End-to-end security with Azure Synapse Analytics

Wide World Importers is host to a plethora of data coming from many disparate sources. The idea of bringing all of their data together into Azure Synapse Analytics for them to query, gain insights, and consume in ways they have never done before is exhilarating! As much as it is an exciting game-changer for this business, it opens up a large amount of surface area for potential attack. Security must be established in the forefront at the time of design of this solution.

This lab will guide you through several security-related steps that cover an end-to-end security story for Azure Synapse Analytics. Some key take-aways from this lab are:

1. Leverage Azure Key Vault to store sensitive connection information, such as access keys and passwords for linked services as well as in pipelines.

2. Introspect the data that is contained within the SQL Pools in the context of potential sensitive/confidential data disclosure. Identify the columns representing sensitive data, then secure them by adding column-level security. Determine at the table level what data should be hidden from specific groups of users then define security predicates to apply row level security (filters) on the table. If desired, you also have the option of applying Dynamic Data Masking to mask sensitive data returned in queries on a column by column basis.

Resource naming throughout this lab

For the remainder of this guide, the following terms will be used for various ASA-related resources (make sure you replace them with actual names and values):

Azure Synapse Analytics Resource	**To be referred to**
Workspace resource group	`WorkspaceResourceGroup`
Workspace / workspace name	`Workspace`
Primary Storage Account	`PrimaryStorage`
Default file system container	`DefaultFileSystem`
SQL Pool	`SqlPool01`
SQL Serverless Endpoint	`SqlServerless01`
Active Directory Principal of New User	`user@domain.com`
SQL username of New User	`newuser`
Azure Key Vault	`KeyVault01`
Azure Key Vault Private Endpoint Name	`KeyVaultPrivateEndpointName`
Azure Subscription	`WorkspaceSubscription`
Azure Region	`WorkspaceRegion`

Exercise 1 - Securing Azure Synapse Analytics supporting infrastructure

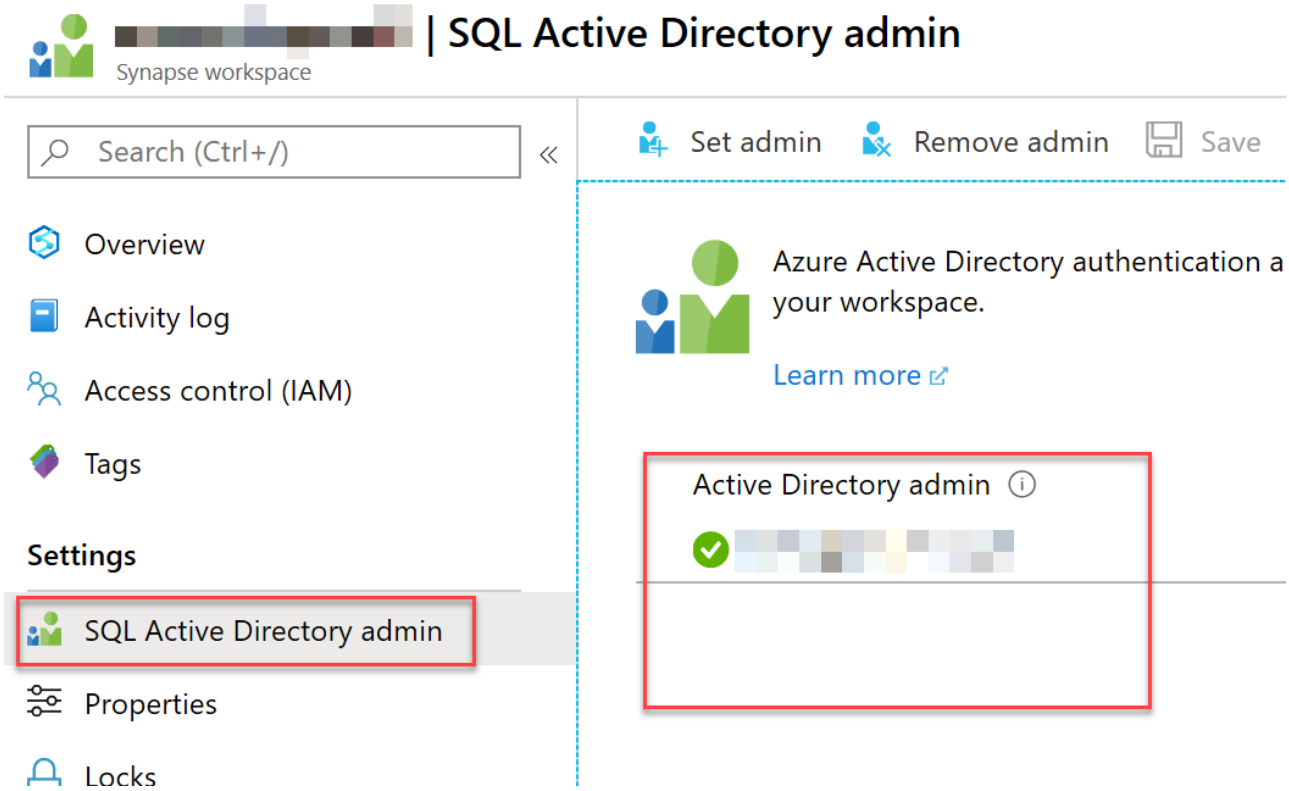
Azure Synapse Analytics (ASA) is a powerful solution that handles security for many of the resources that it creates and manages. In order to run ASA, however, some foundational security measures need to be put in place to ensure the infrastructure that it relies upon is secure. In this exercise, we will walk through securing the supporting infrastructure of ASA.

Task 1 - Observing the SQL Active Directory admin

The SQL Active Directory Admin can be a user (the default) or group (best practice so that more than one user can be provided these permissions) security principal. The principal assigned to this will have administrative permissions to the SQL Pools contained in the workspace.

1. In the **Azure Portal**, browse to your `L400` resource group and from the list of resources open your Synapse workspace (do not launch Synapse Studio).

2. From the left menu, select **SQL Active Directory admin** and observe who is listed as a SQL Active Directory Admin. Is it a user or group?

