CREST
DATA SYSTEMS

Version Control

#	Document Version	Date	Owner	Document Status	Comments
1	1.0.0	22th Jul 2022	Crest Data Systems	Initial Draft	

Contents

Version Control 2

Contents 3

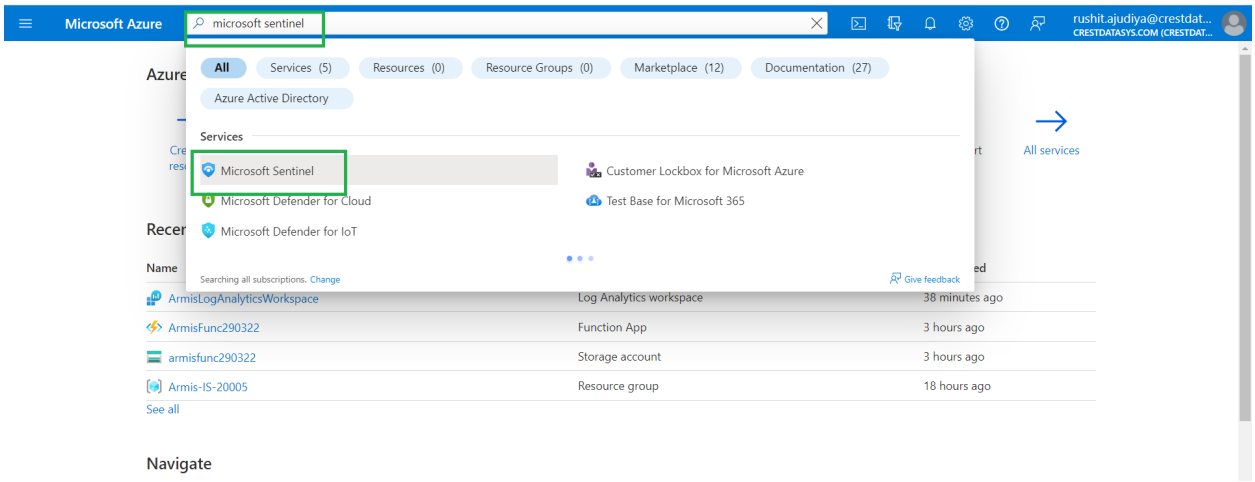
Armis Data Connector Installation Guide 4

Armis Logic App Installation Guide 11

Armis Data Connector Installation Guide

1. Login to Azure portal with below given link (https://portal.azure.com/?feature.BringYourOwnConnector=true) using the MS Azure credentials

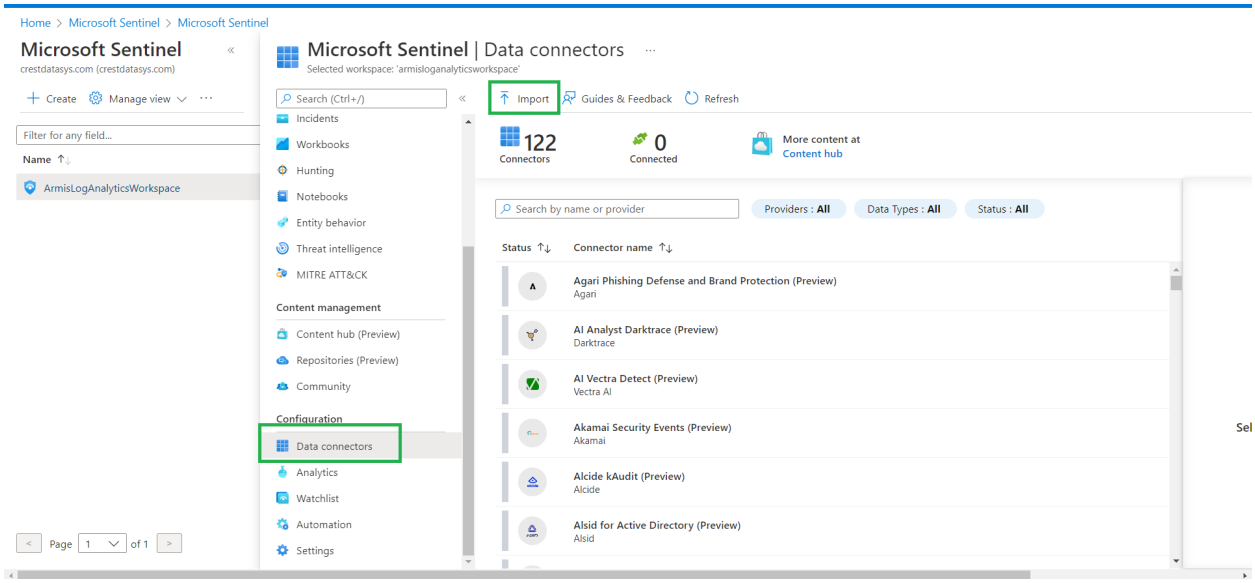
2. Now we have to search for the Microsoft Sentinel service in the azure portal search bar and select **Microsoft Sentinel** service.



3. Then select the Microsoft Sentinel Workspace in which you want to deploy the data connector.

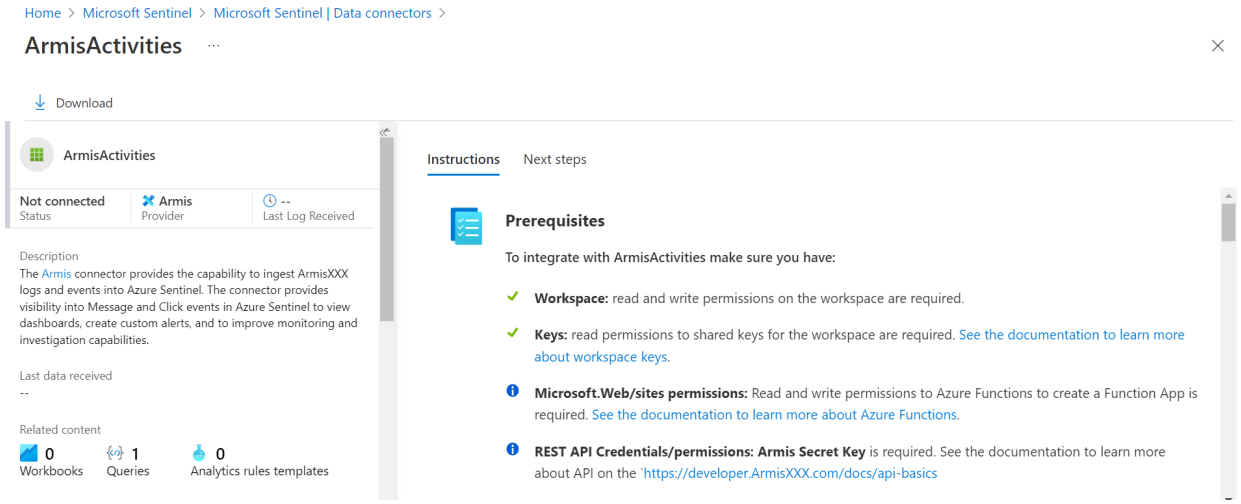
4. After that in the left panel scroll down and select the “**Data Connectors”.**

5. Now click on the import button.

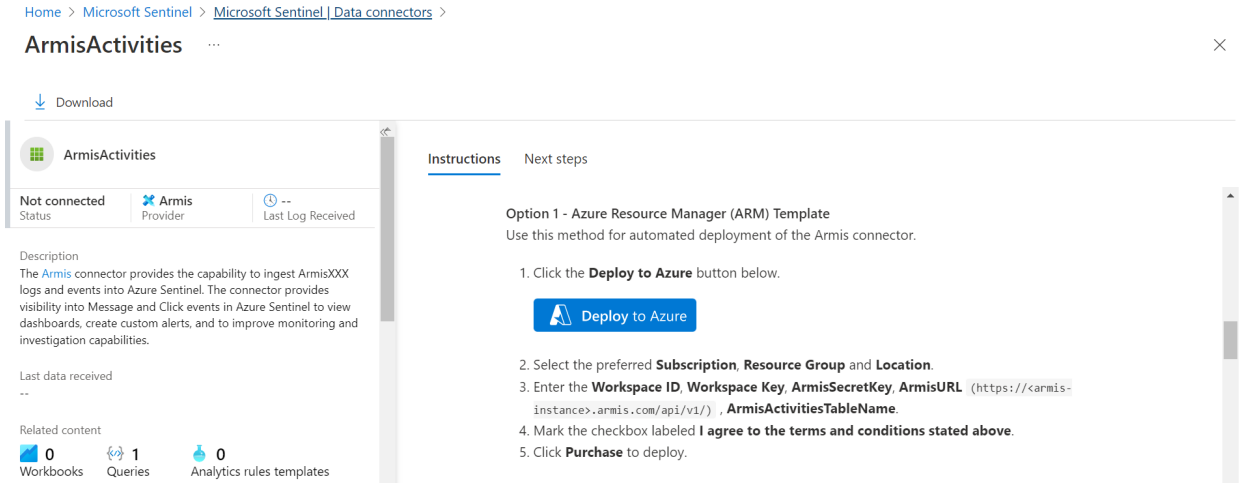


6. Now it will show a window in which you have to select the JSON file **(Armis_Activity_API_FunctionApp.json)** which will be provided by us.

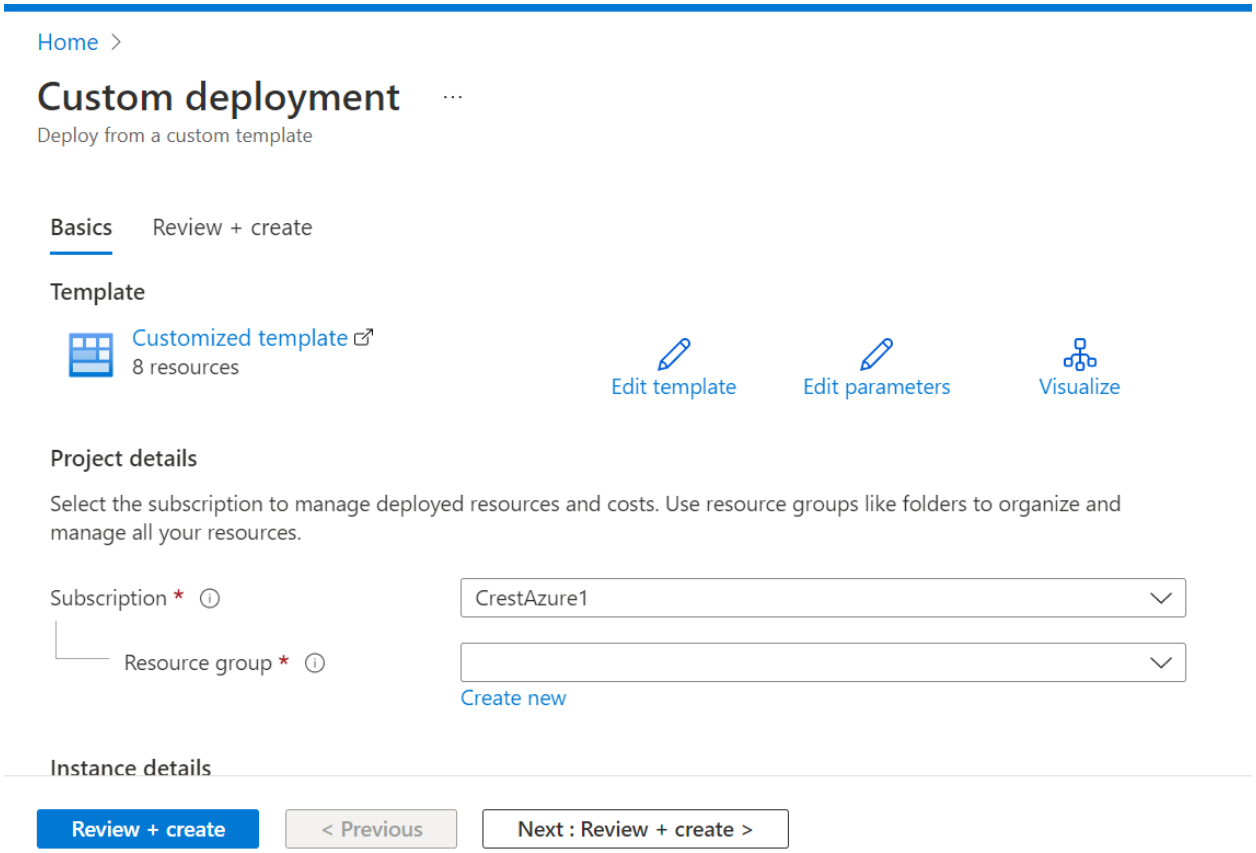
7. After that it will show a window like below.



