Microsoft Azure Sentinel – ServiceNow app installation guide

Table of Contents

[Introduction 4](#_Toc75190297)

[Prerequisites 5](#_Toc75190298)

[Azure: Create the Service Principal 5](#_Toc75190299)

[Delegate permissions to the Service Principal 8](#_Toc75190300)

[ServiceNow: create a user for Azure Sentinel 10](#_Toc75190301)

[Installation 11](#_Toc75190302)

[Import the application in ServiceNow 11](#_Toc75190303)

[Configure the OAuth credentials 13](#_Toc75190304)

[Configure the application 15](#_Toc75190305)

[Verify the “Closure classification” table entries 16](#_Toc75190306)

[Verify the “Sentinel Severity to ServiceNow” table mapping 17](#_Toc75190307)

[Verify the “Sentinel State to ServiceNow” table mapping 18](#_Toc75190308)

[Verify the “ServiceNow Severity to Sentinel” table mapping 18](#_Toc75190309)

[Verify the “ServiceNow State to Sentinel” table mapping 19](#_Toc75190310)

[Configure the Azure Sentinel workspace(s) details 19](#_Toc75190311)

[Review and validate the system properties 21](#_Toc75190312)

[Additional configuration 23](#_Toc75190313)

[Classes 25](#_Toc75190314)

[Alerts 25](#_Toc75190315)

[getIncidentAlerts 25](#_Toc75190316)

[alertsToHtmlTable 25](#_Toc75190317)

[AppUtils 25](#_Toc75190318)

[CustomMapping 25](#_Toc75190319)

[Entities 25](#_Toc75190320)

[getIncidentEntities 25](#_Toc75190321)

[entitiesToHtmlTable 25](#_Toc75190322)

[getEntitiesByType 26](#_Toc75190323)

[SentinelIncidents 26](#_Toc75190324)

[getSentinelIncidents 26](#_Toc75190325)

[createIncidents 26](#_Toc75190326)

[updateSentinelIncident 26](#_Toc75190327)

[updateChangedIncidents 26](#_Toc75190328)

[createUrlForObject 27](#_Toc75190329)

[getIncidentComments 27](#_Toc75190330)

[addIncidentComments 27](#_Toc75190331)

# Introduction

This integration is a ServiceNow application providing bi-directional synchronization of incidents, between both platforms. It has the following capabilities:

* Incident creation (Azure Sentinel to ServiceNow only)
* Incident alerts synchronization
* Incident entities synchronization
* Incident comments synchronization
* Incident status synchronization
* Incident severity synchronization
* Incident owner assignment synchronization
* Incident custom properties support (requires custom code)