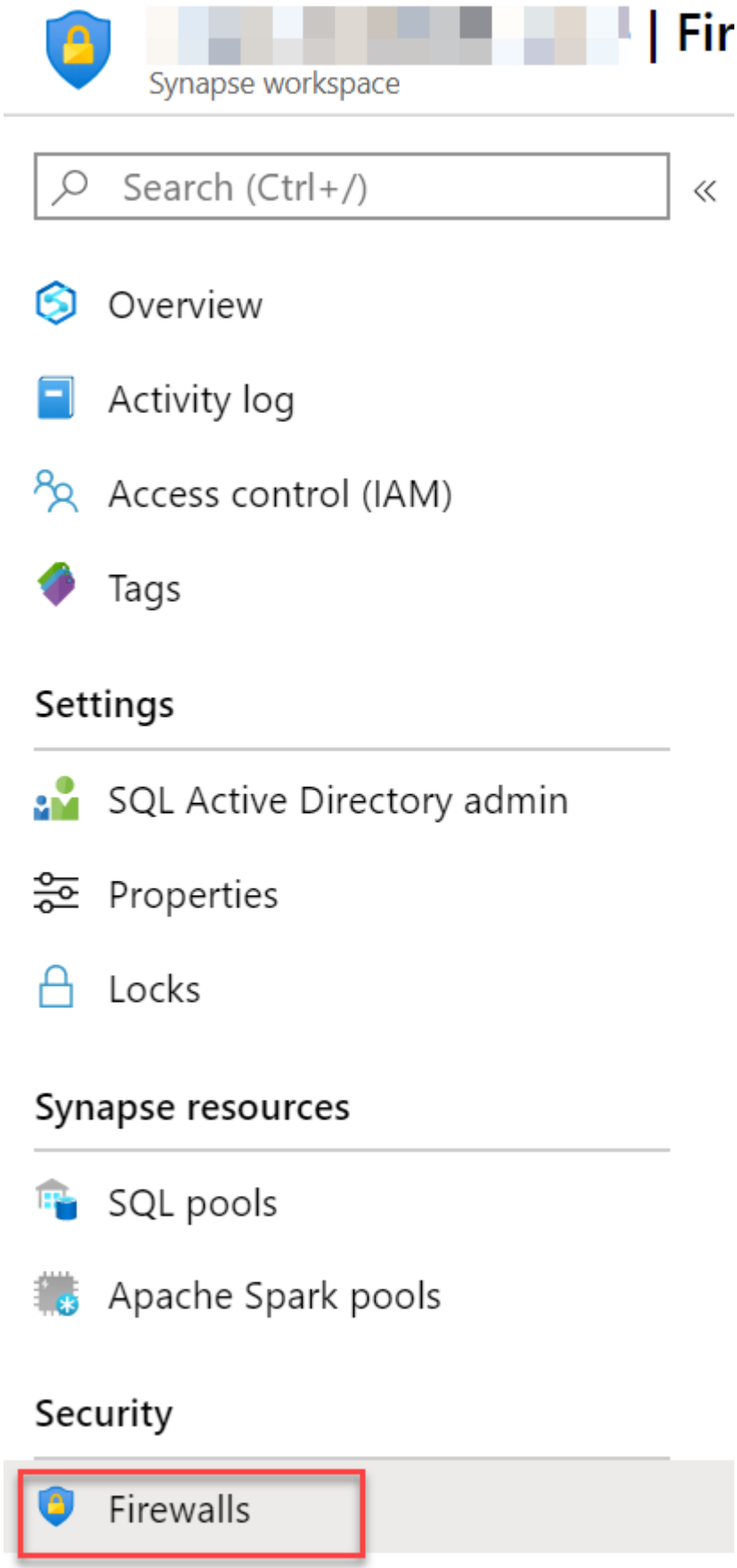


Task 2 - Manage IP firewall rules

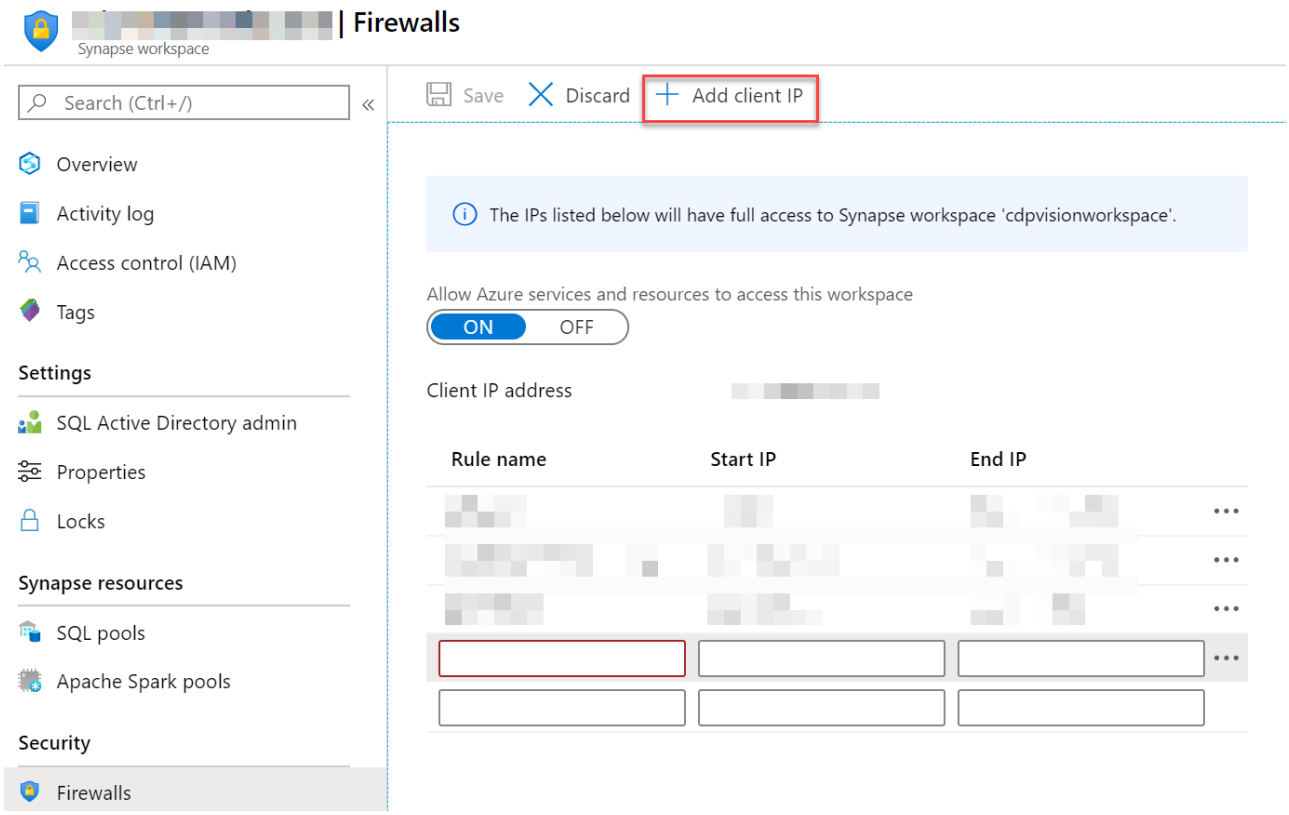
Having robust Internet security is a must for every technology system. One way to mitigate internet threat vectors is by reducing the number of public IP addresses that can access the Azure Synapse Analytics Workspace through the use of IP firewall rules. The Azure Synapse Analytics workspace will then delegate those same rules to all managed public endpoints of the workspace, including those for SQL pools and SQL Serverless endpoints.

1. In the **Azure Portal**, open the Synapse workspace (do not launch Studio).

2. From the left menu of the **Azure Synapse Analytics** page, select **Firewalls**.



3. Notice that an IP Firewall rule of `Allow All` has already been created for you in the lab environment. If you wanted to add your specific IP address you would instead select **+ Add Client IP** from the taskbar menu (you should not do this in this lab).



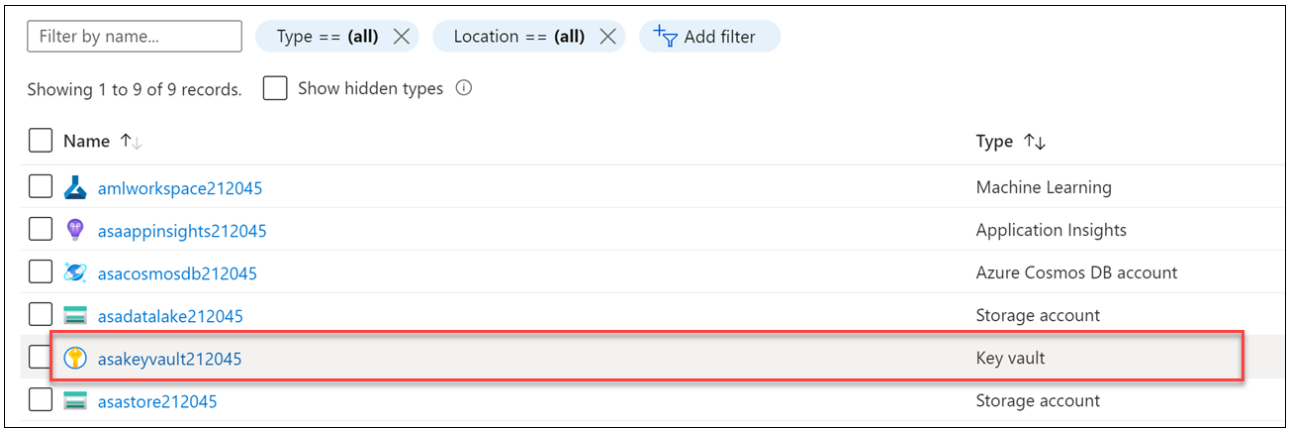
> **Note**: When connecting to Synapse from your local network, certain ports need to be open. To support the functions of Synapse Studio, ensure outgoing TCP ports 80, 443, and 1143, and UDP port 53 are open.

Exercise 2 - Securing the Azure Synapse Analytics workspace and managed services

Task 1 - Managing secrets with Azure Key Vault

When dealing with connectivity to external data sources and services, sensitive connection information such as passwords and access keys should be properly handled. It is recommended that this type of information be stored in an Azure Key Vault. Leveraging Azure Key Vault not only protects against secrets being compromised, it also serves as a central source of truth; meaning that if a secret value needs to be updated (such as when cycling access keys on a storage account), it can be changed in one place and all services consuming this key will start pulling the new value immediately. Azure Key Vault encrypts and decrypts information transparently using 256-bit AES encryption, which is FIPS 140-2 compliant.

1. In the **Azure Portal**, open the `Synapse-WS-L400-NNNNNN` resource group and from the list of resources and select the **Key vault** resource.



2. From the left menu, under Settings, select **Access Policies**.

3. Observe that Managed Service Identity (MSI) representing your Synapse workpace (it has a name similar to `asaworkspaceNNNNNN`) has already been listed under Application and it has 4 selected Secret Permissions.

4. Select the drop-down that reads `4 selected` under `Secret Permissions`, observe that Get (which allows your workspace to retrieve the values of secrets from Key Vault) and List (which allows your workspace to enumerate secrets) are set.

Task 2 - Use Azure Key Vault for secrets when creating Linked Services

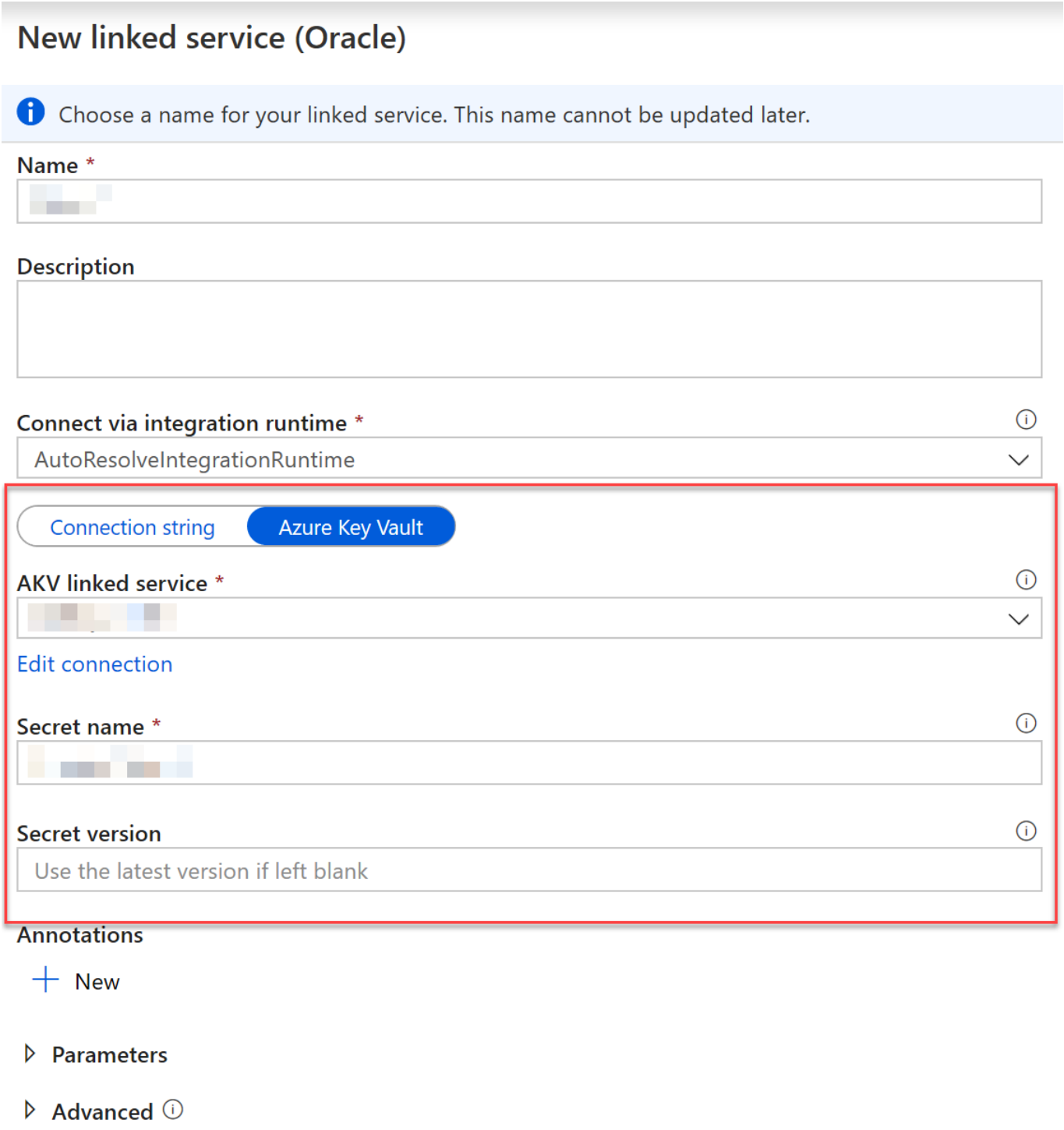
Linked Services are synonymous with connection strings in Azure Synapse Analytics. Azure Synapse Analytics linked services provides the ability to connect to nearly 100 different types of external services ranging from Azure Storage Accounts to Amazon S3 and more. When connecting to external services, having secrets related to connection information is almost guaranteed. The best place to store these secrets is the Azure Key Vault. Azure Synapse Analytics provides the ability to configure all linked service connections with values from Azure Key Vault.

In order to leverage Azure Key Vault in linked services, you must first add `asakeyvaultXX` as a linked service in Azure Synapse Analytics.

