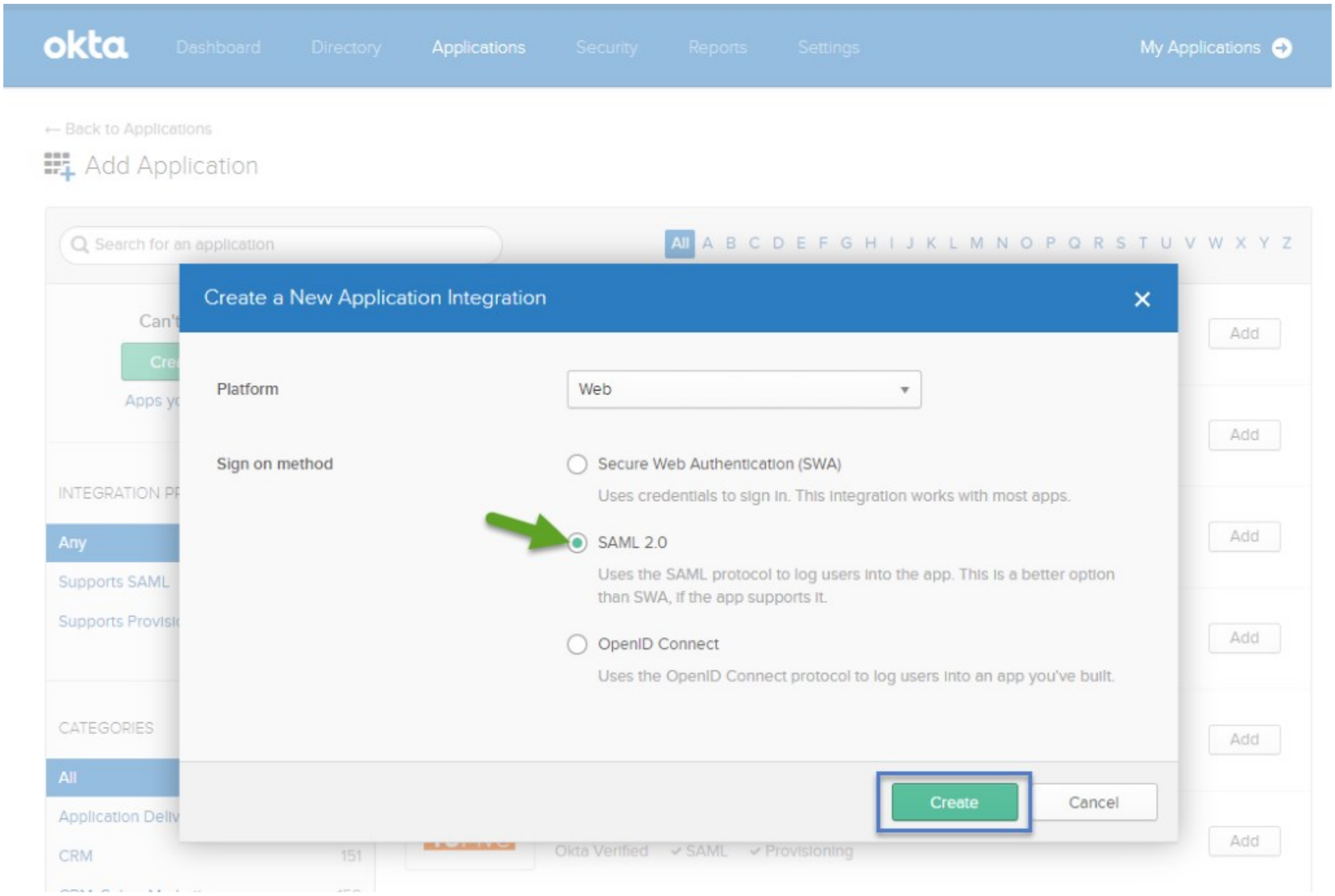
1. Login to your Okta instance using an administrative account.
2. Navigate to the App Home page ([Instance Name]/app/UserHome), and click **Admin** | **Applications** | **Add Application** | **Create New App**.



You don't have any apps.
Please contact [redacted] for assistance.

Add Apps

3. In the **Create a New Application** pop-up window, select **SAML 2.0** and click **Create**.



Create a New Application Integration

Platform: Web

Sign on method:

○ Secure Web Authentication (SWA)
Uses credentials to sign in. This integration works with most apps.

● SAML 2.0
Uses the SAML protocol to log users into the app. This is a better option than SWA, if the app supports it.

○ OpenID Connect
Uses the OpenID Connect protocol to log users into an app you've built.

Create | Cancel