# Prerequisites

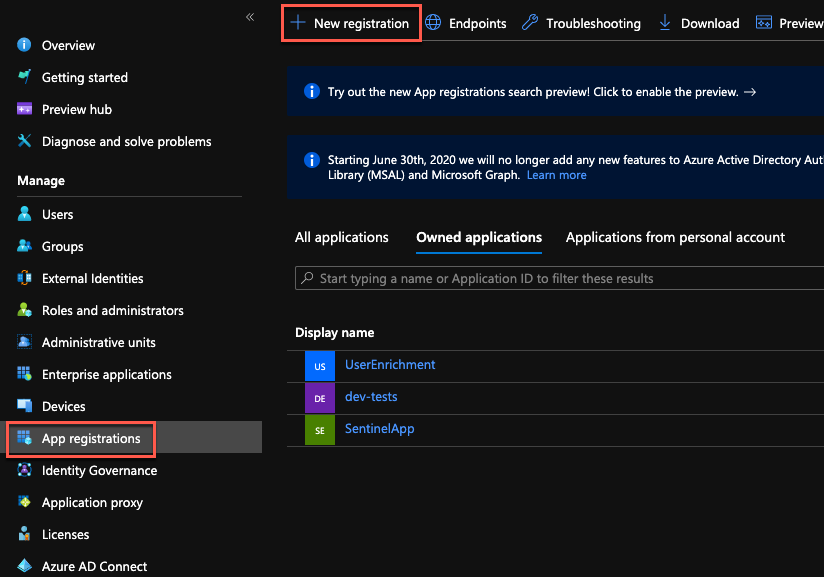
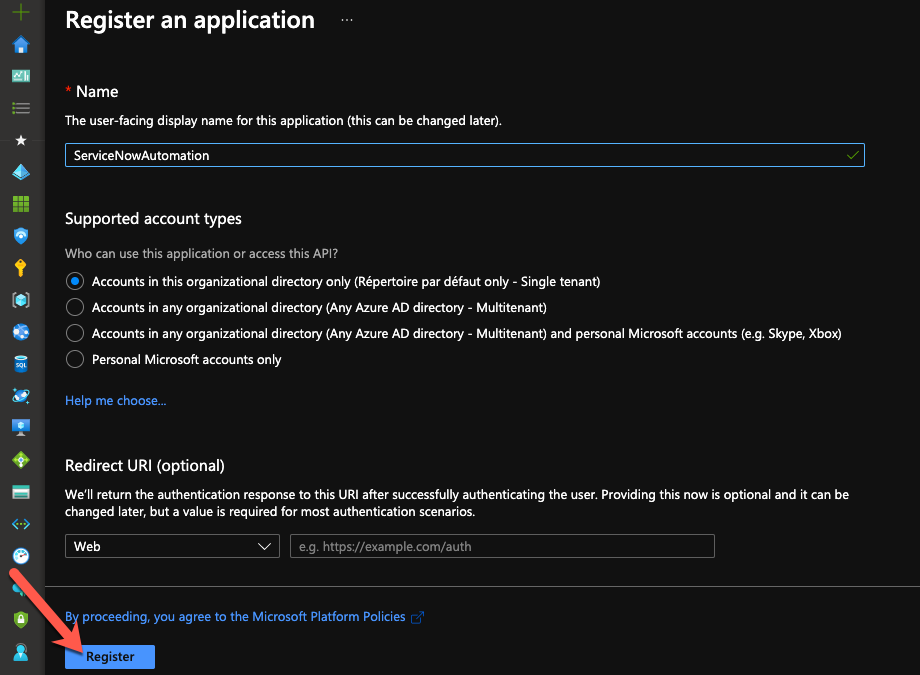
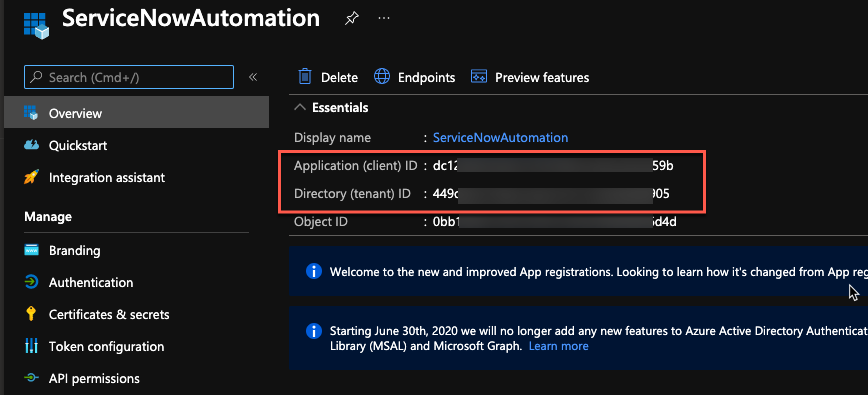
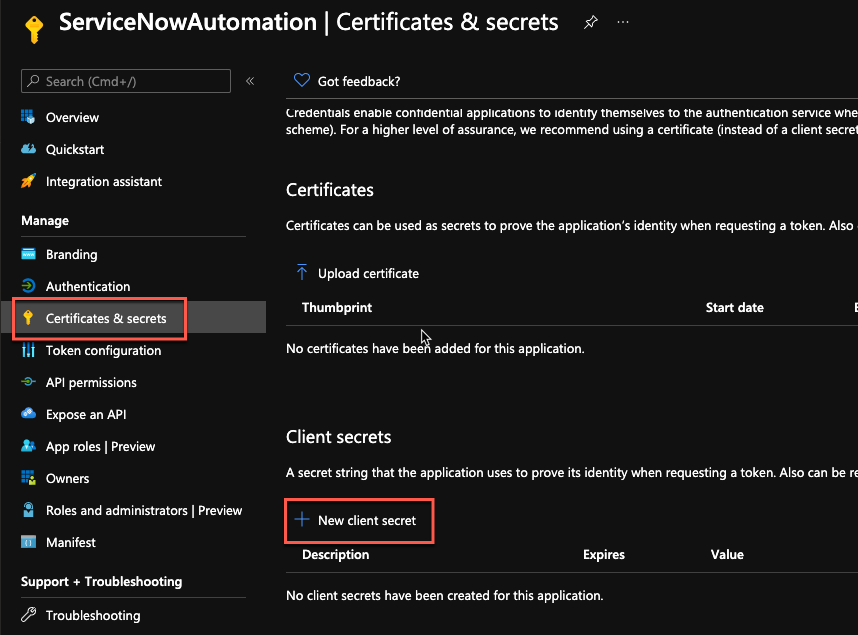
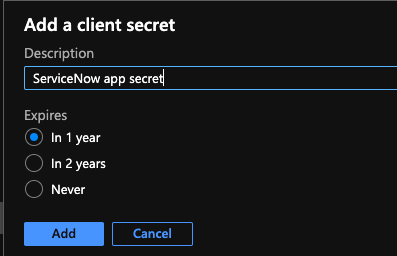
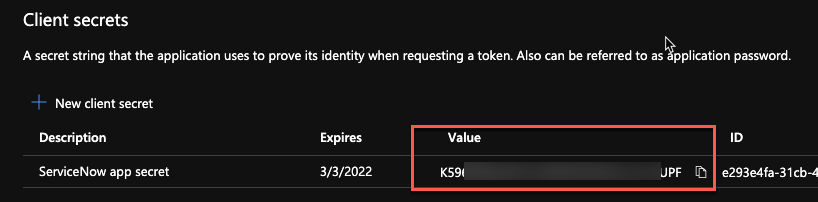
This ServiceNow application fully rely on the Azure Sentinel **Management API** to provide bi-directional sync between both platforms.

thumbnail image 1 of blog post titled 
 
 
  
 
 
 
    
  
   
    
      
       Azure Sentinel API 101
       
      
     
   
  
 
   
 
 
 
 
 