1. In **Azure Synapse Studio** (https://web.azuresynapse.net/), select **Manage** from the left menu.

2. Beneath **External Connections**, select **Linked Services**, observe that a Linked Service pointing to your Key Vault has been provided in the environment.

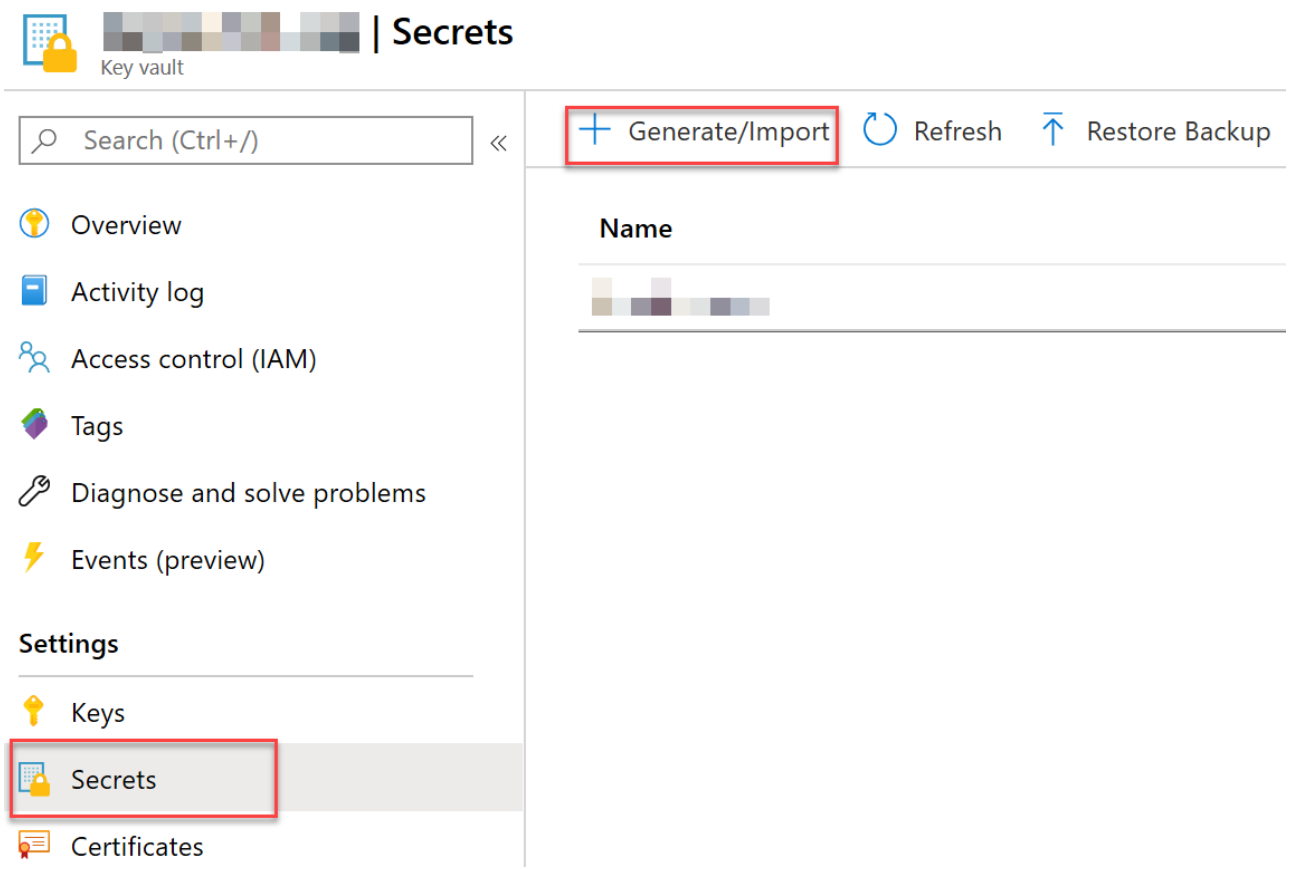
Since we have the Azure Key Vault set up as a linked service, we can leverage it when defining new linked services. Every New linked service provides the option to retrieve secrets from Azure Key Vault. The form requests the selection of the Azure Key Vault linked service, the secret name, and (optional) specific version of the secret.



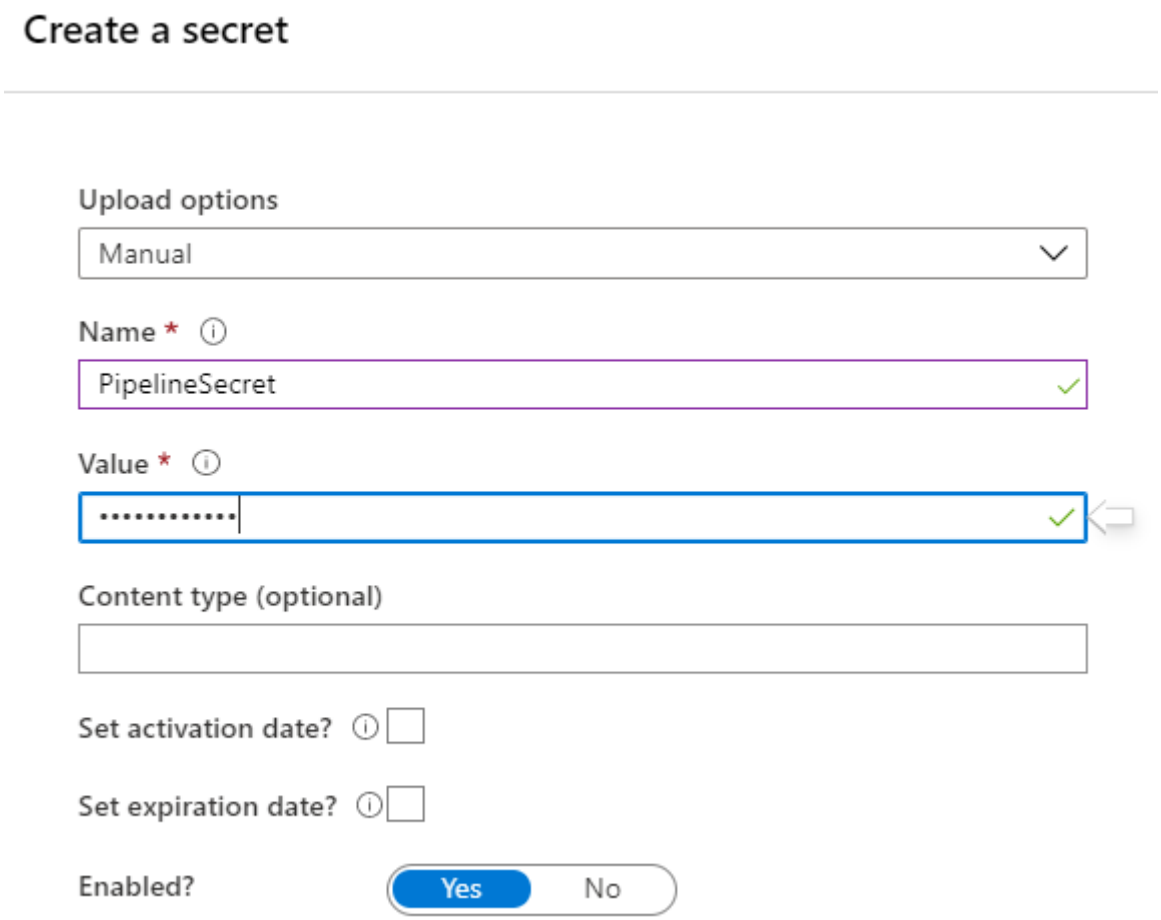
Task 3 - Secure workspace pipeline runs

It is recommended to store any secrets that are part of your pipeline in Azure Key Vault. In this task you will retrieve these values using a Web activity, just to show the mechanics. The second part of this task demonstrates using a Web activity in the pipeline to retrieve a secret from the Key Vault.

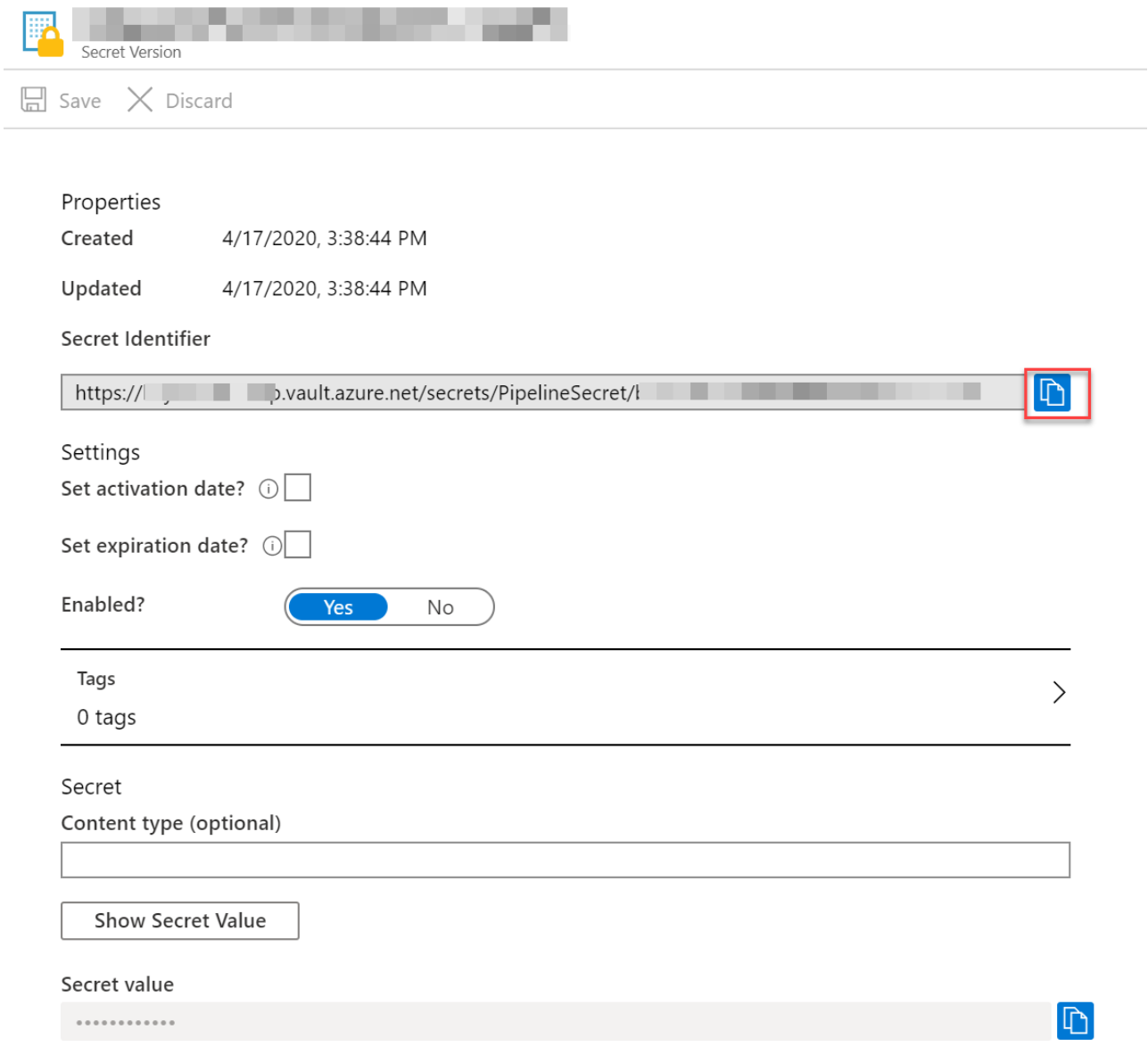
1. Open the `asakeyvaultXX` Azure Key Vault resource, and select **Secrets** from the left menu. From the top toolbar, select **+ Generate/Import**.