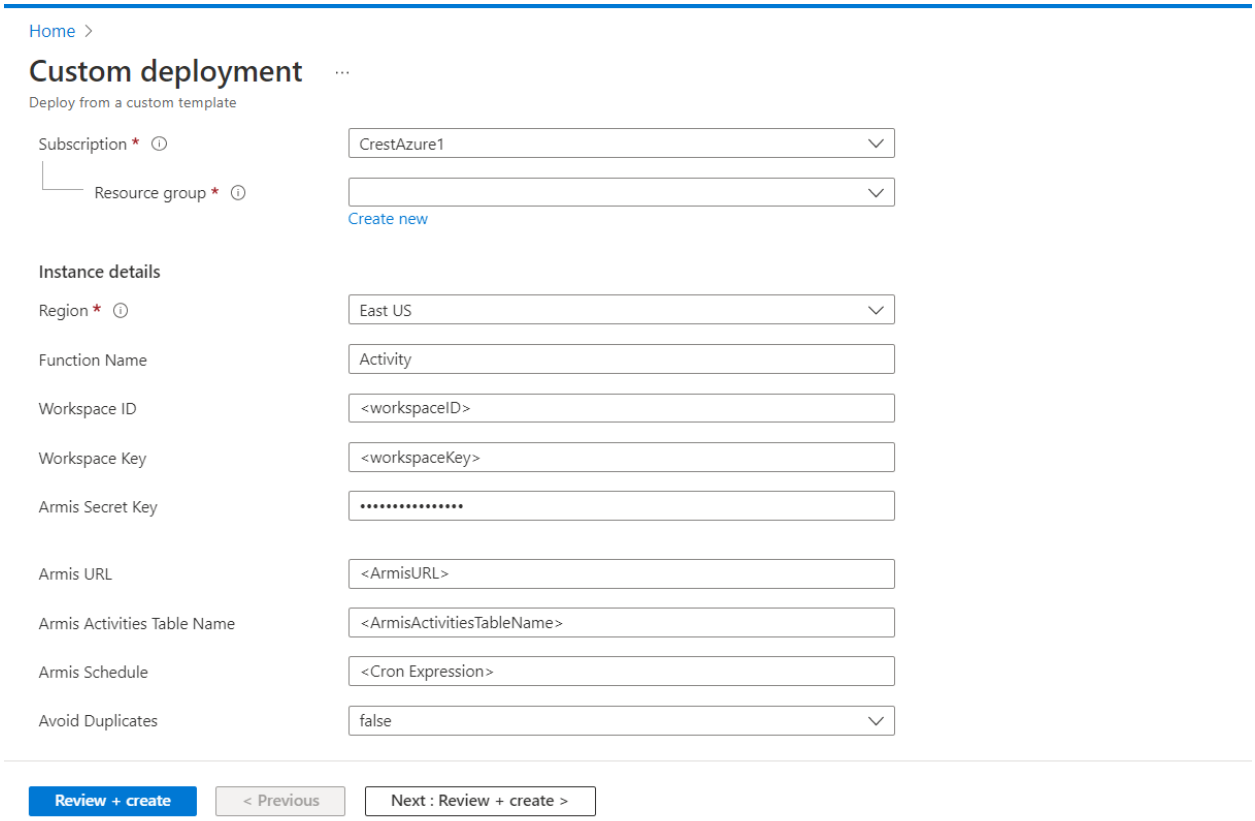
8. In the right panel scroll down and you will be able to see the "**Deploy to Azure"** button.



9. It will redirect you to the custom deployment screen where you can see the input fields and window like the below.



10. Add the information as mentioned below in input fields and click on “**Review + Create**’.

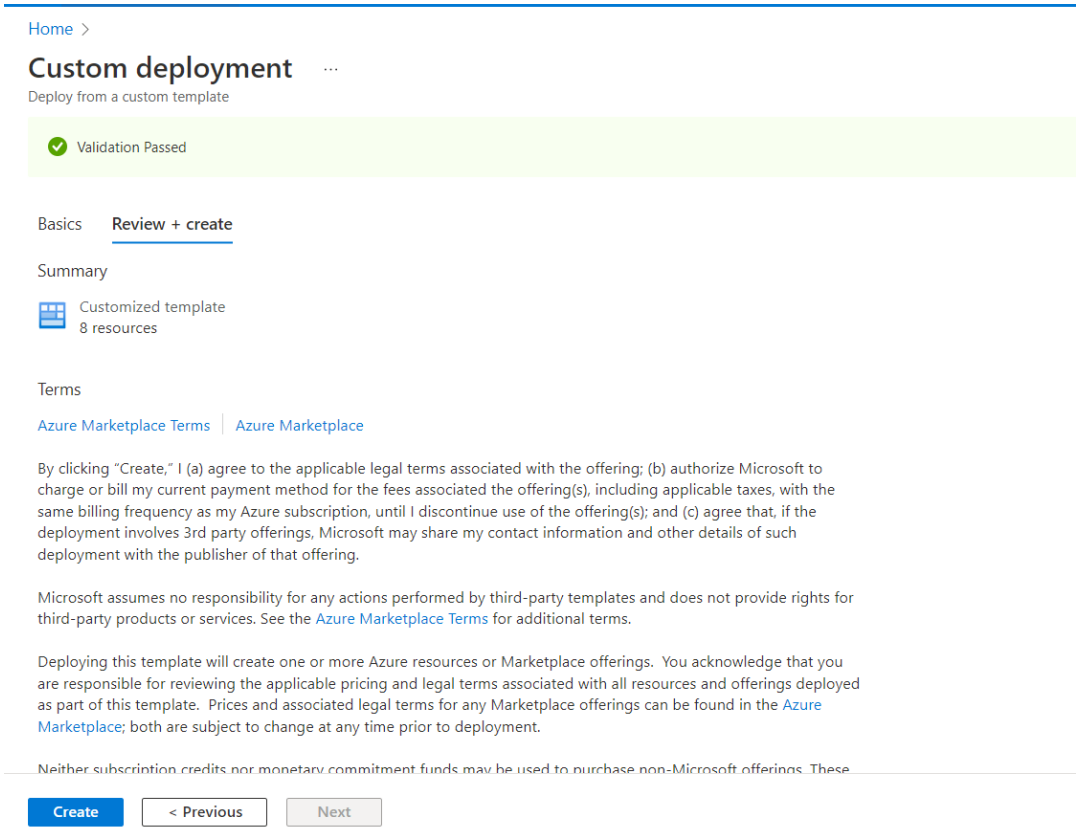


- **Subscription** : Select the subscription of your account.
- **Resource group** : Resource group in which you want to deploy data connector(Function App)
- **Region** : Select the region (By default it will be selected according to your Resource group selection.
- **Function Name** : Name of the function which you would like.

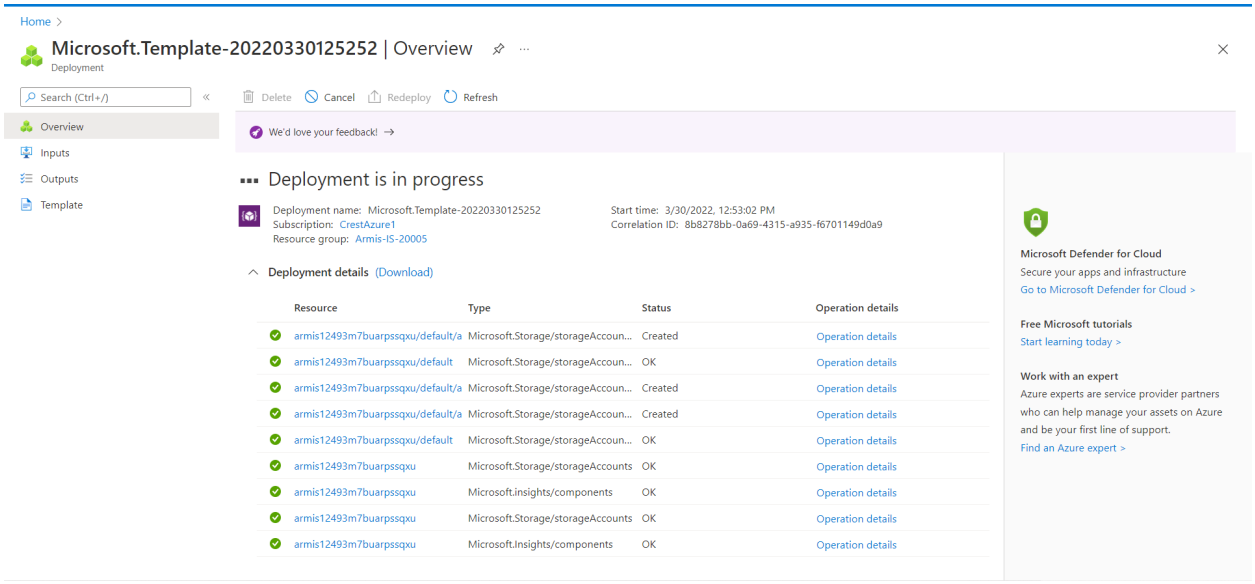
Note : You are able to input Maximum 11 characters in this field as a validation from sentinel.

- **Workspace ID** : Workspace ID of your Log Analytics Workspace in which you want to load ingested armis data.
- **Workspace Key** : Workspace Key of your Log Analytics Workspace in which you want to load ingested armis data.
- **Armis Secret Key** : API Secret Key required for Armis Authentication.
- **Armis URL** : Armis portal URL.
  (Example : **https://<armis-instance>.armis.com/api/v1/**)
- **Armis Alerts Table Name** : Alerts table name in which ingested Armis Alerts data will be loaded.
- **Avoid Duplicates** : If true, data duplication is avoided, but there might be a change of data loss
  If false, data may get duplicated

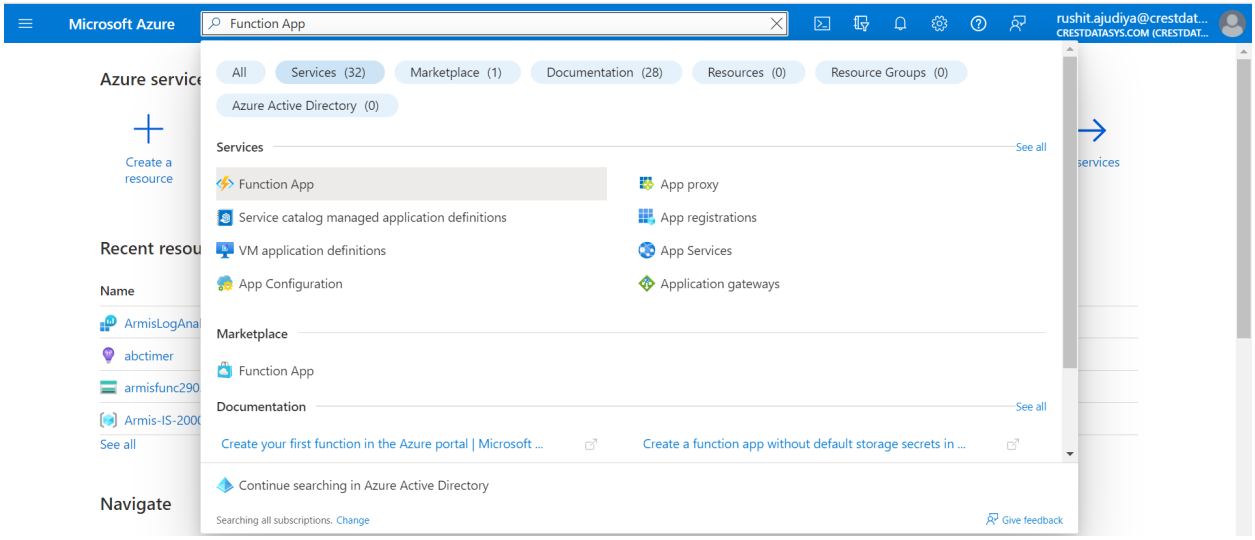
11. It will validate the Armis Data Connector deployment configuration content and if the validation is successful, it will show a **“Create”** button at the bottom of the screen.



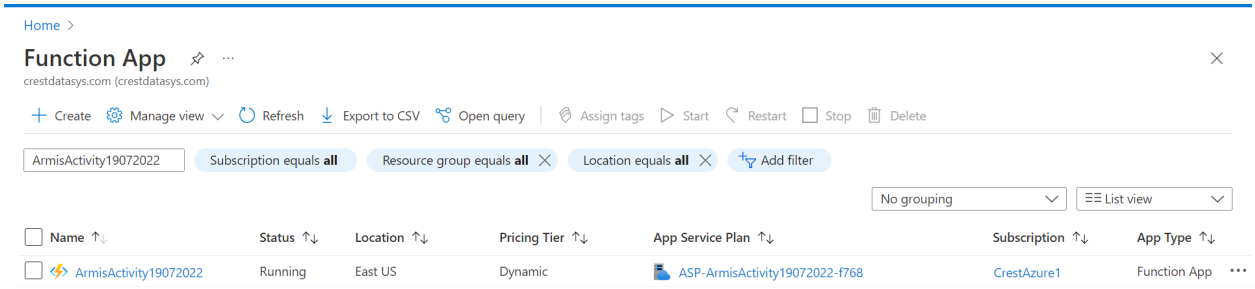
12. It will take some time to create the function app and storage. You can see the deployment status on the screen below.