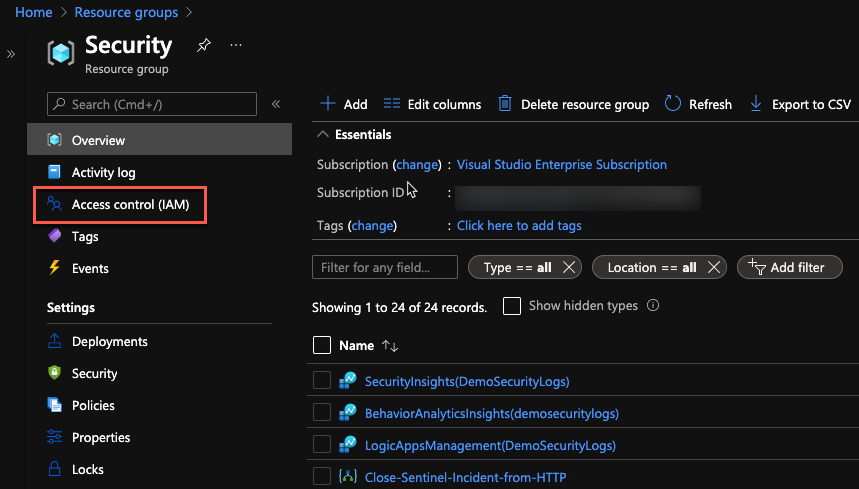
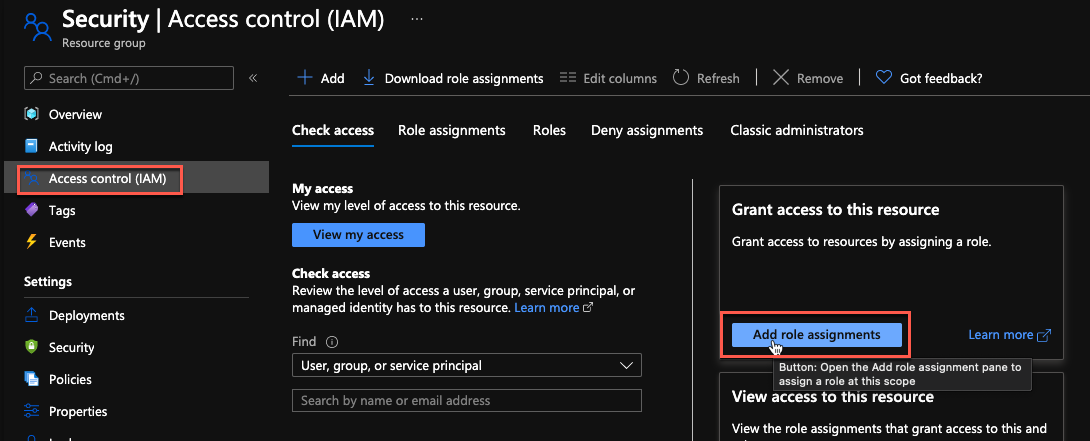
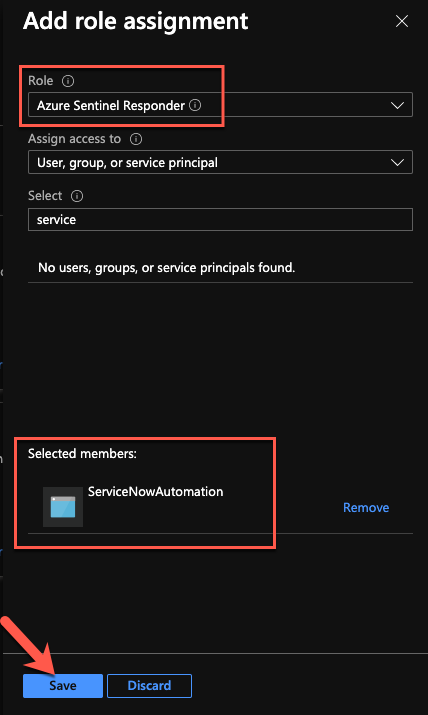

Azure Sentinel APIs reference

To provide access to our application, we have to create a Service Principal in Azure Active Directory, and assign to it the required permissions.

## Azure: Create the Service Principal

1. Go to the Azure portal, in Azure AD service, App Registrations:  
   <https://portal.azure.com/#blade/Microsoft_AAD_IAM/ActiveDirectoryMenuBlade/RegisteredApps>
2. Click on “New registration”.  
   
3. Provide a name for the app and click “Register”.  
   
4. Take note of the Application (client) ID and Directory (tenant) ID. We’ll need them during the ServiceNow configuration.
5. Go to “Certificates & secrets” and click on “New client secret”.  
   
6. Provide a name for the secret and a validity period.  
   Important: when the secret will expire, you’ll have to create a new one and update the ServiceNow configuration.  
   
7. Note the secret and keep it in a safe location for later use.  
   

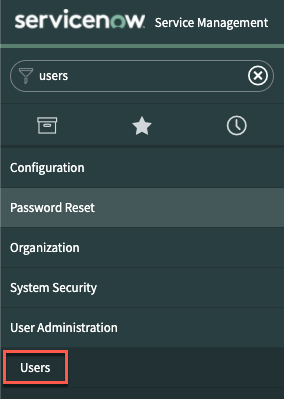
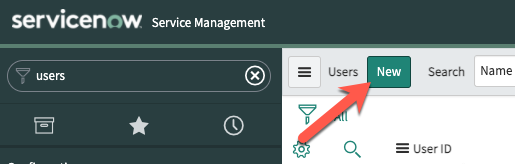
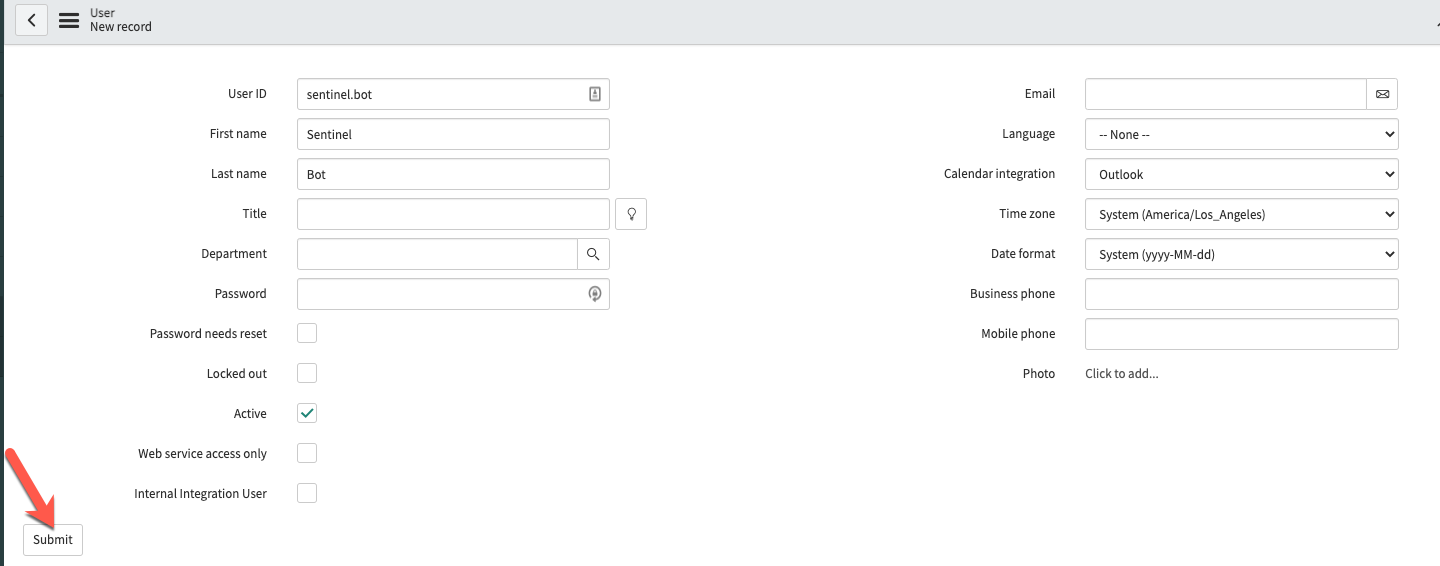
## Delegate permissions to the Service Principal