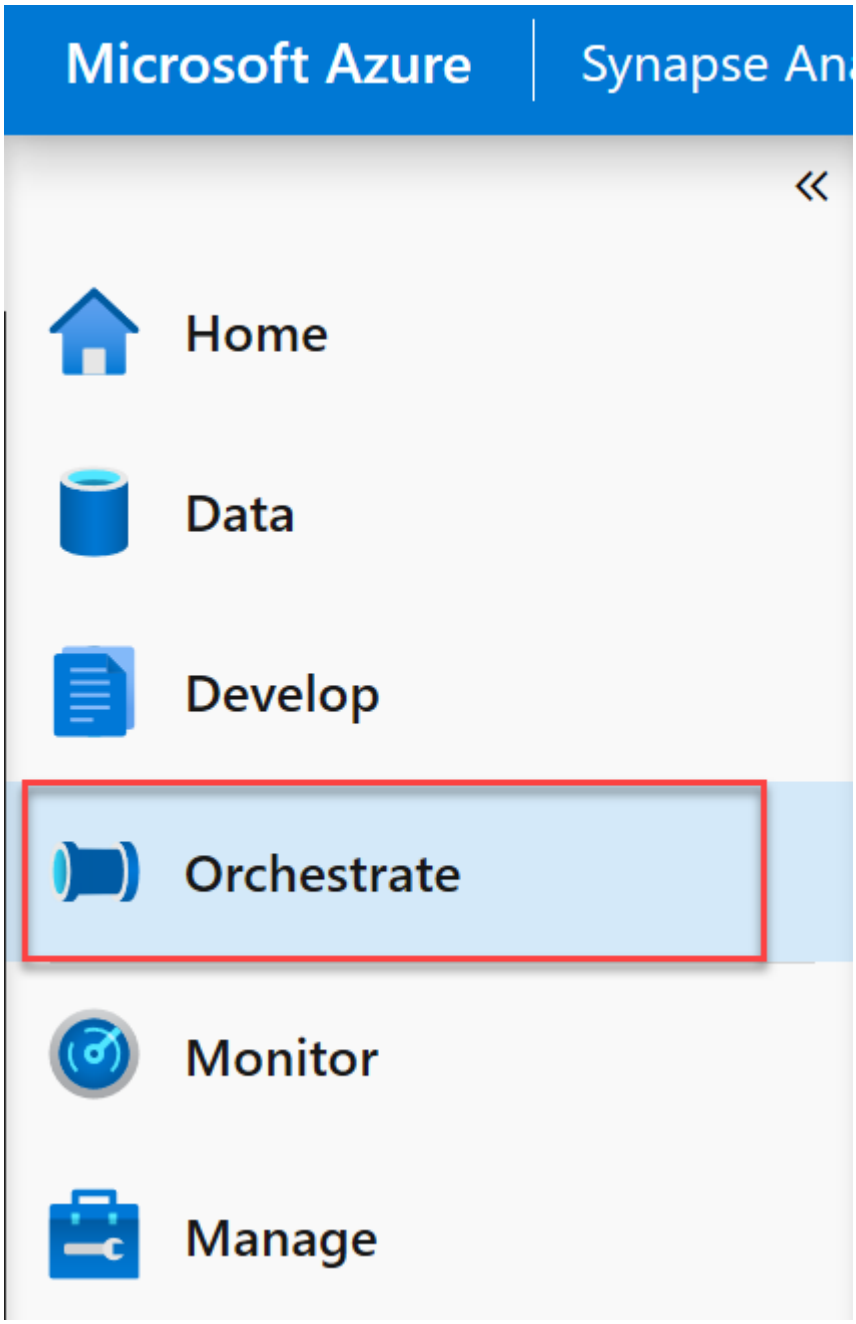
2. Create a secret, with the name **PipelineSecret** and assign it a value of **IsNotASecret**, and select the **Create** button.



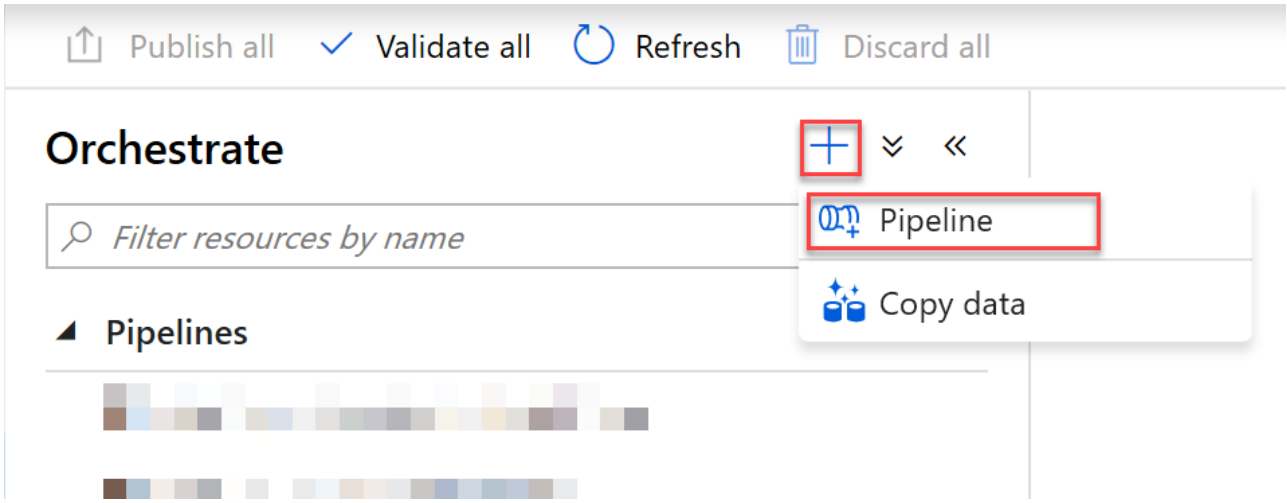
3. Open the secret that you just created, drill into the current version, and copy the value in the Secret Identifier field. Save this value in a text editor, or retain it in your clipboard for a future step.



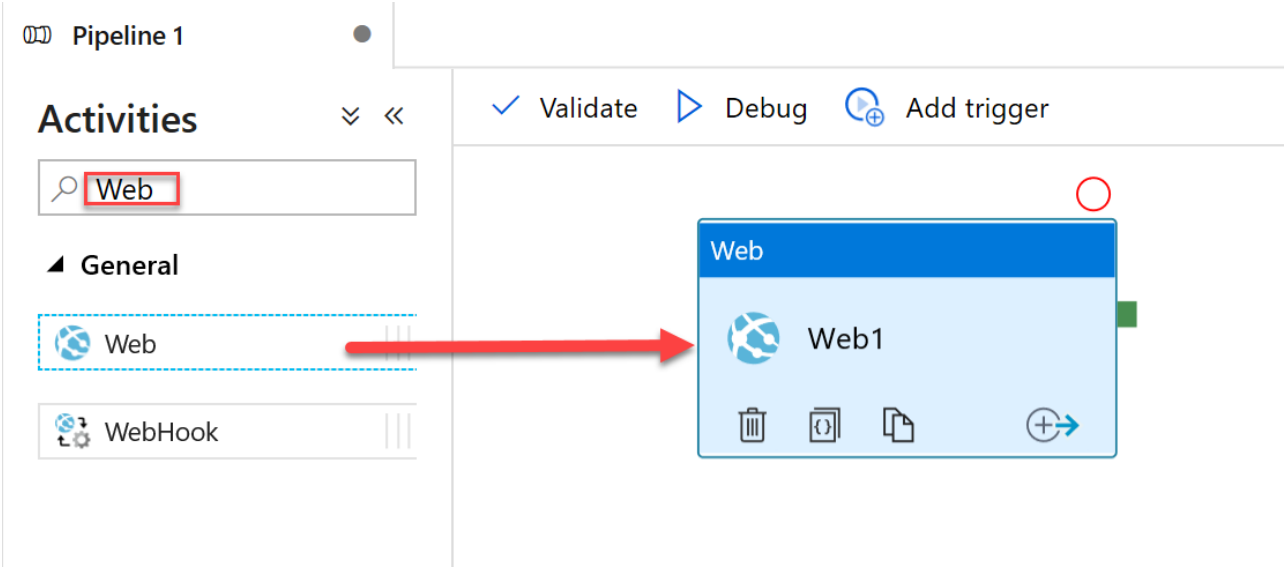
4. Open the Azure Synapse Analytics Studio, select **Orchestrate** from the left menu.



5. From the **Orchestrate** blade, select the **+** button and add a new **Pipeline**.



6. On the **Pipeline** tab, in the **Activities** pane search for **Web** and then drag an instance of a **Web** activity to the design area.