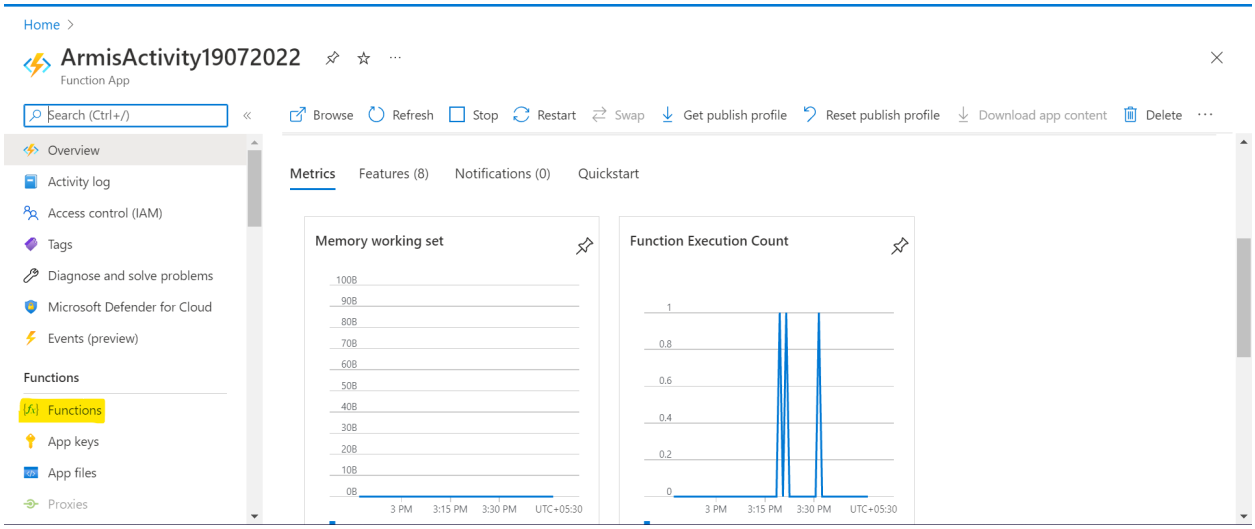
13. Now we need to see if the function app is properly deployed or not. And for that go to the search bar and search for the function app service in the azure portal.



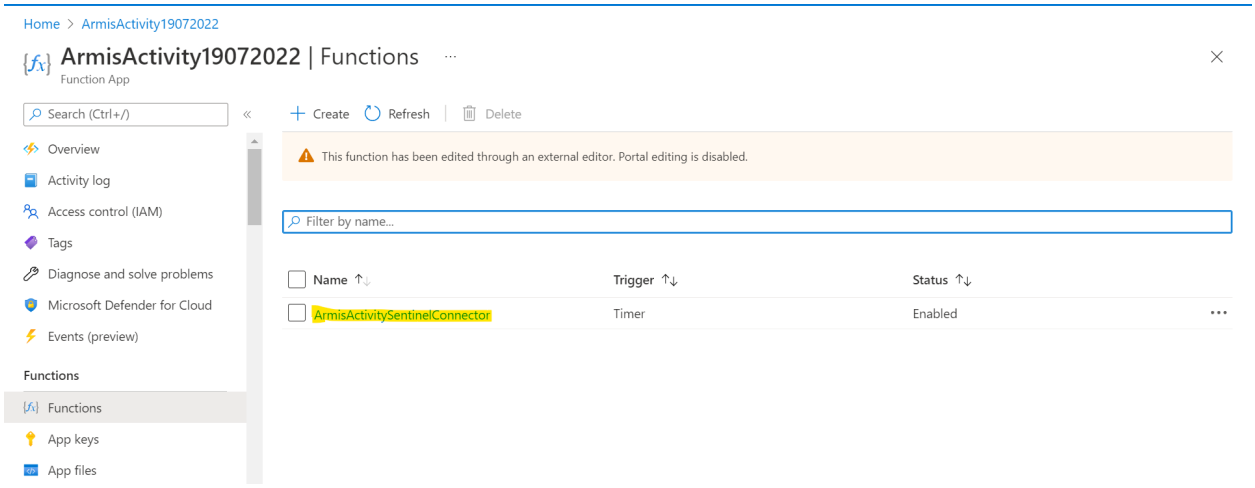
14. Now select the Function App(Data Connector) that you have deployed.
    Example : We have deployed a Function App(Data Connector) with the name **ArmisActivity19072022.**



15. Click on the Functions from the left panel.



16. Now select a Function App in the right panel, It will redirect you to the Function App.



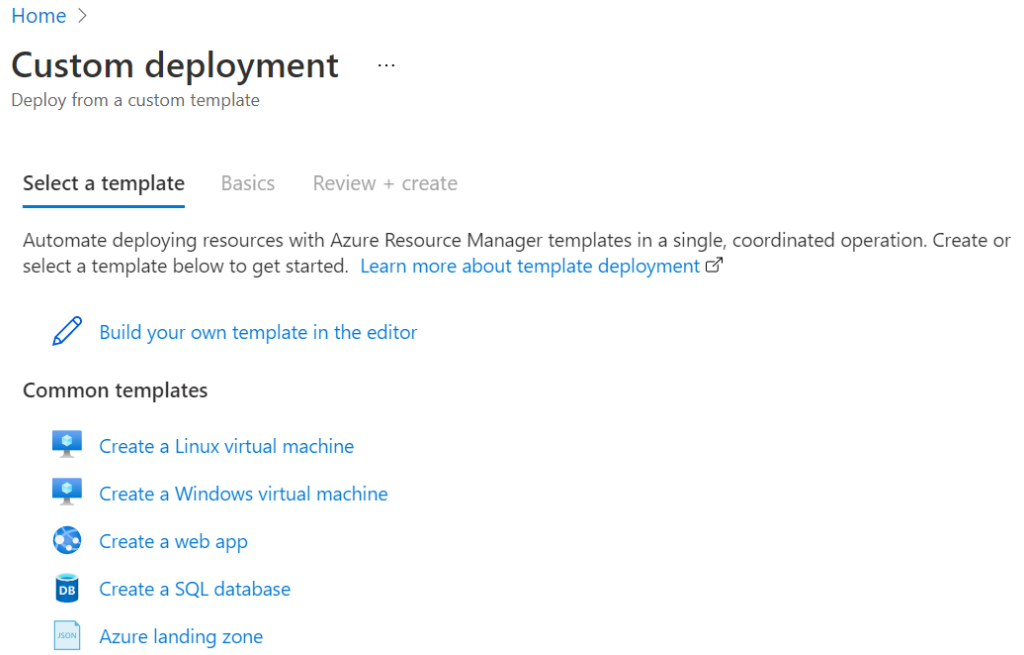
17. Now to test the Function App manually, click on the **code + test** and then click on **Test/Run.** It will pop up the window like in the image below.

18. Click on the **Run** button. If you get message **202 Accepted** then it runs successfully.

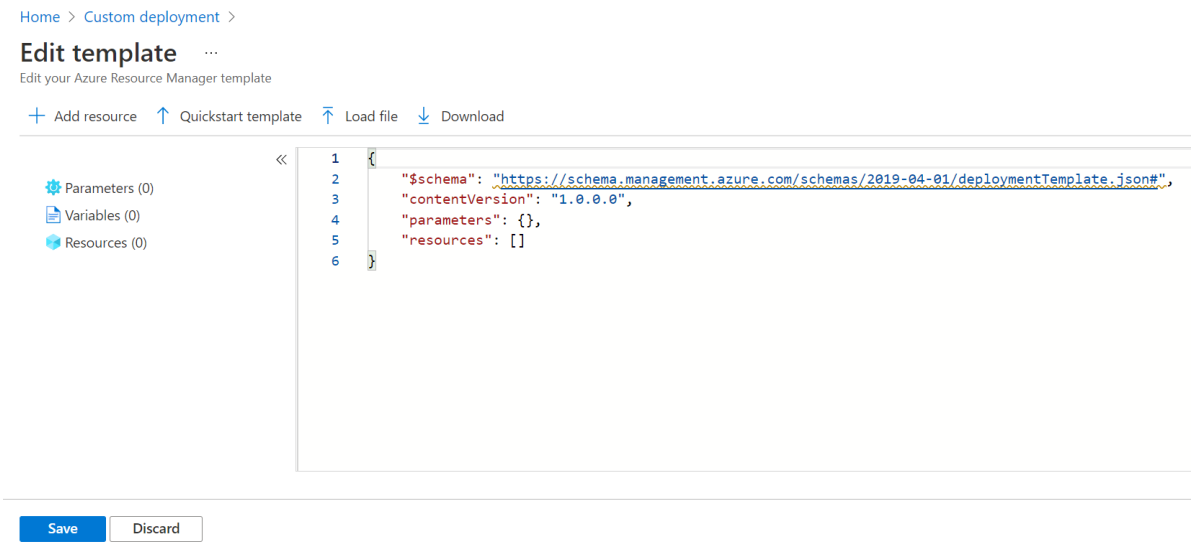
Note : To deploy Alert and Device function app you can follow the same step no 1 to 18

Armis Logic App Installation Guide

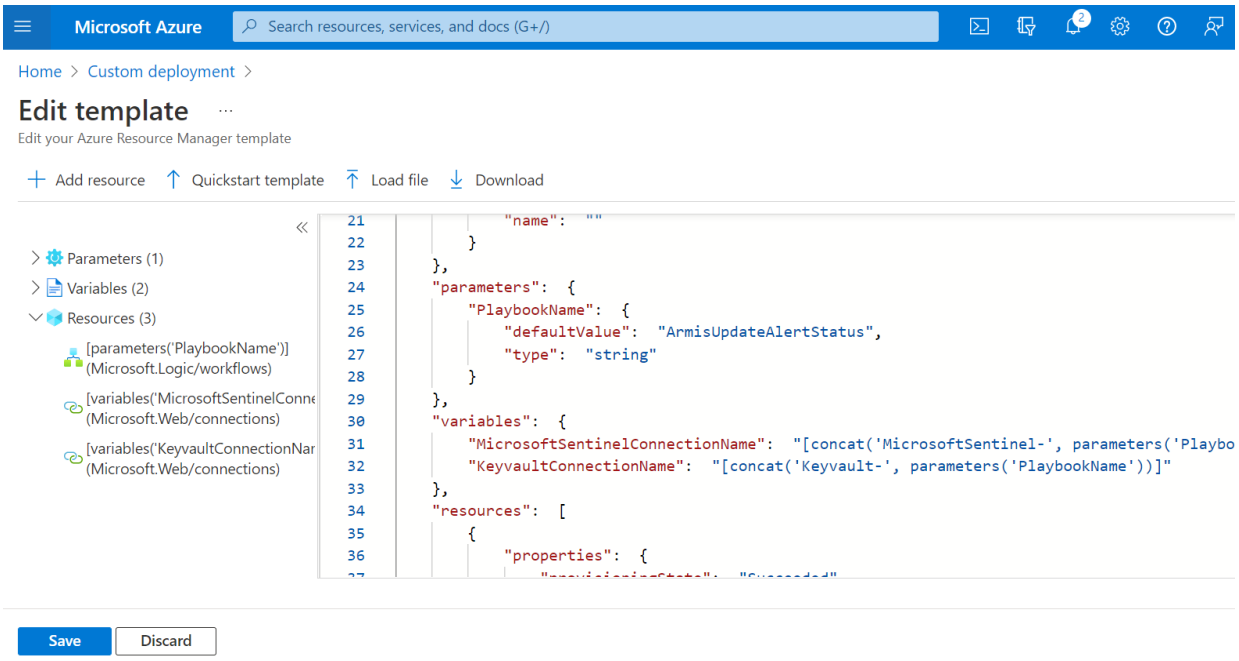
1. Login to Azure portal (https://portal.azure.com/#home) using the MS Azure credentials
2. **Logic App (Create a Logic App using an exported template)**
    a. Go to this link to import the Logic App using the template which has been provided to you
    b. Click on “Build your own template in the editor” to go to the Edit template screen