1. In the Azure portal, go to the Resource Group containing your Azure Sentinel workspace and click on “Access control (IAM)”.  
   
2. Click on “Add role assignments”.  
   
3. In the new blade, select the “Azure Sentinel Responder” role, then select the Service Principal we created before, and click on the “Save” button.  
   

We are now done with the Azure configuration part.

## ServiceNow: create a user for Azure Sentinel

To identify the incidents created from Azure Sentinel incidents, we will create a user. This user will be used as the “caller\_id” property, when creating new records.

1. In ServiceNow, under “User Administration”, click on “Users”.  
   
2. Click on the ”New” button.  
   
3. Provide the required details and click on “Submit”.  
   

# Installation

## Import the application in ServiceNow

1. Search for “update set” and select the “Retrieved Update Sets” module. Then, click on the “**Import Update Set from XML**” link.

Graphical user interface, text, application

Description automatically generated

1. Click on the “Choose File” button and select the application XML file.  
   Then, click on the “Upload” button.

Graphical user interface

Description automatically generated

1. Once uploaded, you will see the new imported update set. Click on it to open it.

Graphical user interface, application

Description automatically generated

1. Click on the “Preview Update Set” button.

Graphical user interface

Description automatically generated

Shape, rectangle

Description automatically generated

1. Click on the “Commit Update Set” button.

Graphical user interface, application

Description automatically generated

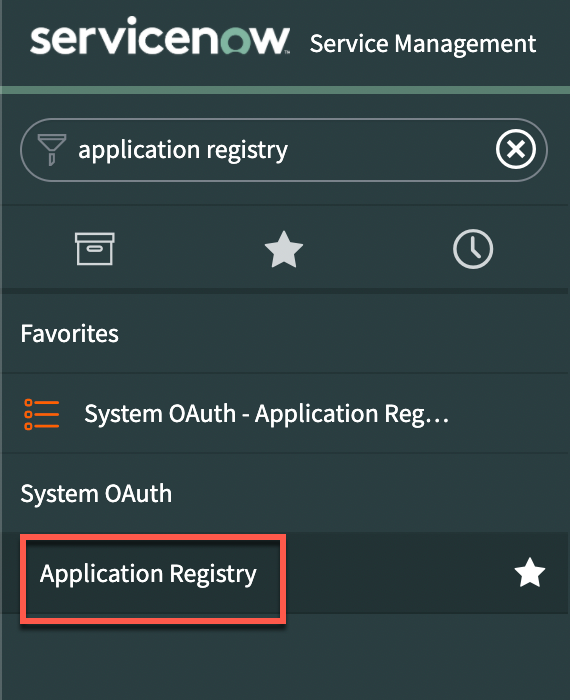
Rectangle

Description automatically generated

The application is now imported and is available in “Studio”.