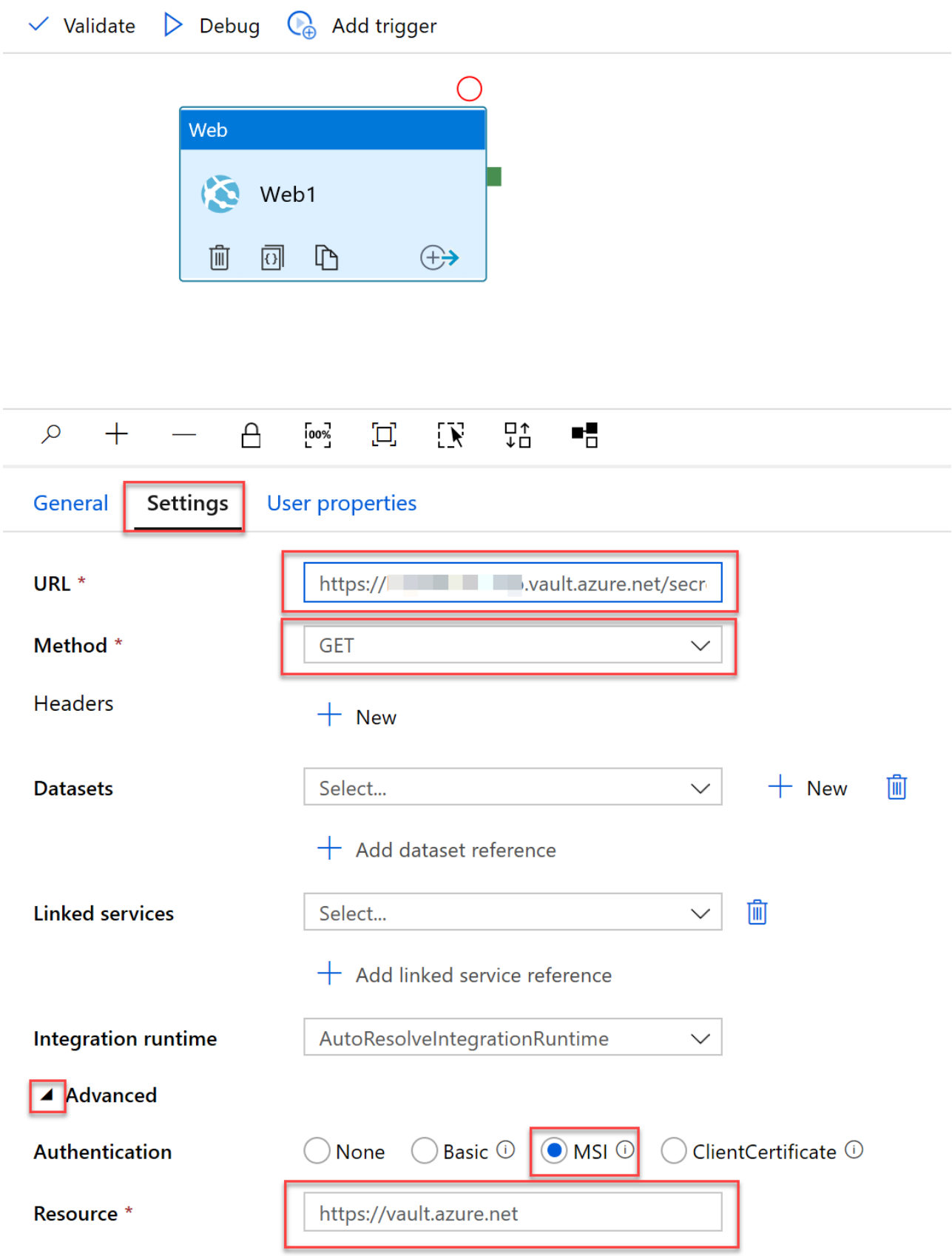
7. Select the **Web1** web activity, and select the **Settings** tab. Fill out the form as follows:

    1. **URL**: Paste the Secret Identifier value for the secret **append** `?api-version=7.0` to this value.

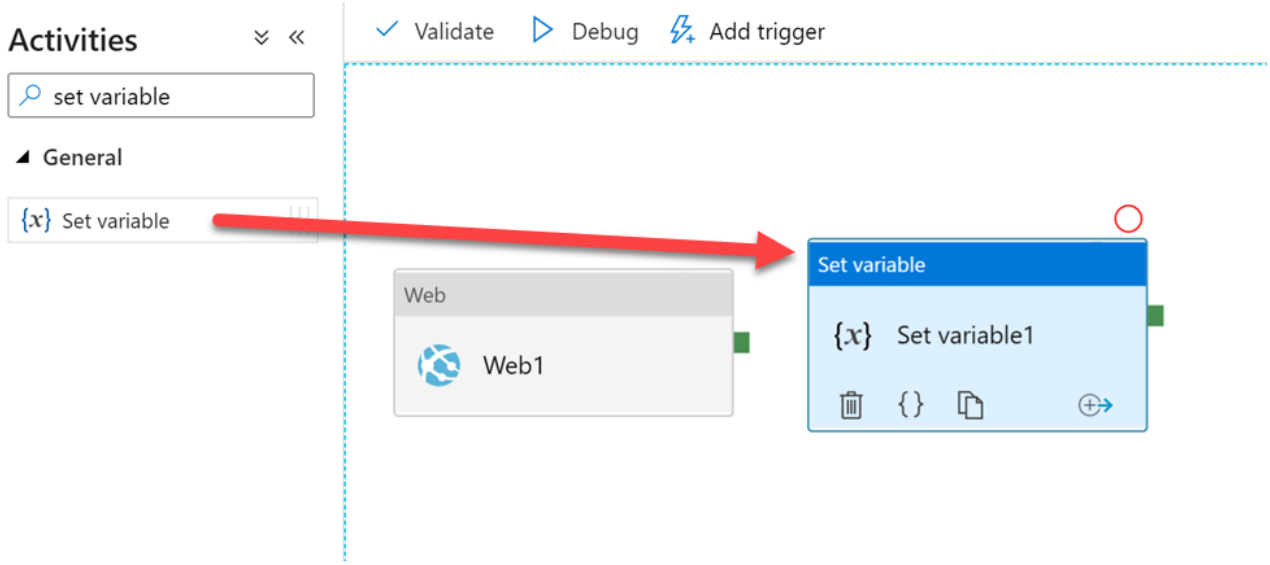
    2. **Method**: Select **Get**.

    3. Expand the **Advanced** section, and for **Authentication** select **MSI**. We have already established an Access Policy for the Managed Service Identity of our Synapse workspace, this means that the pipeline activity has permissions to access the key vault via an HTTP call.

    4. **Resource**: Enter **https://vault.azure.net**

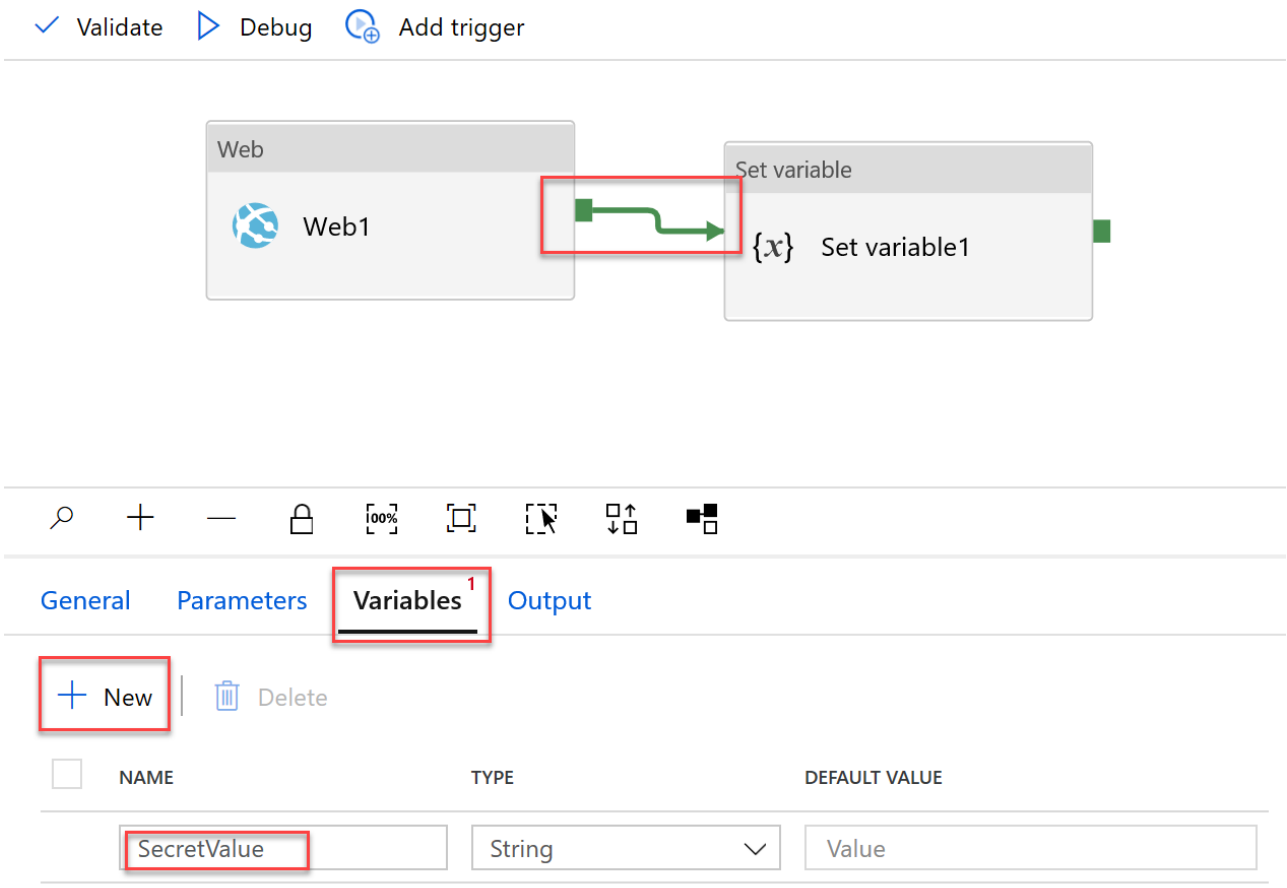


8. From the Activities pane, add a **Set variable** activity to the design surface of the pipeline.



9. On the design surface of the pipeline, select the **Web1** activity and drag a **Success** activity pipeline connection (green box) to the **Set variable1** activity.