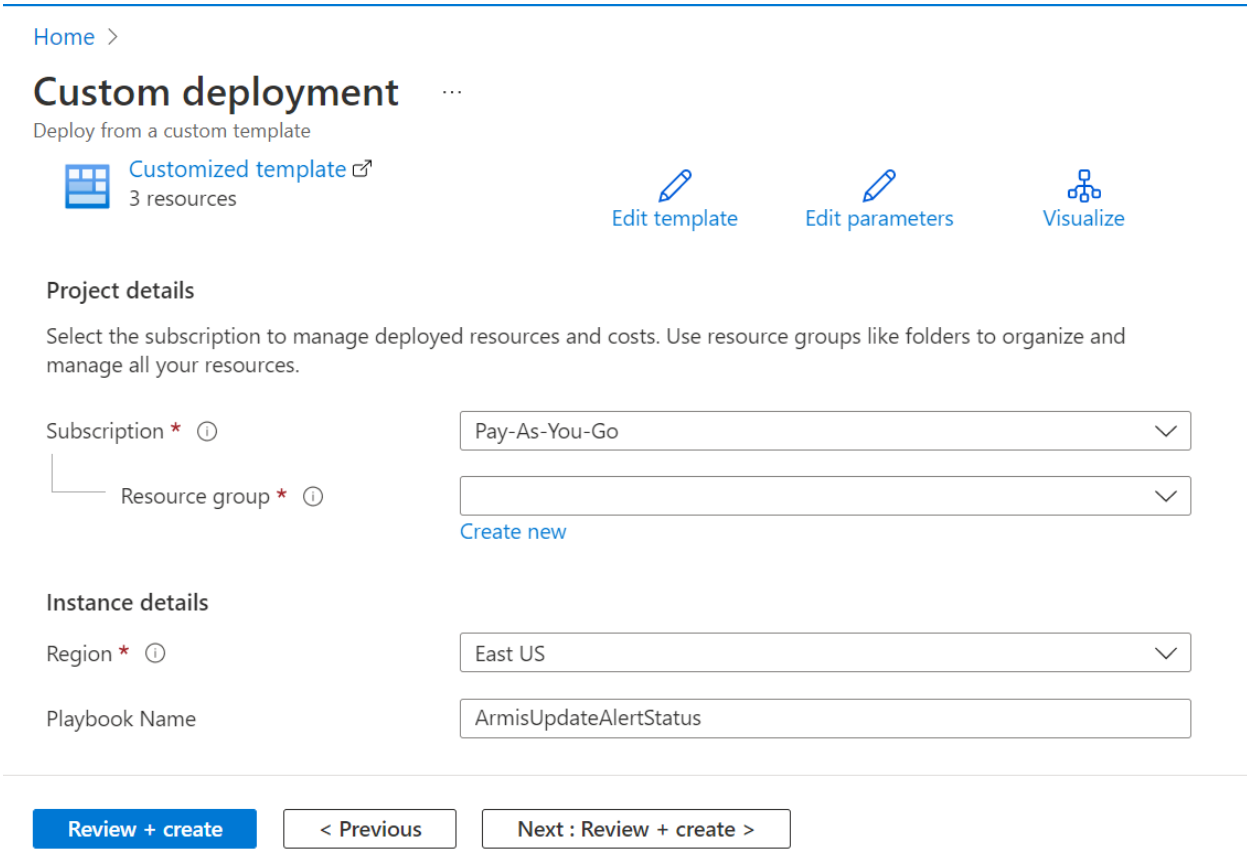
    c. Click on **“Load file”** and select **“armis_update_alert_status_playbook_template.json”** file which has been provided to you



d. Click on “Save” button and it will auto populate the data from the uploaded template

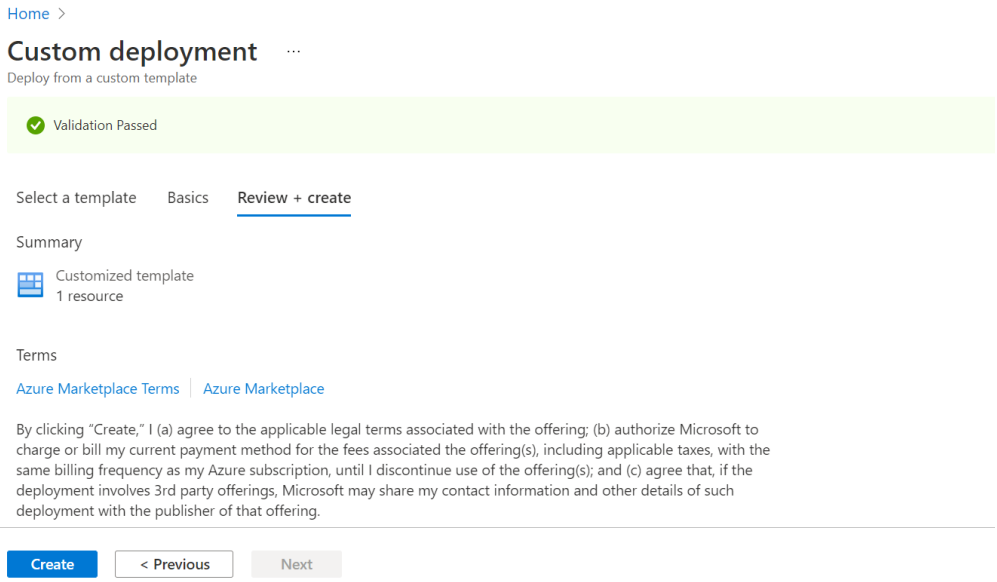


e. Select the appropriate “Resource Group”
f. Enter Playbook Name
g. Click on “Review + create” button



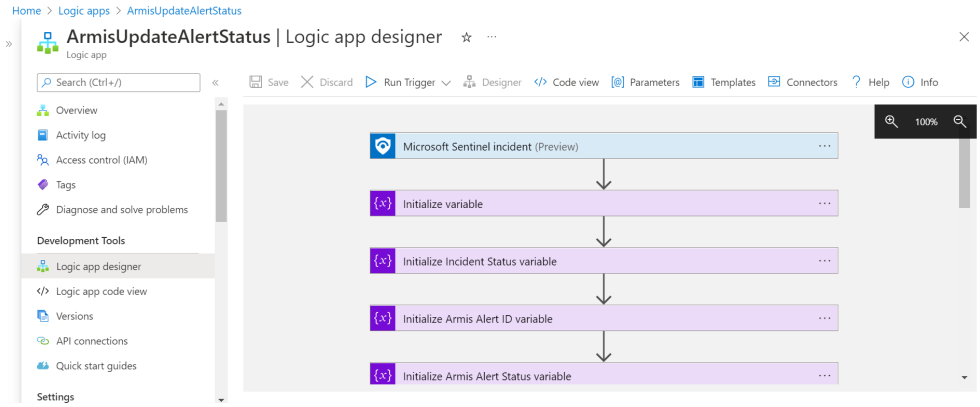
h. It will validate the Logic app content and if the validation is successful, it will show a “Create” button at the bottom of the screen.

i. Click on “Create” button

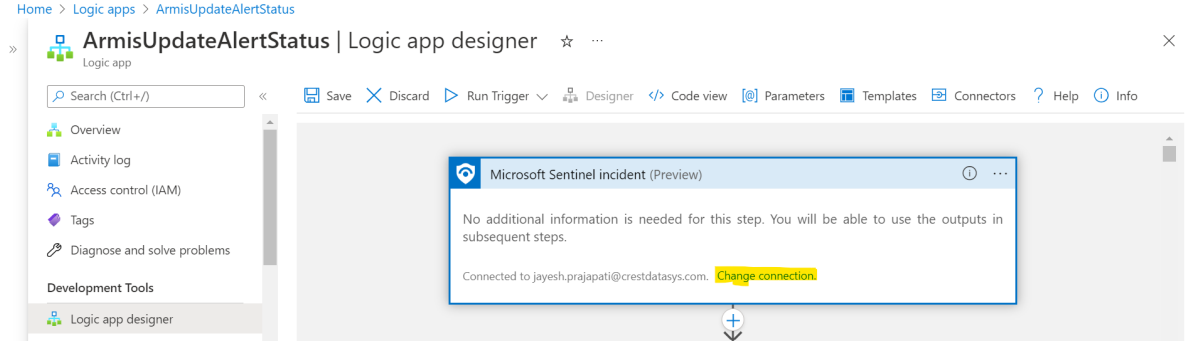


j. The above step will deploy a Armis Update Alert Status Logic App(Playbook) on Azure Portal

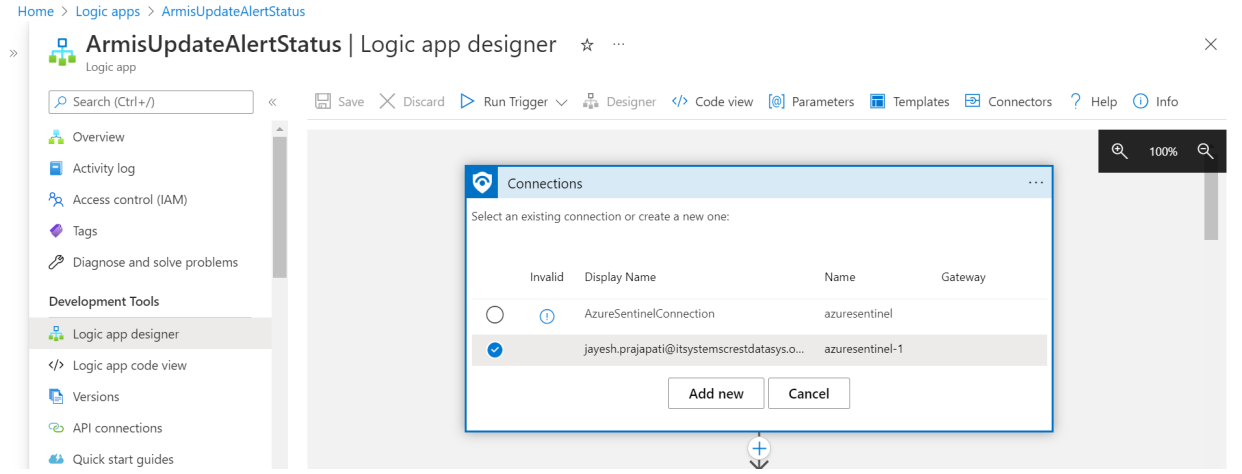
k. Finally logic app template is deployed successfully and it will be visible in “Logic App” service in azure portal



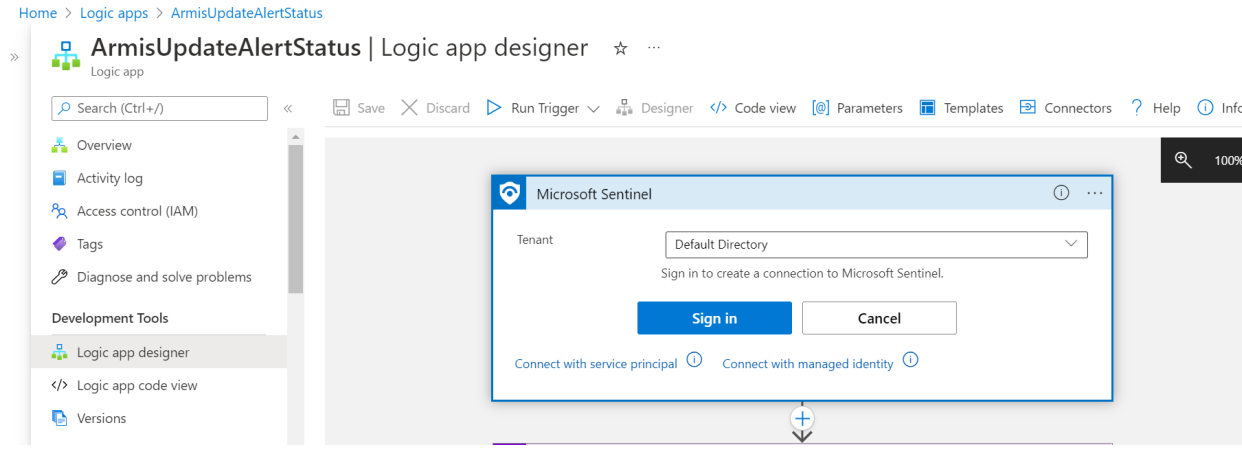
l. Expand “Microsoft Sentinel incident (Preview)” trigger and click on “Change connection”



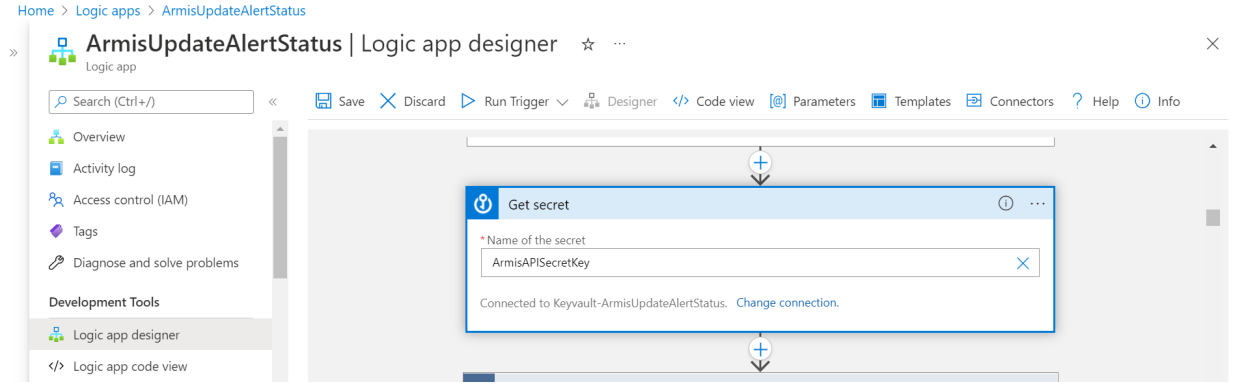
m. Click on “Add new” button from connection window