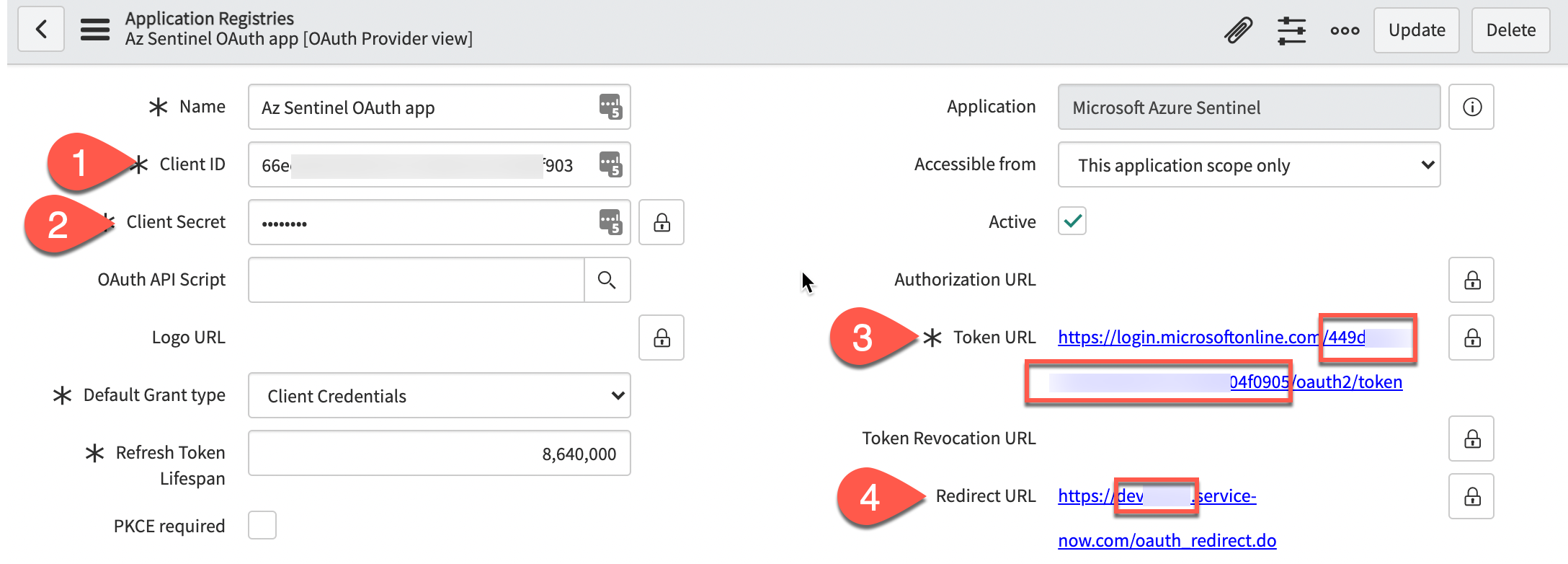
## Configure the OAuth credentials

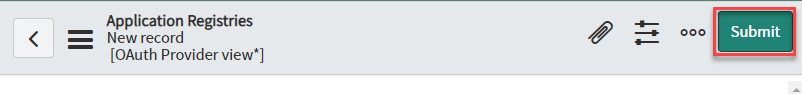
To be able use the Azure Sentinel Management API from ServiceNow, we must configure the credentials we created previously in Azure AD. This is done using an “Application Registry”.

1. Search for “Application Registry” and click on the link.  
   
2. Create a new set of credentials. We’ll use “Az Sentinel OAuth app” but you can use any name you want.  
   Graphical user interface, text, application

   Description automatically generated.
3. Select “Connect to a third party OAuth Provider”.  
   Graphical user interface, text, application

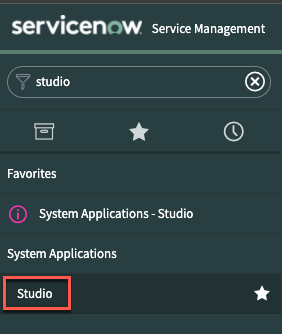
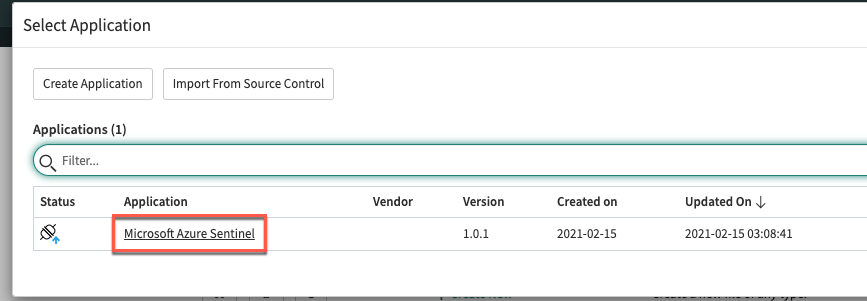
   Description automatically generated
4. On the credentials configuration page, we must provide the information we collected during the Service Principal creation:
   * Name: Az Sentinel OAuth app (can be different. This is the default name used by the workspace configuration)
   * Client ID (1): Azure AD application/client ID
   * Client secret (2): Azure AD client secret
   * Default Grant type: Client Credentials
   * Token URL (3): add your Azure AD tenant ID in the URL:  
     [https://login.microsoftonline.com/{AAD\_tenant\_id}/oauth2/token](https://login.microsoftonline.com/%7bAAD_tenant_id%7d/oauth2/token)
   * Token Revocation URL (4): add your ServiceNow instance name in the URL:  
     https://{instance\_name}.service-now.com/oauth\_redirect.do



1. Click on the “Submit” button to save your changes.  
   

## Configure the application

Now that we have imported the application, we must configure the details to connect to the Azure Sentinel Management API.

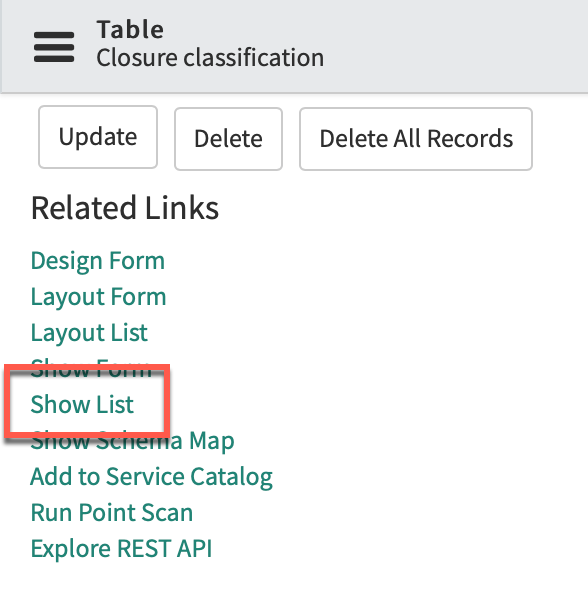
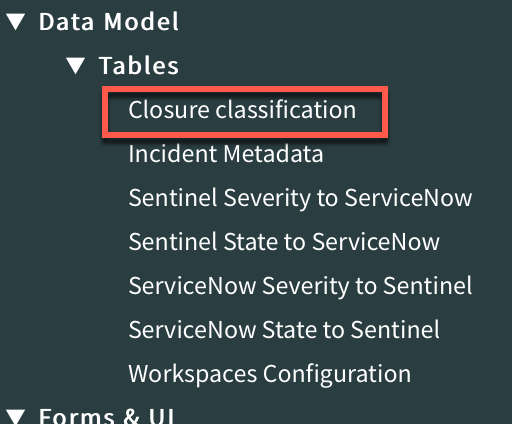
1. Search for “Studio” and open it. Then, select the newly imported application.  
     