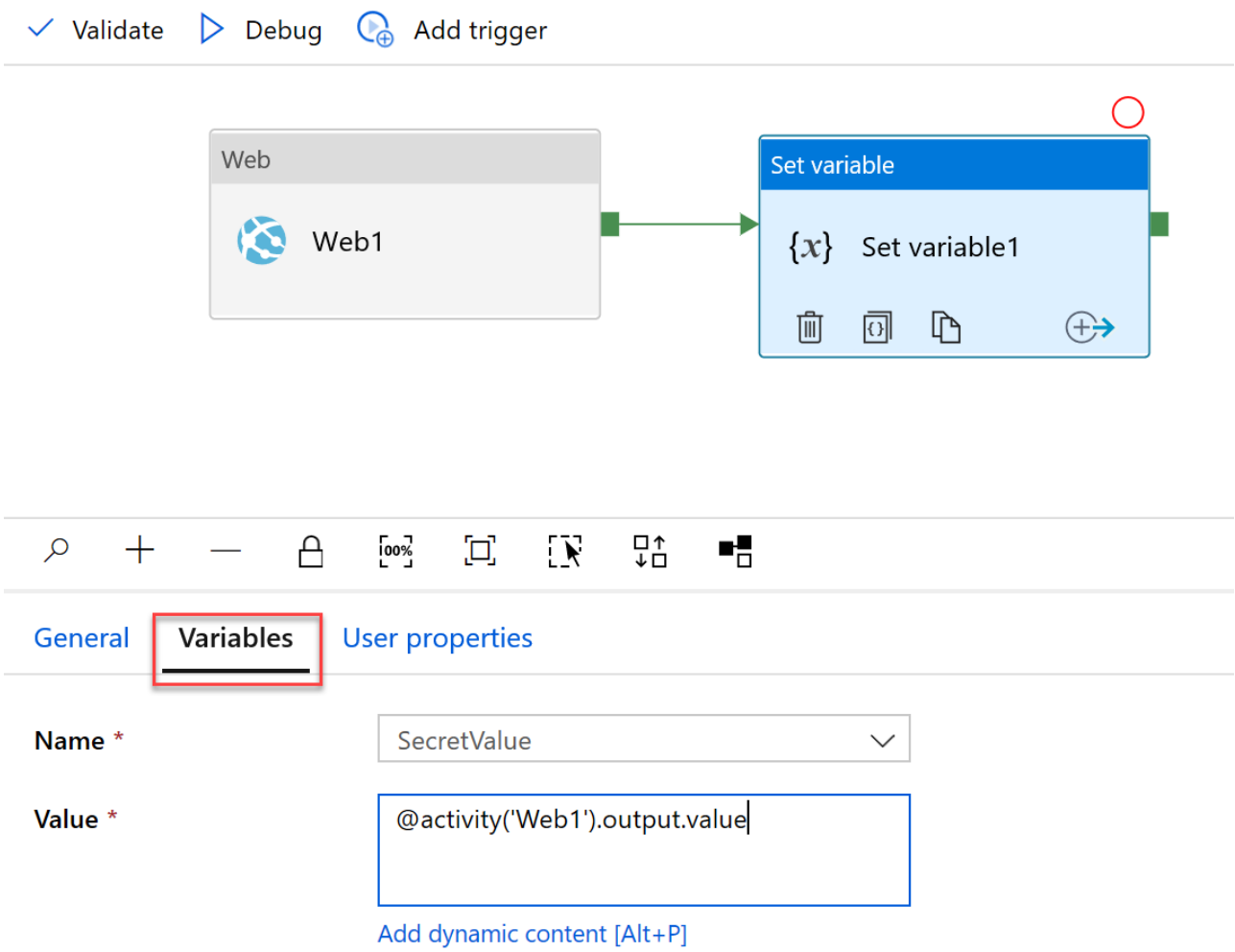
10. With the pipeline selected in the designer (e.g., neither of the activities are selected), select the **Variables** tab and add a new **String** parameter named **SecretValue**.



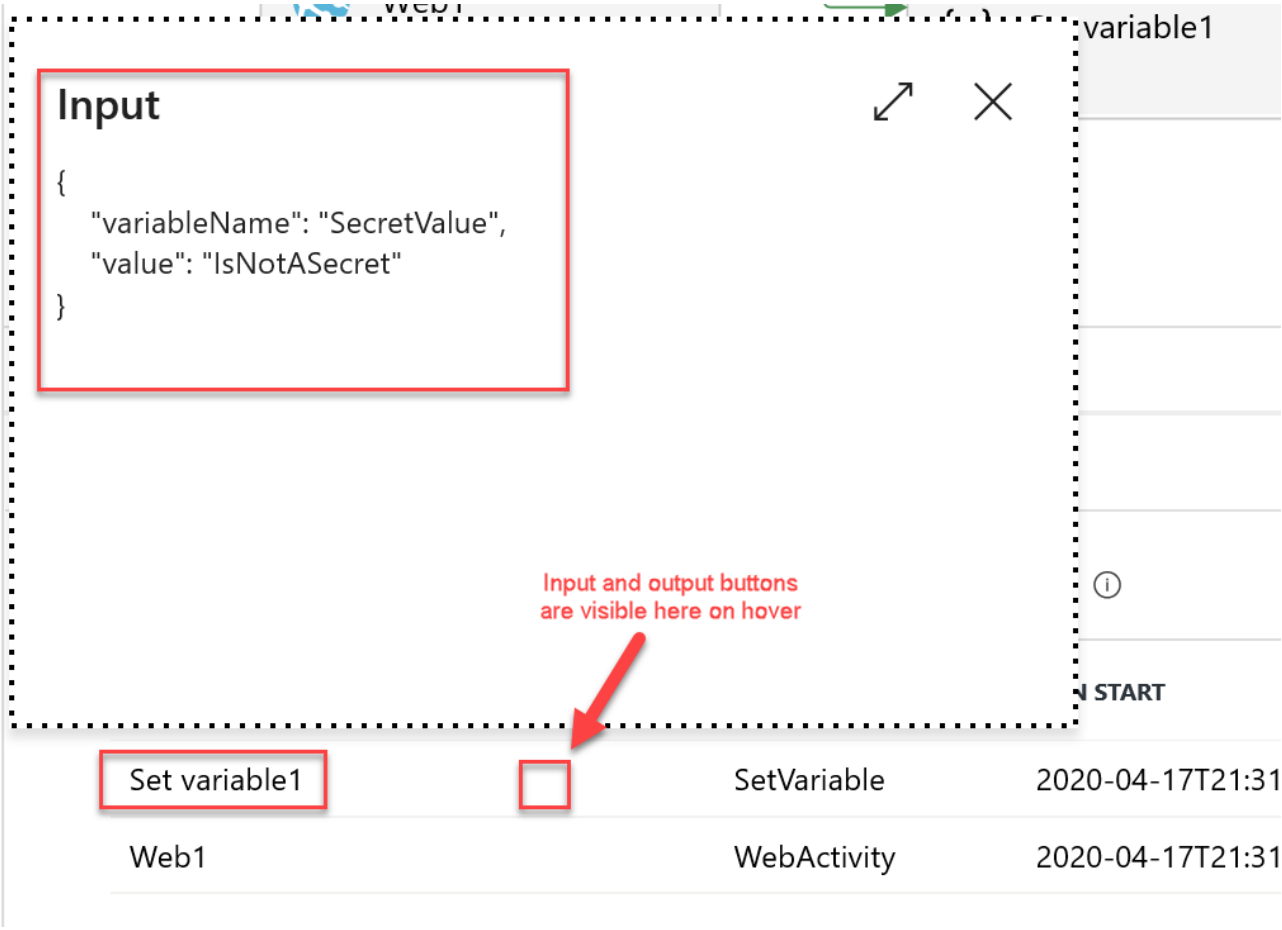
11. Select the **Set variable1** activity and select the **Variables** tab. Fill out the form as follows:

    1. **Name**: Select `SecretValue` (the variable that we just created on our pipeline).

    2. **Value**: Enter `@activity('Web1').output.value`



12. Debug the pipeline by selecting **Debug** from the toolbar menu. When it runs observe the inputs and outputs of both activities from the **Output** tab of the pipeline.

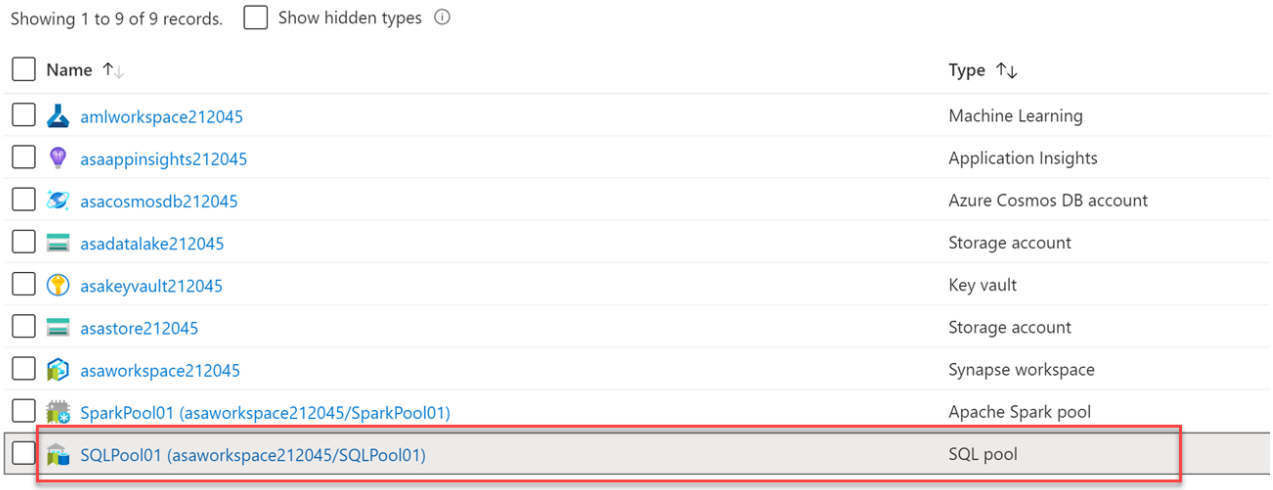


> **Note**: On the **Web1** activity, on the **General** tab there is a **Secure Output** checkbox that when checked will prevent the secret value from being logged in plain text, for instance in the pipeline run, you would see a masked value ***** instead of the actual value retrieved from the Key vault. Any activity that consumes this value should also have their **Secure Input** checkbox checked.

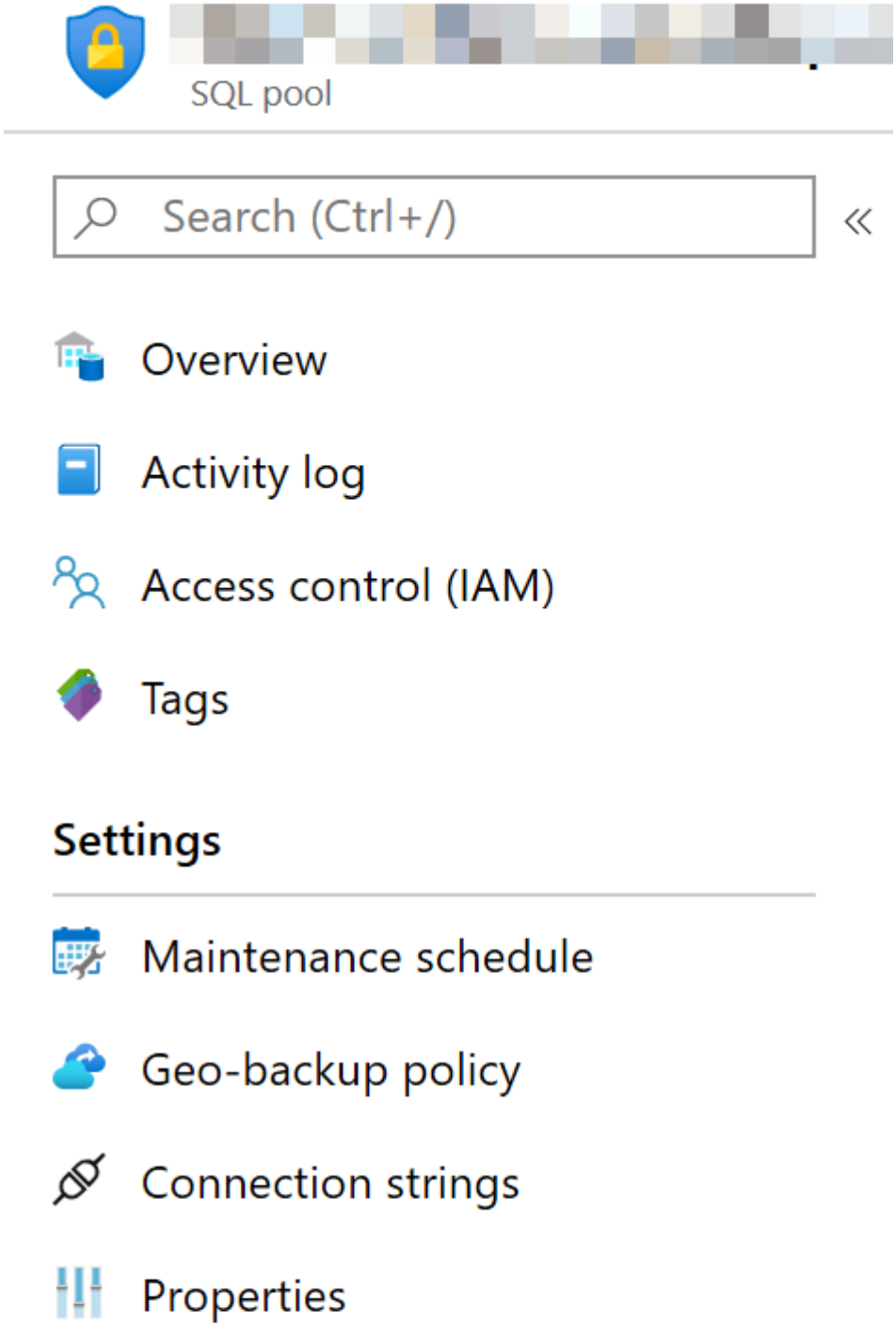
Task 4 - Secure Azure Synapse Analytics SQL Pools