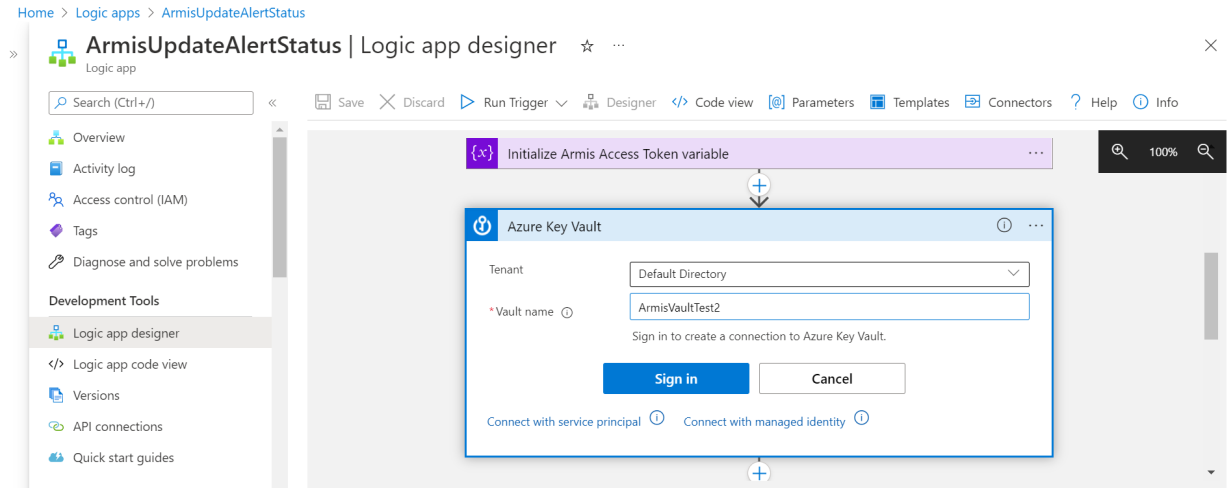
n. Click on “Sign in” button and it will pop-up once dialogue



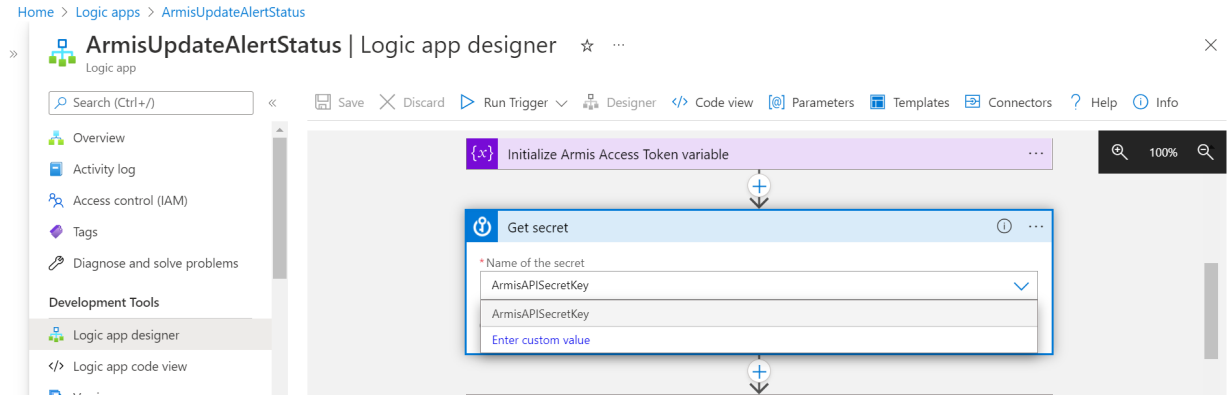
o. Sign in with your azure portal credentials
p. After successful sign in click on “Save” button to save changes
q. Now perform **step 3** to create key vault
r. After successfully creating Key Vaults and Secret go to logic app again and expand “Get Secret” action



s. Click on “Change connection” and click on “Add new” button from connections screen
t. Now select “Tenant” and provide “Vault name” which you have created in step 3

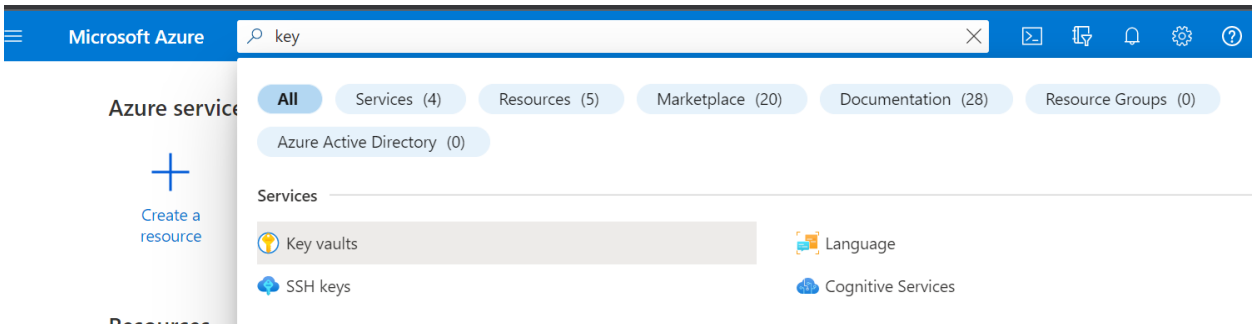


u. Click on “Sign in” and provide your azure portal credentials in open pop-up window
v. After that it will show all the secret available in the provided key vault

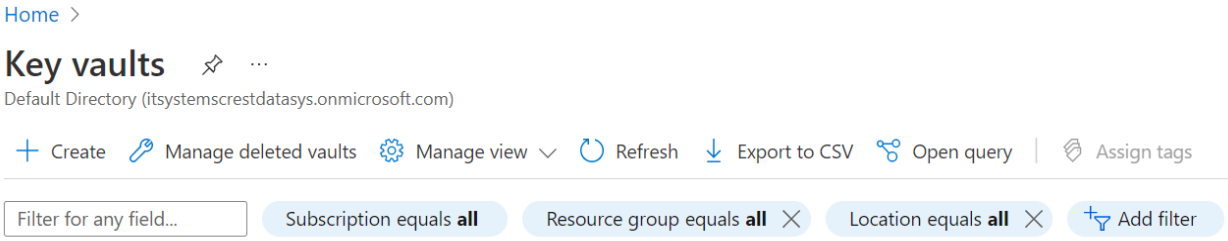


w. Select the secret in which you have provided Armis Secret keys which will be used for Authentication
x. Click on “Save” button to save the changes in the playbook

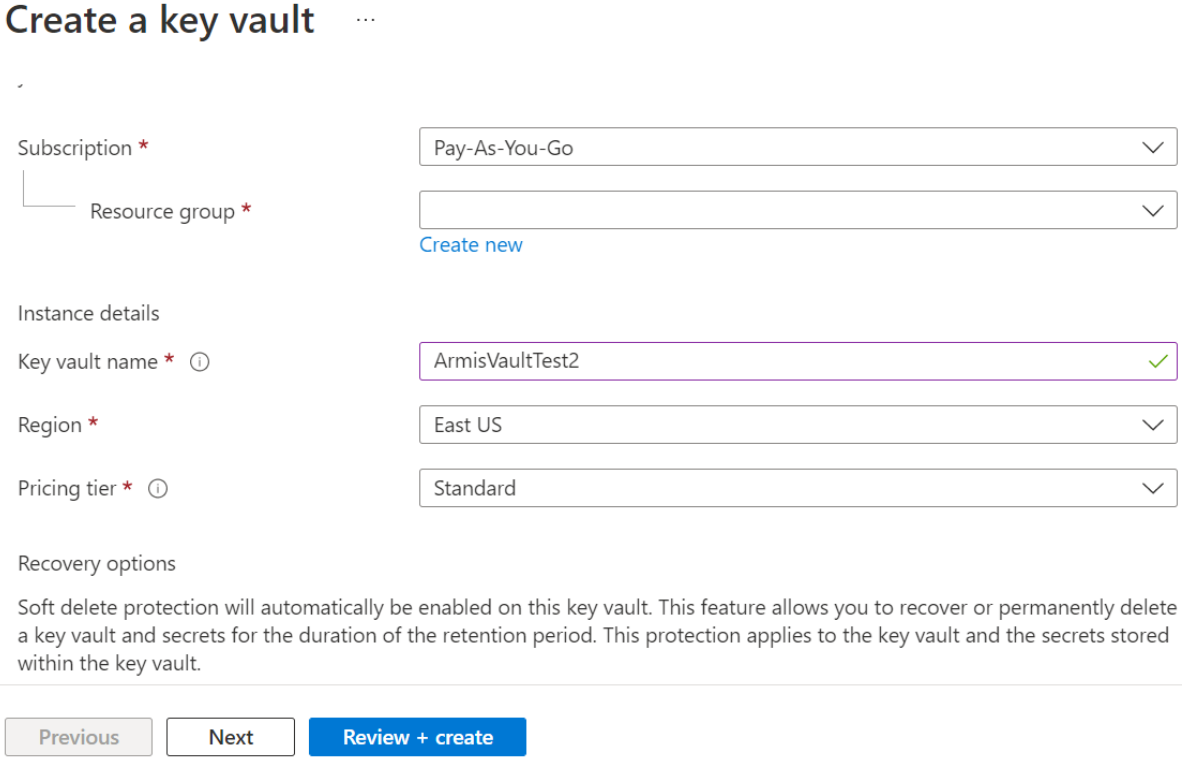
3. **Create Key Vault**
   a. Go to “Key vaults”



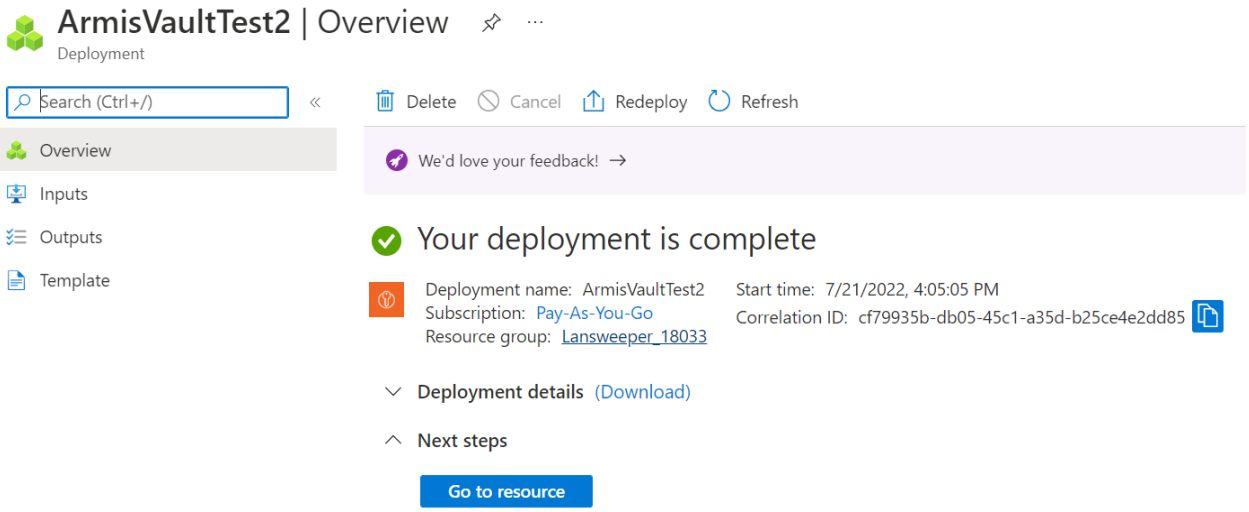
   b. Click on “Create” button



   c. Select appropriate Resource group from dropdown
   d. Enter Key vault name which globally unique
   e. Select region of your resource group
   f. Click on “Review + create” button