   
2. Scroll to the “FixScript” section and run the “tablesInit” script.  
   This script will populate the tables.  
   Graphical user interface, text, application

   Description automatically generated  
     
   Graphical user interface, application

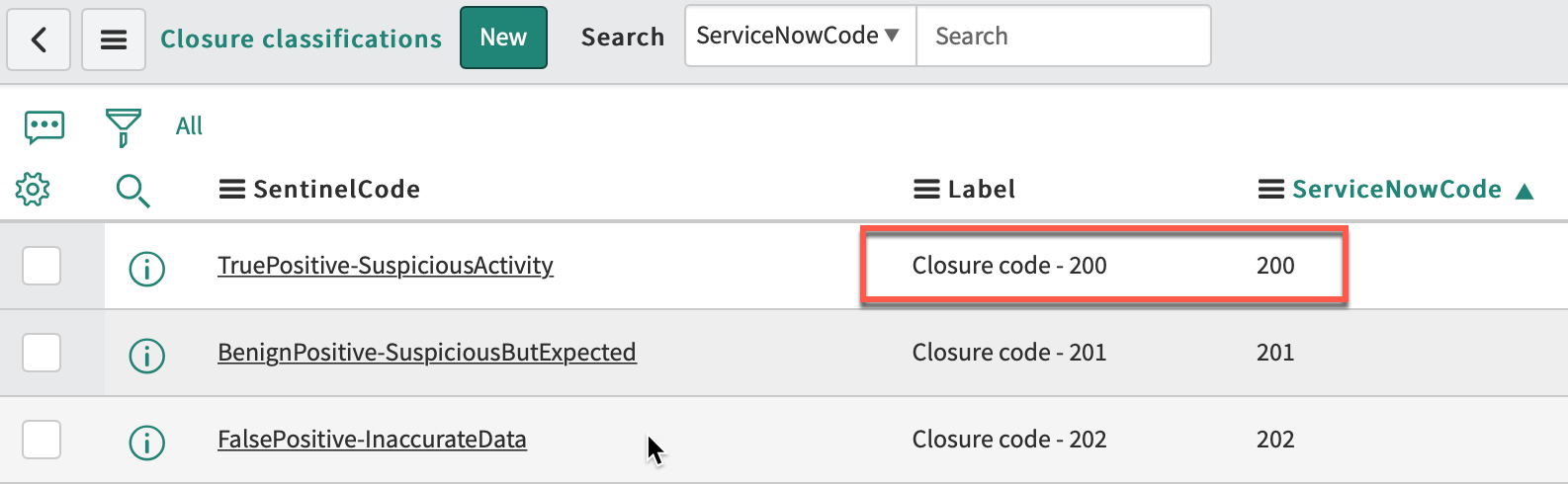
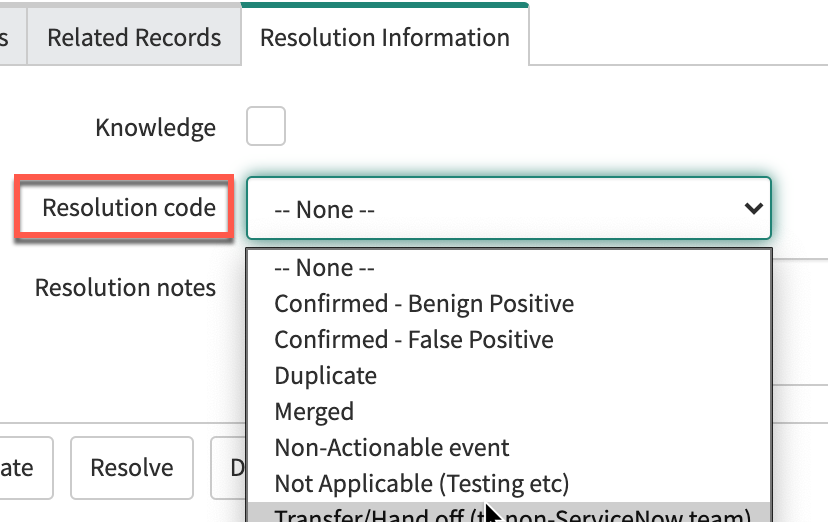
   Description automatically generated  
     