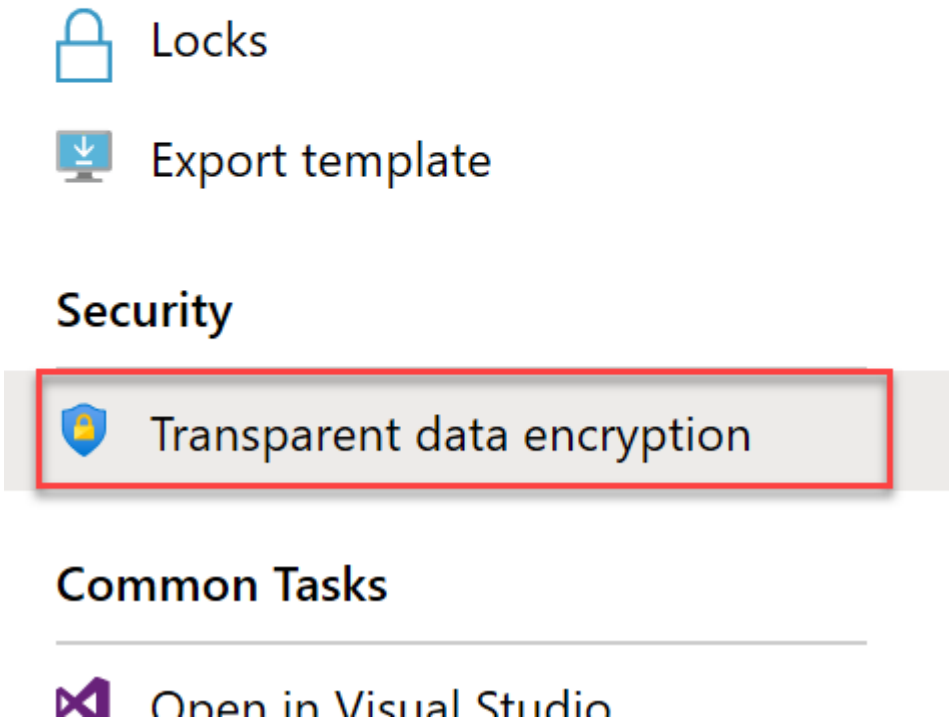
Transparent Data Encryption (TDE) is a feature of SQL Server that provides encryption and decryption of data at rest, this includes: databases, log files, and back ups. When using this feature with ASA SQL Pools, it will use a built in symmetric Database Encryption Key (DEK) that is provided by the pool itself. With TDE, all stored data is encrypted on disk, when the data is requested, TDE will decrypt this data at the page level as it's read into memory, and vice-versa encrypting in-memory data before it gets written back to disk. As with the name, this happens transparently without affecting any application code. When creating a SQL Pool through ASA, Transparent Data Encryption is not enabled. The first part of this task will show you how to enable this feature.

1. In the **Azure Portal**, open your resource group, then locate and open the `SqlPool01` resource.

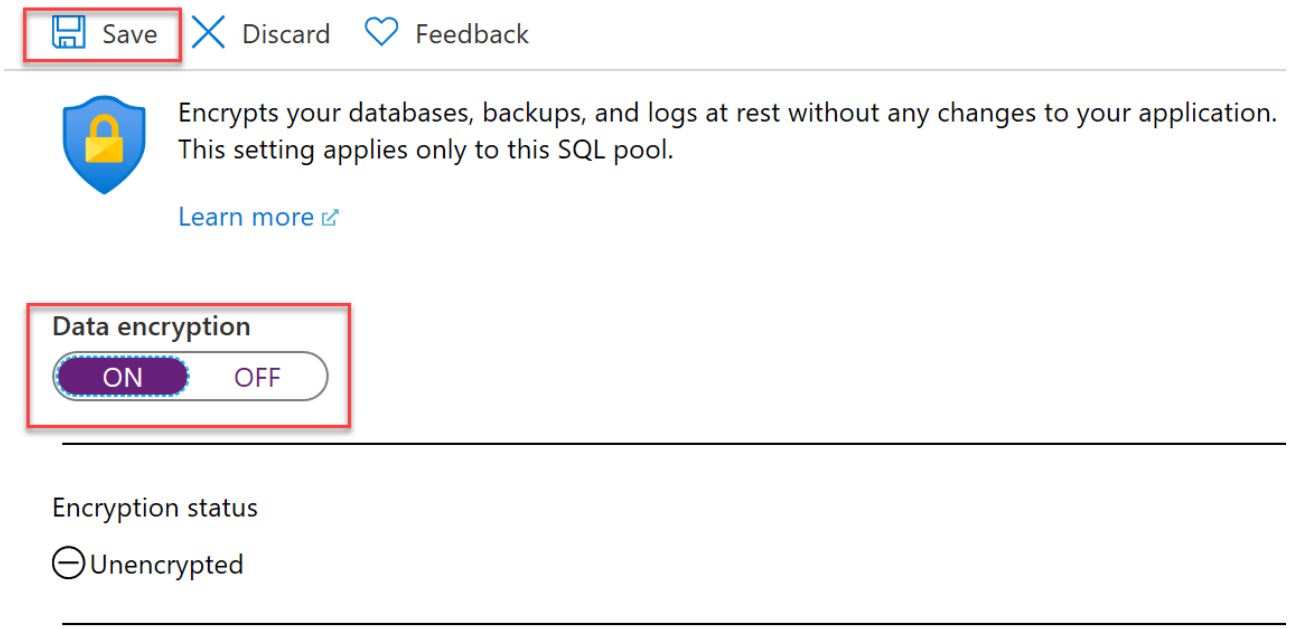


2. On the **SQL pool** resource screen, select **Transparent data encryption** from the left menu.





3. If your SQL Pool is not currently taking advantage of TDE, slide the **Data encryption** slider to the **ON** position, and select **Save**.



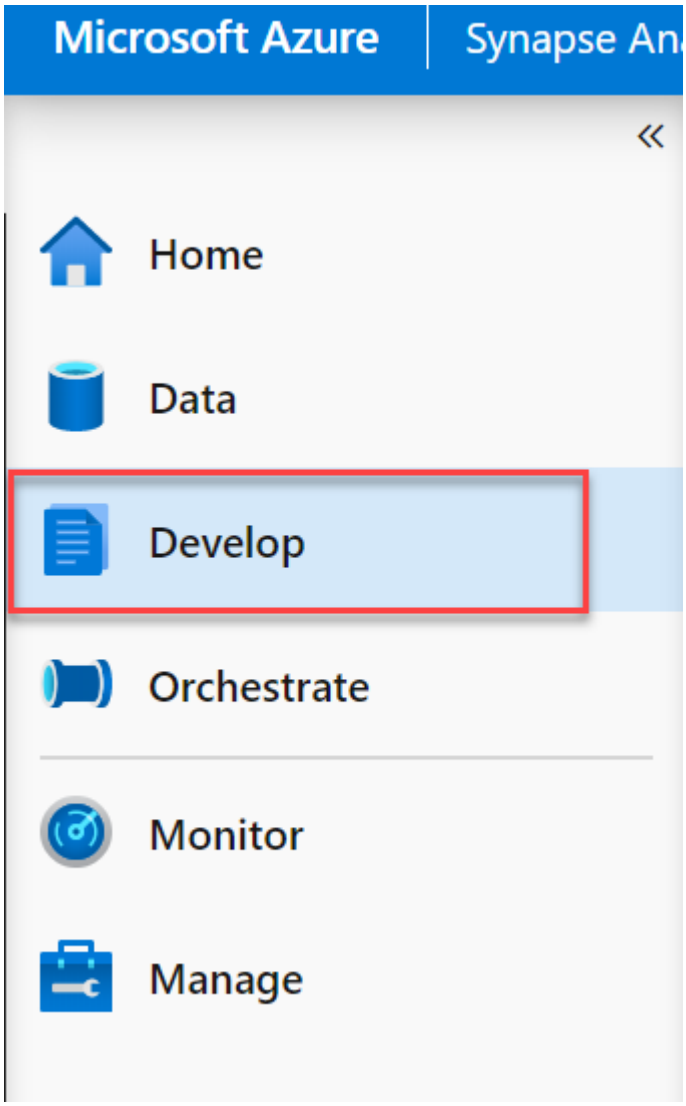
> **Note**: This operation can take several minutes to complete. You will not be able to the next exercise until this operation is completed.