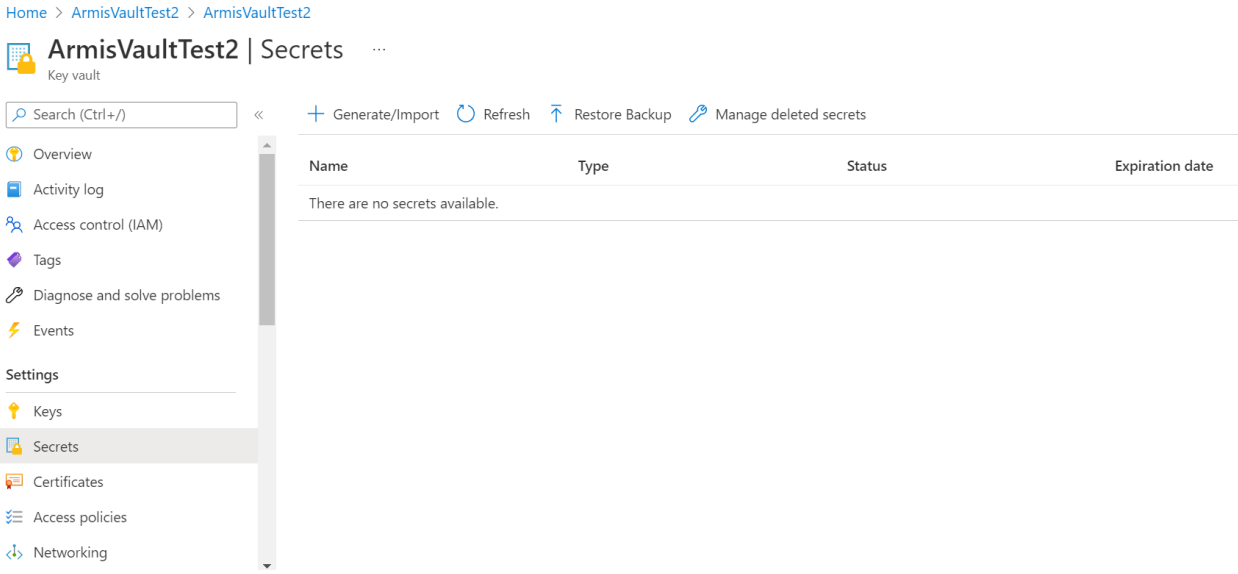


g. The above step will create Key Vault

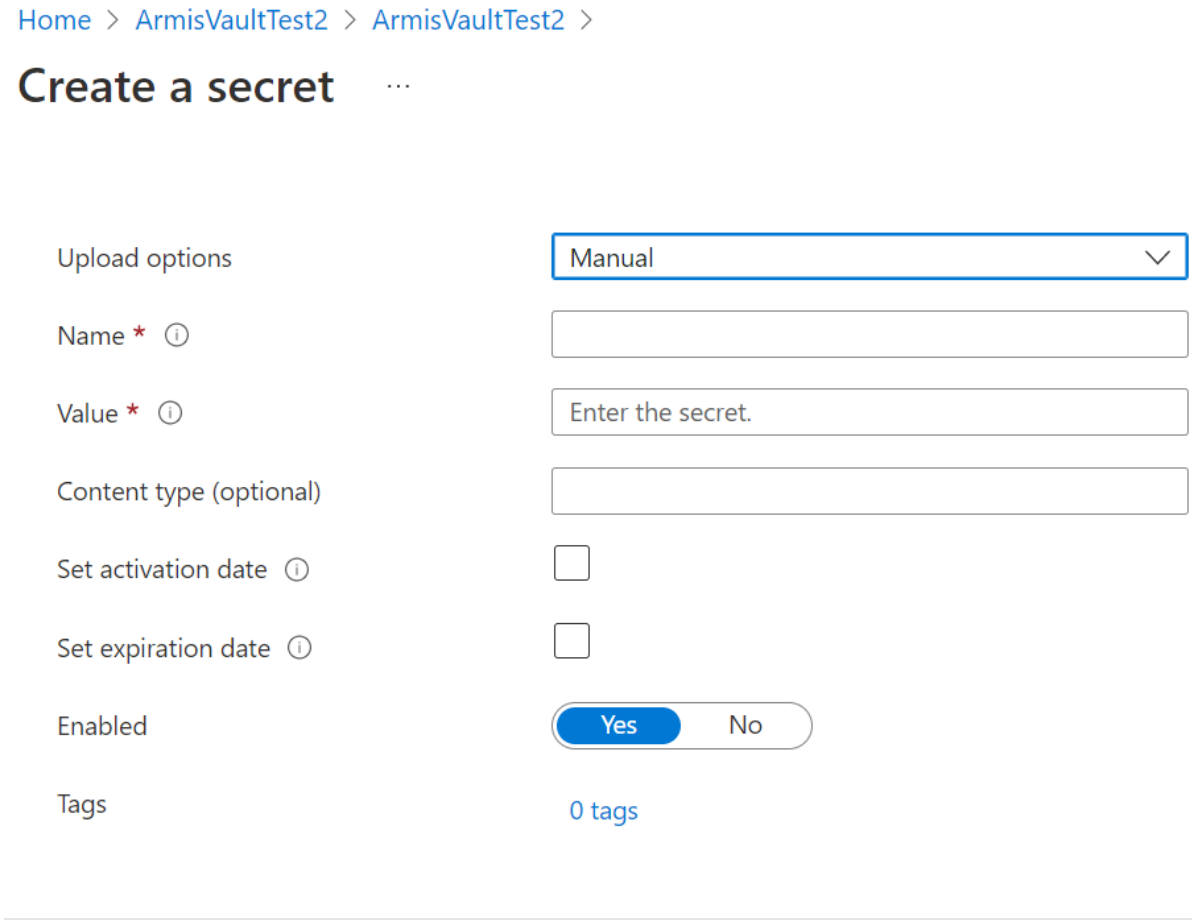


h. Click on “Go to resource” button

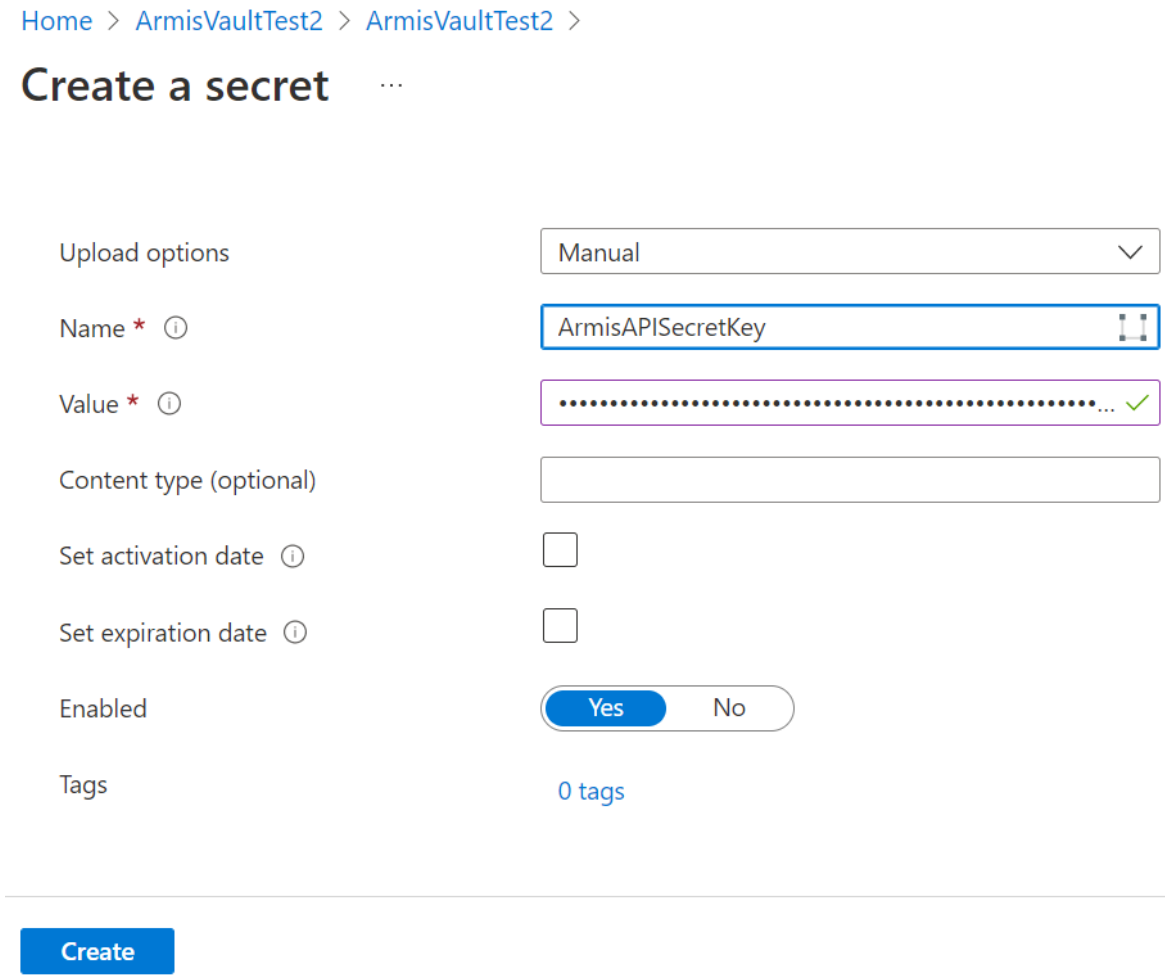
i. Click on “Secrets” from left menu



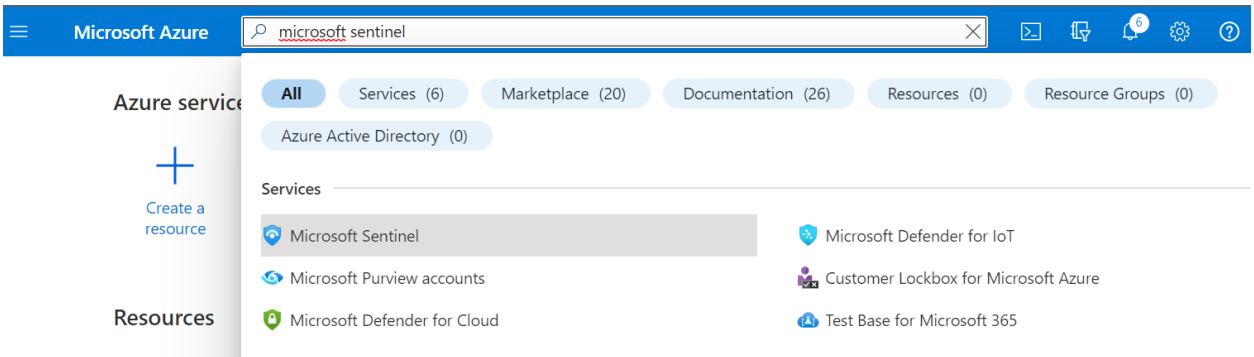
j. Click on “Generate/Import” button



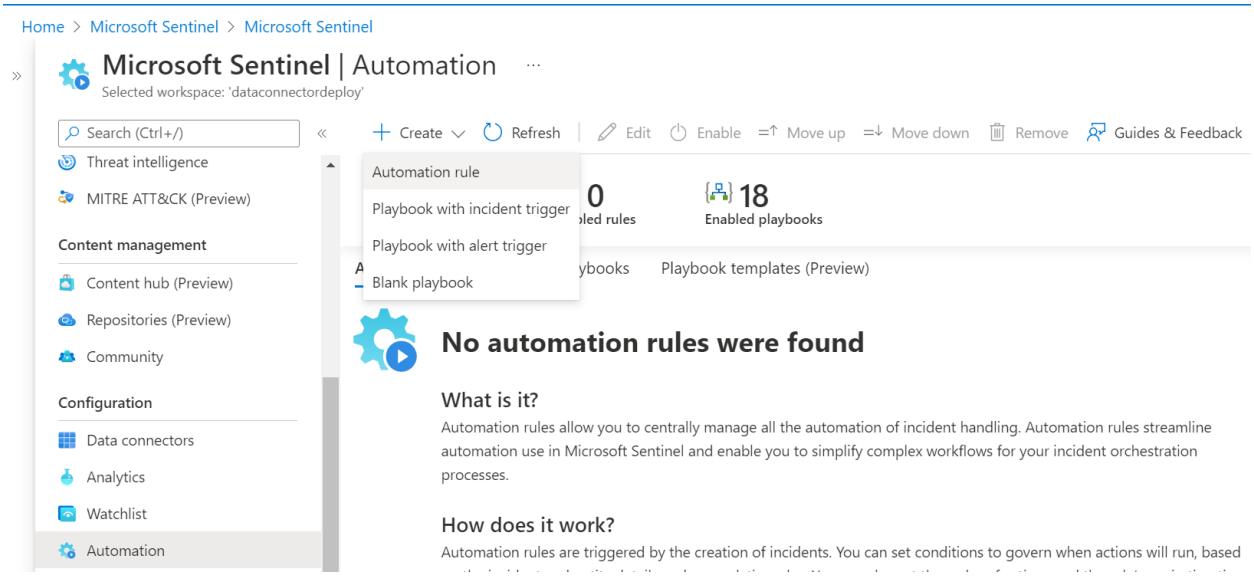
k. Provide Name of secret
l. Provide Value for secret
m. Click on create button



4. **Create Automation Rule which Runs Logic App**
   a. Go to “Microsoft Sentinel” service



   b. Go to the Sentinel Workspace in which you want to create Automation Rule
   c. Go to “Automation” from the left menu by clicking on it
   d. From the **Automation** blade in the Microsoft Sentinel navigation menu, select **Create** from the top menu and choose the **Automation rule**.