   Graphical user interface, text, application

   Description automatically generated

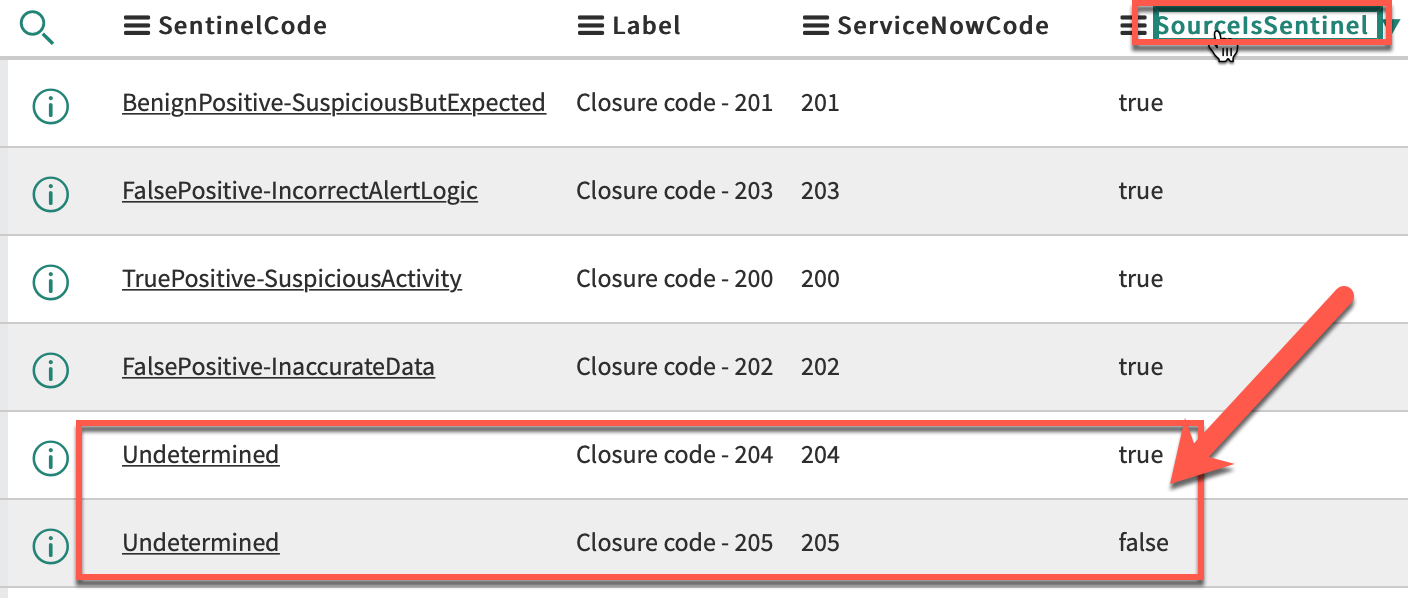
Verify the “Closure classification” table entries  
This table should match the closure codes you are using when closing your incidents.  
To verify the values, open the table list view at the **bottom** of the page (note: the procedure will be identical for all tables):



You should update the provided values with your environment ones:

**IMPORTANT**: in this table, the last column, “SourceIsSentinel” contains Boolean values to define which values should be used in ServiceNow when a close status has been set in Sentinel incidents.  
You should have only one “true” row per Sentinel possible status:



Verify the “Sentinel Severity to ServiceNow” table mapping  
This table is used to map the Sentinel severity to the ServiceNow value, when creating or updating Azure Sentinel incidents.  
Open the table => Show list, and review the values.  
Note that in our case, because Sentinel has four different severities values while we have only three in ServiceNow, both “Informational” and “Low” have been assigned the value **3**:



Verify the “Sentinel State to ServiceNow” table mapping  
This table is used to map the Sentinel state to the ServiceNow value, when creating or updating Azure Sentinel incidents.  
Open the table => Show list, and review the values to validate that they map your environment configuration.



Verify the “ServiceNow Severity to Sentinel” table mapping  
This table is used to map the ServiceNow severity to the Sentinel value, when updating ServiceNow incidents.  
Open the table => Show list, and review the values to validate that they map your environment configuration.



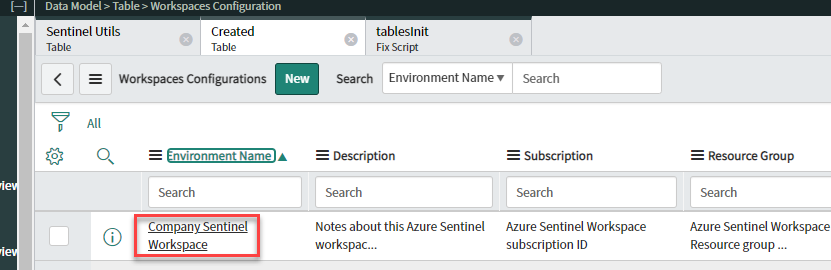
Verify the “ServiceNow State to Sentinel” table mapping  
This table is used to map the ServiceNow severity to the Sentinel value, when updating ServiceNow incidents.  
Open the table => Show list, and review the values to validate that they map your environment configuration.