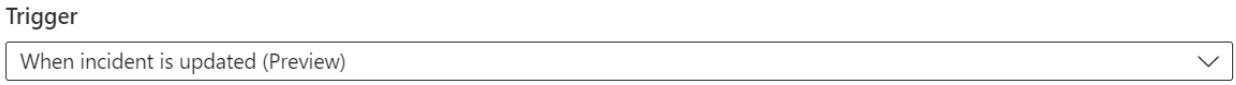


e. The Create new automation rule panel opens. Enter a name for your rule.

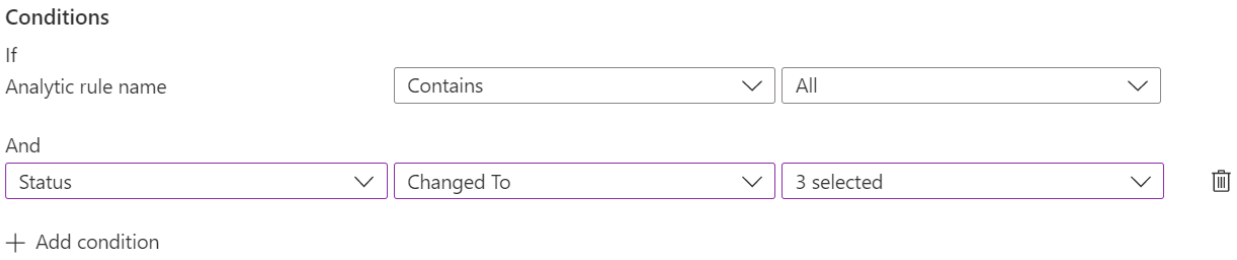
f. Enter below information

- **Automation Rule Name :** Enter automation rule name

- **Trigger :** From the Trigger drop-down, select “When incident is updated (Preview)”

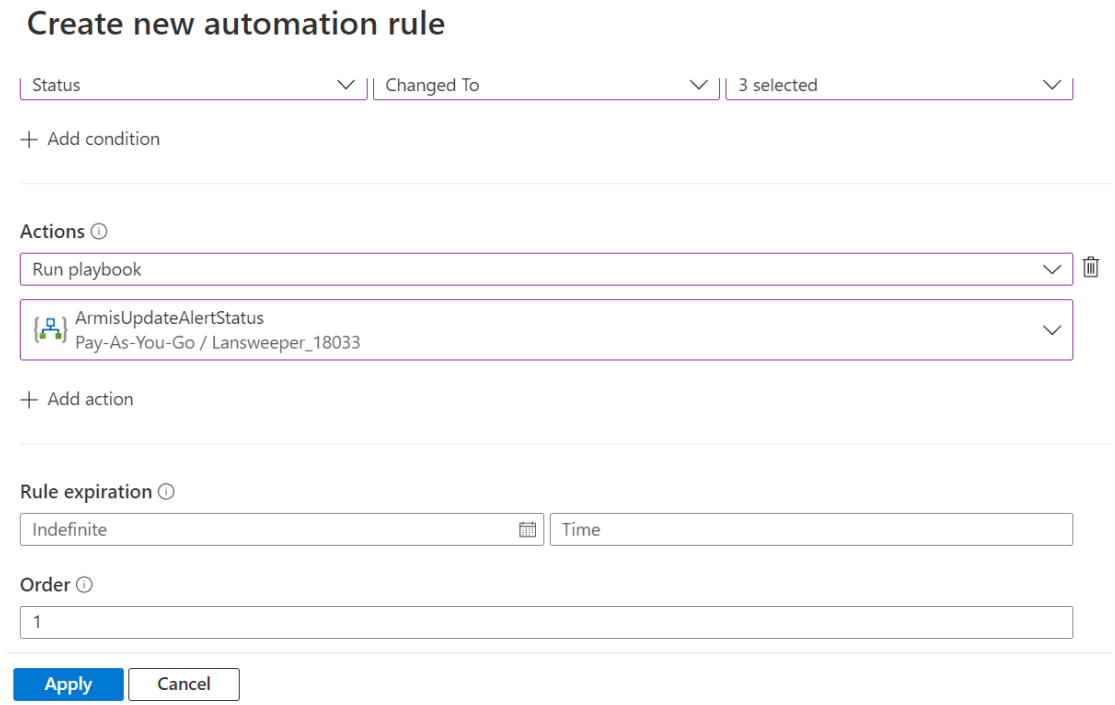


- **Add conditions :**
  - If Analytic rule name :
    Select **“Contains”** and **“All”**
  - Click on + Add Condition :
    Select from dropdown value as **“Status”**, **“Changed To”** and **“New, Active, Closed”**

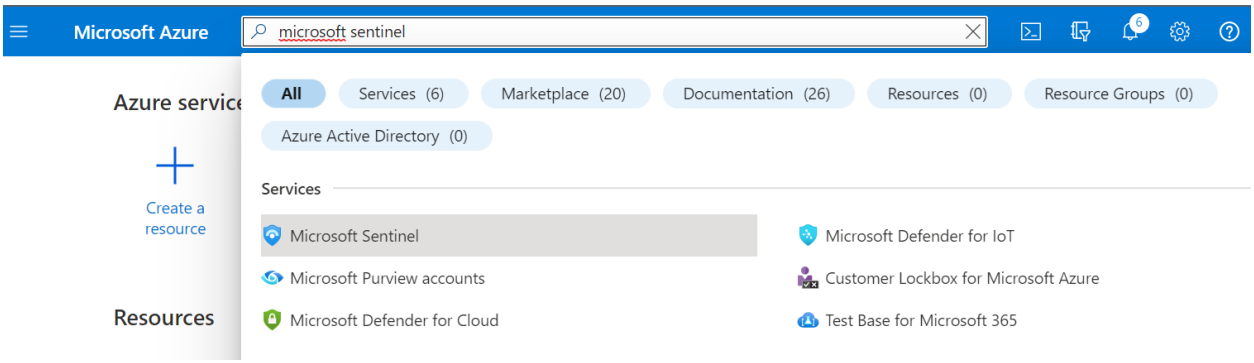


- **Add actions :**
  - Choose “Run playbook” from dropdown
  - Select Playbook which we want to execute when the automation rule condition gets true.

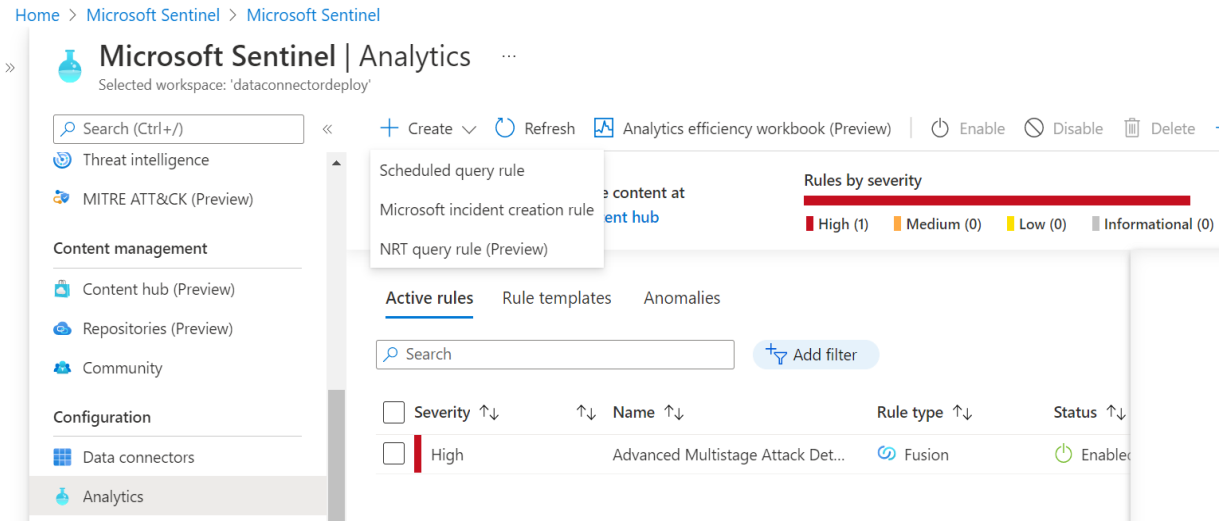
g. Click on “Apply” button to create automation rule



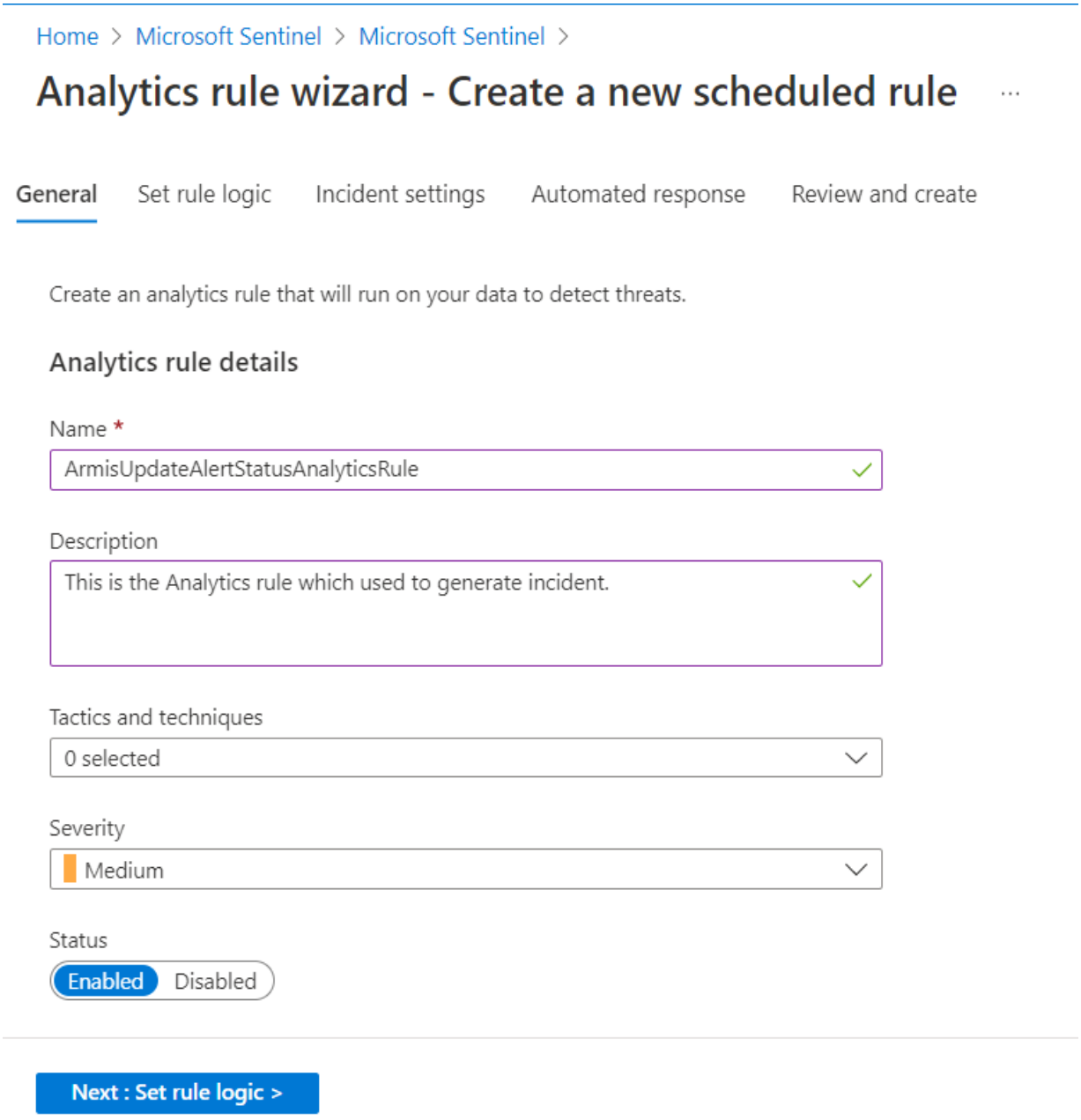
5. **Create Analytics Rule**
   a. Go to “Microsoft Sentinel” service



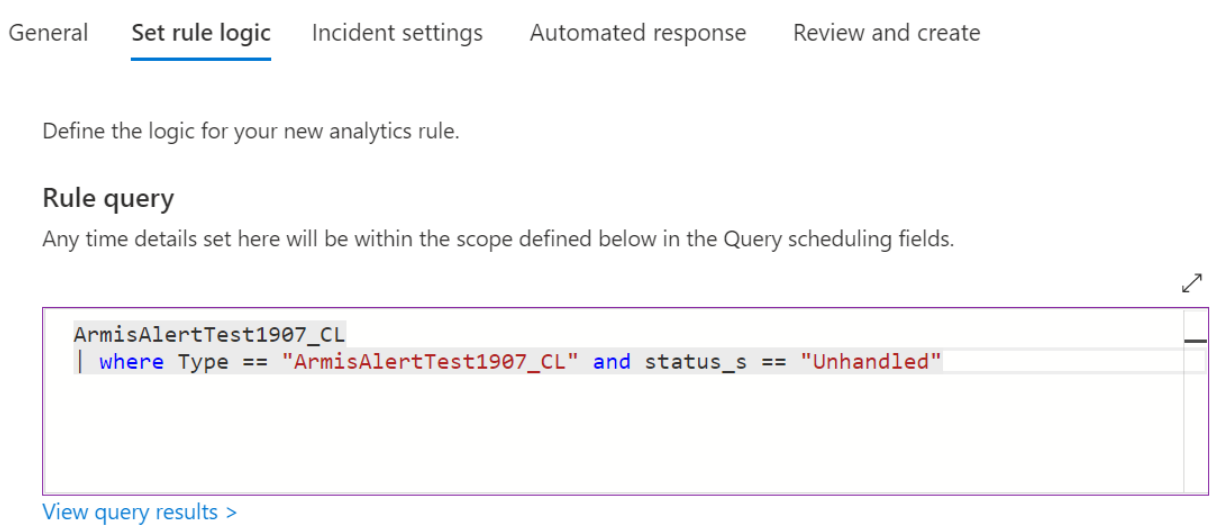
   b. Go to the Sentinel Workspace in which you want to create Analytics Rule
   c. Go to **“Analytics”** from the left menu by clicking on it
   d. From the **Analytics** blade in the Microsoft Sentinel navigation menu, select **Create** from the top menu and choose the “**Scheduled query rule”**.



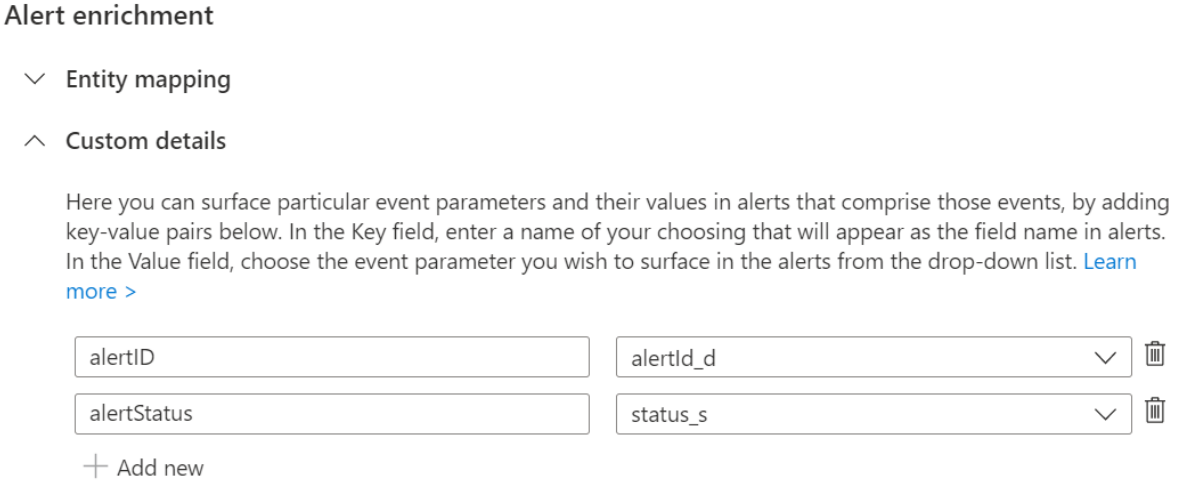
   e. Analytics rule wizard - **General tab**
      - **Name :** Provide unique Analytics Rule Name
      - **Description :** Provide Analytics Rule description
      - **Other fields are not relevant to our use case so keep it as it is**



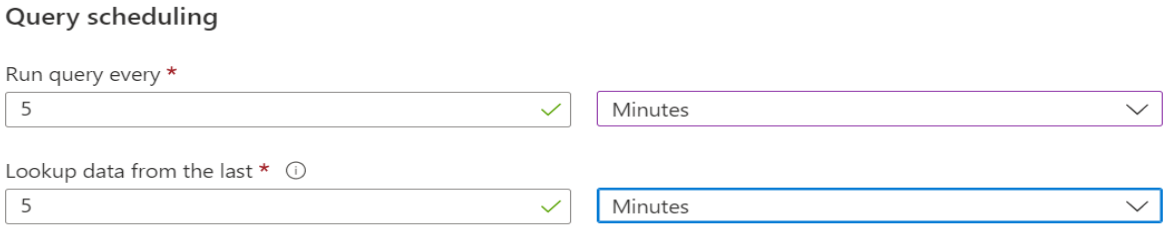
f. Define the rule query logic and configure settings
   - **Rule Query :** Queries are written in Kusto Query Language (KQL)



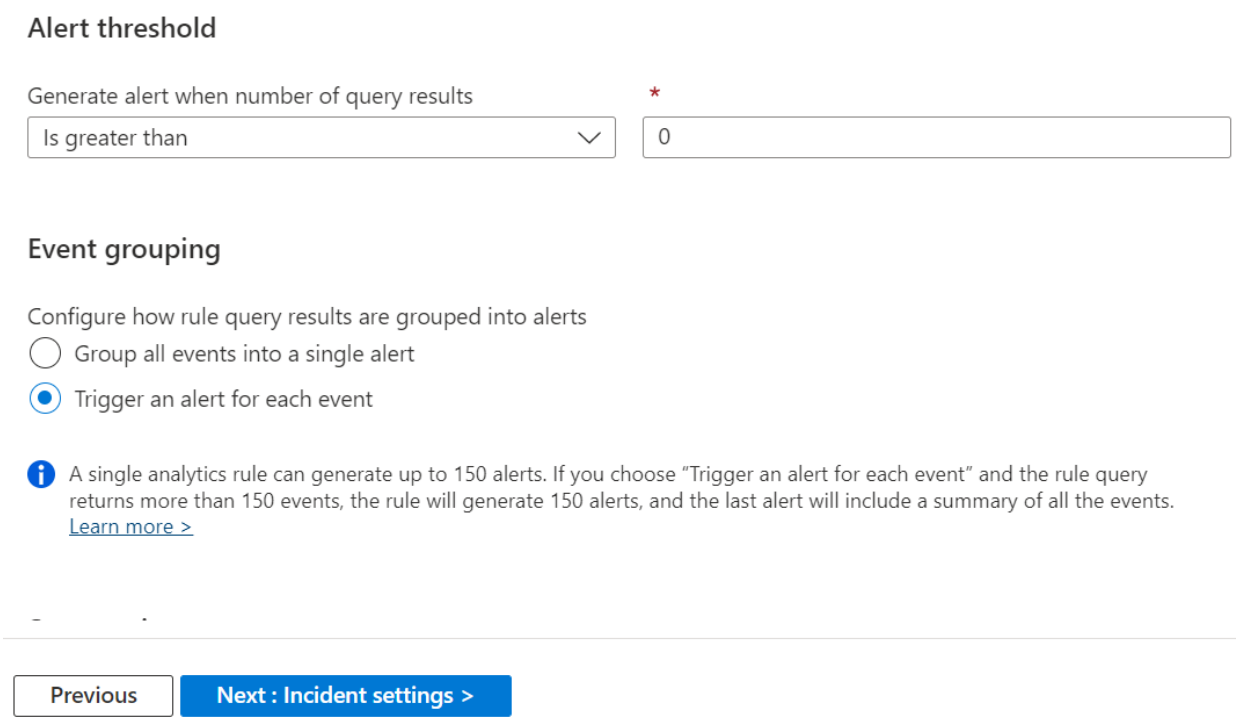
- **Custom details :** section to extract event data items from your query and surface them in the alerts produced by this rule, giving you immediate event content visibility in your alerts and incidents.



- **Query scheduling :** Set as mentioned in below image

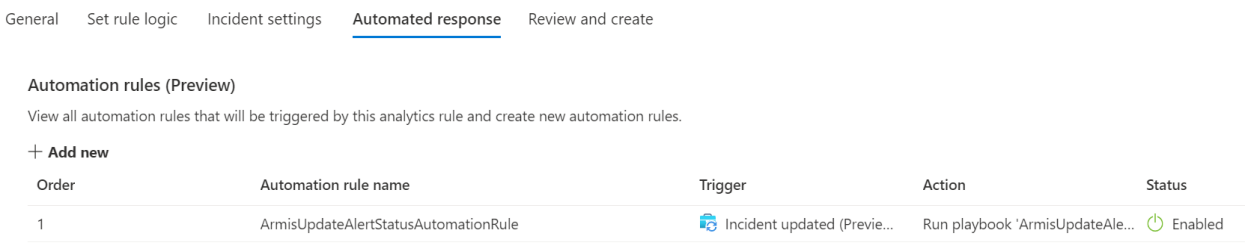


- Event grouping :
  Select **“Trigger an alert for each event”** radio button

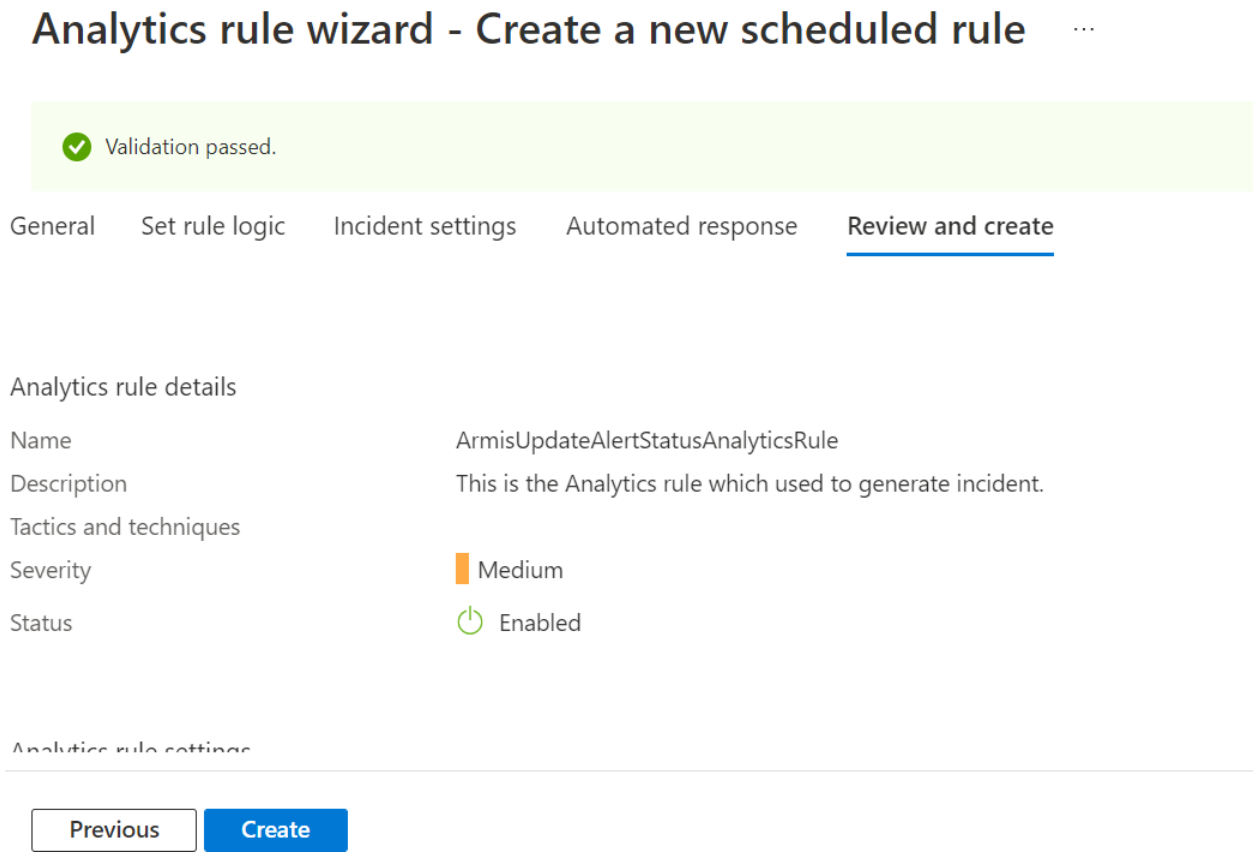


g. Configure the incident creation settings
   - No need to set/update any setting in this tab.

h. Set automated responses and create the rule
   - **Automation rules(Preview) :** The grid displayed under automation rules(Preview) shows the automation rules that already apply to this analytics rule



i. Click on “Next : Review >” button
j. It will validate the Analytics rule content and if the validation is successful, it will show a “Create” button at the bottom of the screen.
k. Click on “Create” button



I. The above step will create an Analytics rule.