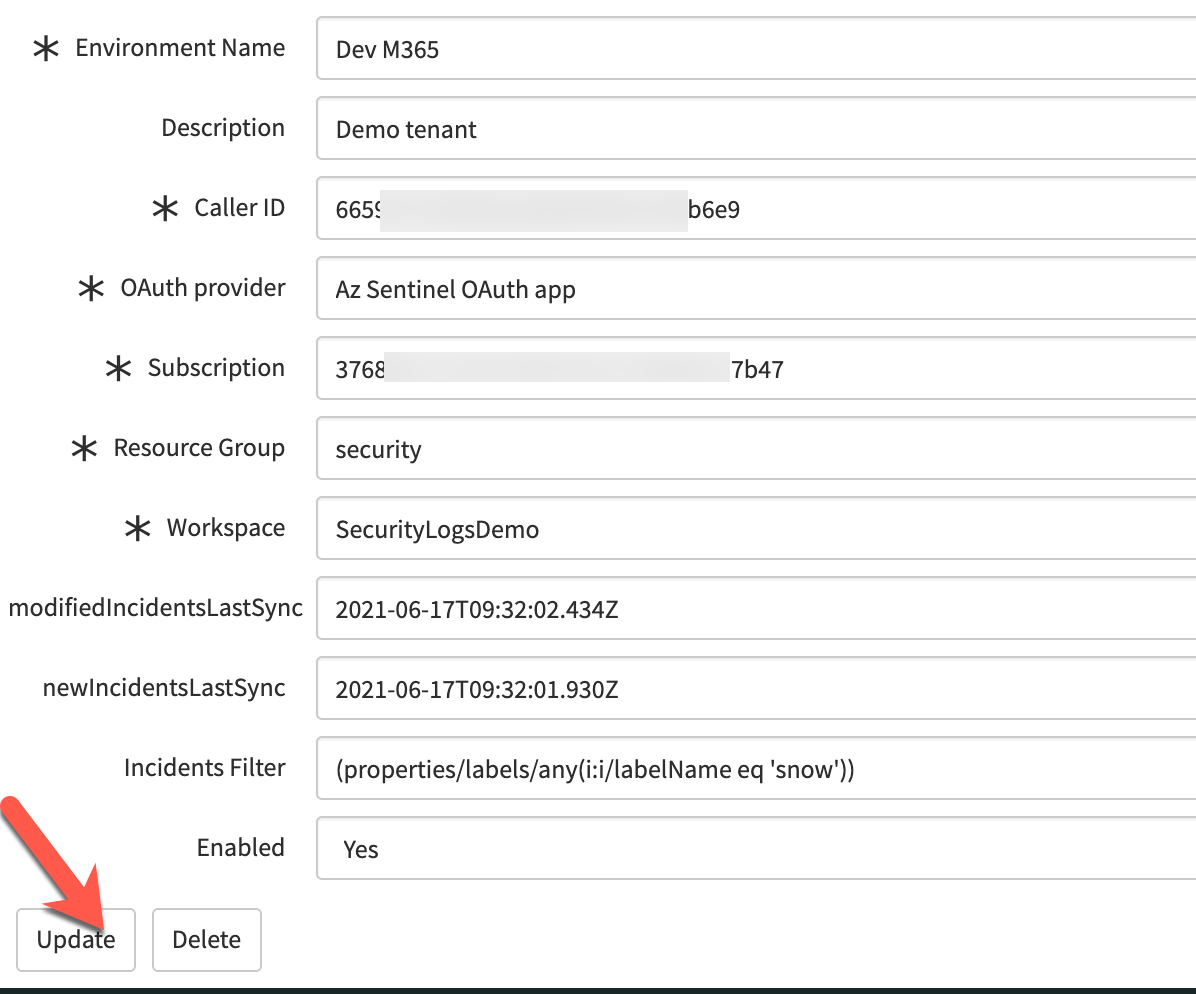


Configure the Azure Sentinel workspace(s) details  
The “Workspaces Configuration” table contains the Azure Sentinel workspaces configuration.  
 To do so, click on **Show List** at the bottom of the page.

Graphical user interface, text

Description automatically generated

Open to edit the current row (you can create multiple configurations):

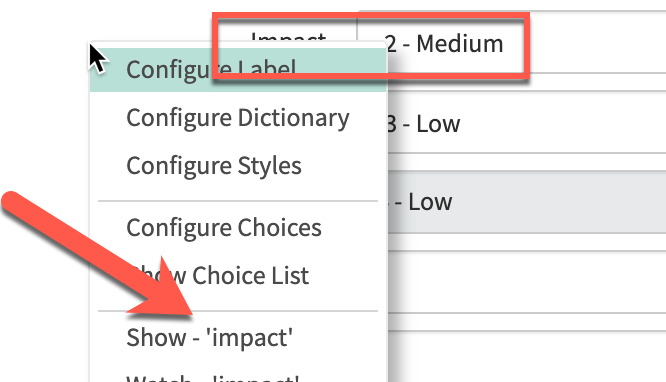
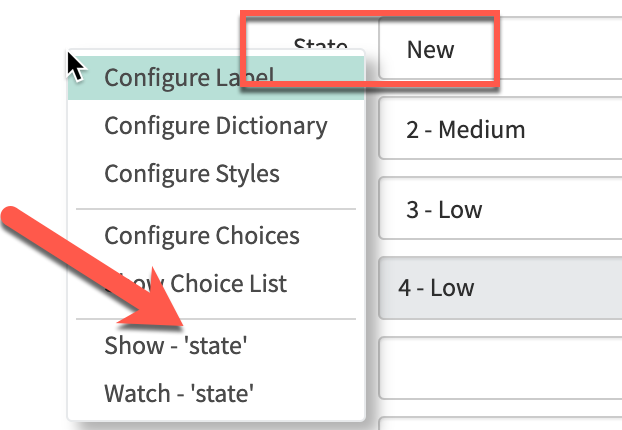
Provide the required values (available in Azure Sentinel) and click on the **Update** button:  


**Note**: In addition to the workspace configuration, you have the following properties:

* **newIncidentsLastSync**: timestamp automatically updated once the app successfully contact the Sentinel API to retrieve the new incidents since last update. If needed, you can manually change the value to query incidents newer than your specified date.
* **modifiedIncidentsLastSync**: timestamp automatically updated once the app successfully contact the Sentinel API to retrieve the updated incidents since last update
* **Incidents filter**: filter used to retrieve only the matching incidents from Sentinel API. By default, it filters the incidents with a tag “snow”. To get all incidents, just delete the content of this field.
* **Enabled**: boolean value to specify if the workspace is enabled or not. When disabled, the incidents are not retrieved and the timestamps are not updated.

### Review and validate the system properties

In addition to the configuration stored in the tables, the app keeps some information in system properties.  
Review the default values and change it to match your environment.

* **apiUrl**: URL to the Azure Sentinel API. If your workspace is in Gov Cloud, you must change it to [*https://management.usgovcloudapi.net*](https://management.usgovcloudapi.net)
* **apiVersion**: Azure Sentinel API version (should not be changed)
* **incidentTableName**: table where the incident are created
* **incidentUniqueKey**: ServiceNow incident property used to uniquely map incidents between Sentinel and ServiceNow. By default, the app uses “*correlation\_id*”. If you are already using this property, you should specify or create another one
* **severityField**: incident property to store the incident severity. By default, the app uses “*impact*”.  
  
* **statusField**: incident property to store the incident state. By default, the app uses “*state*”.  
  

Additional configuration  
If needed, create a new Business Rule to assign specific properties, like the “Assignment Group” or the “Category”.  
This can be achieved by clicking on “Create Application File”, selecting “Server Development”, “Business Rule”.  
  