Exercise 3 - Securing Azure Synapse Analytics workspace data

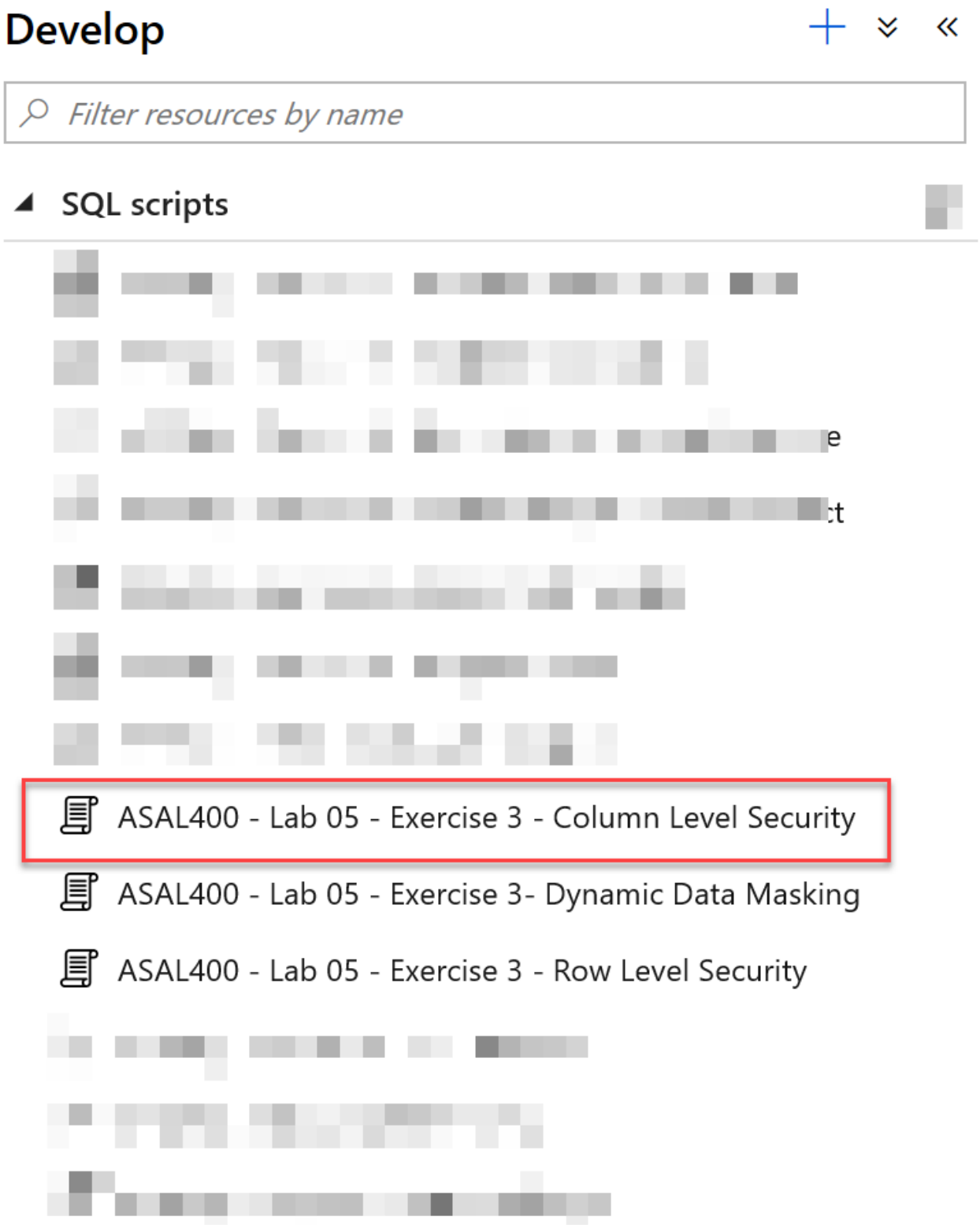
Task 1 - Column Level Security

It is important to identify data columns of that hold sensitive information. Types of sensitive could be social security numbers, email addresses, credit card numbers, financial totals, and more. Azure Synapse Analytics allows you define permissions that prevent users or roles select privileges on specific columns.

1. In **Azure Synapse Studio**, select **Develop** from the left menu.

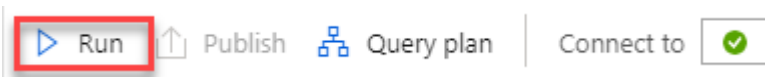


2. From the **Develop** menu, expand the **SQL scripts** section, and select **Lab 05 - Exercise 3 - Column Level Security**.



3. In the toolbar menu, connect to the database on which you want to execute the query, `SQLPool01`.

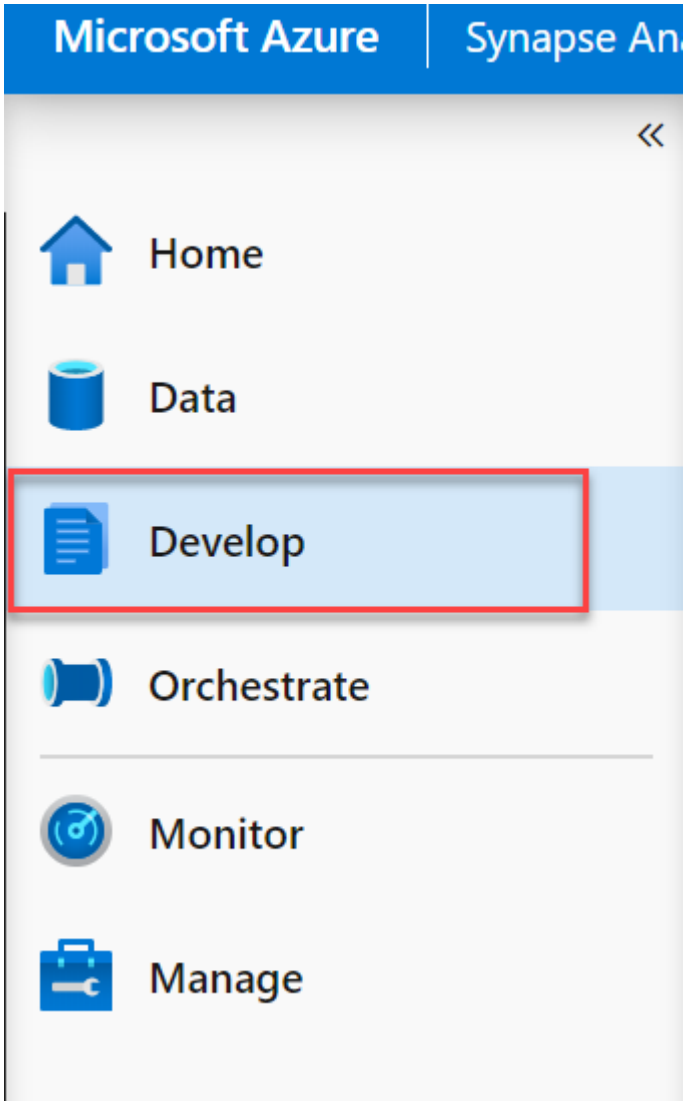
4. In the query window, **run each step individually** by highlighting the statement(s) in the step in the query window, and selecting the **Run** button from the toolbar.



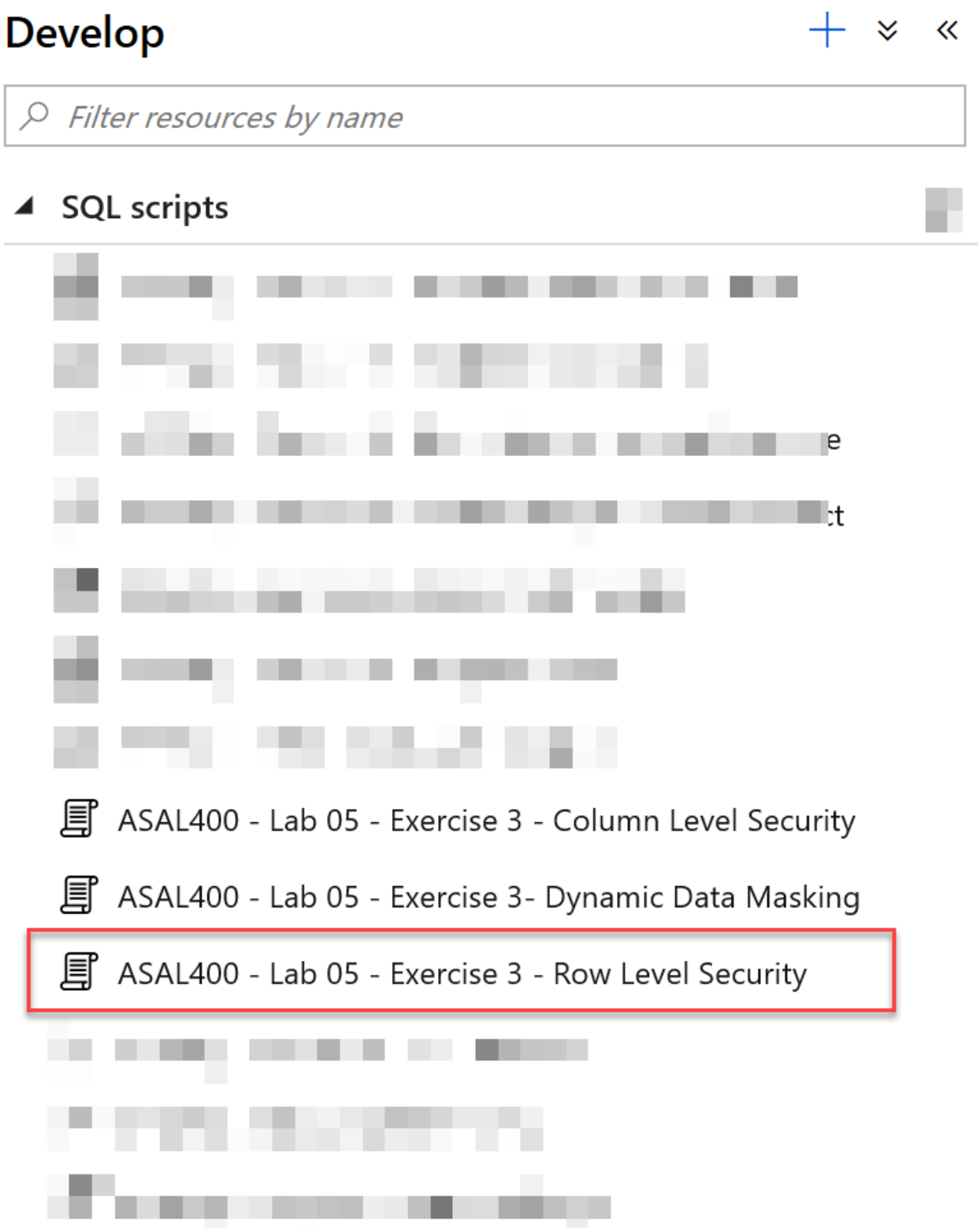
5. You may now close the script tab, when prompted choose to **Discard all changes**.

Task 2 - Row level security

1. In **Azure Synapse Studio**, select **Develop** from the left menu.

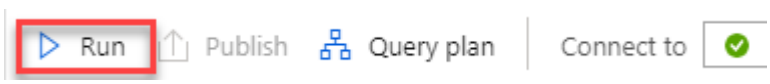


2. From the **Develop** menu, expand the **SQL scripts** section, and select **Lab05 - Exercise 3 - Row Level Security**.



3. In the toolbar menu, connect to the database on which you want to execute the query, `SQLPool01`.

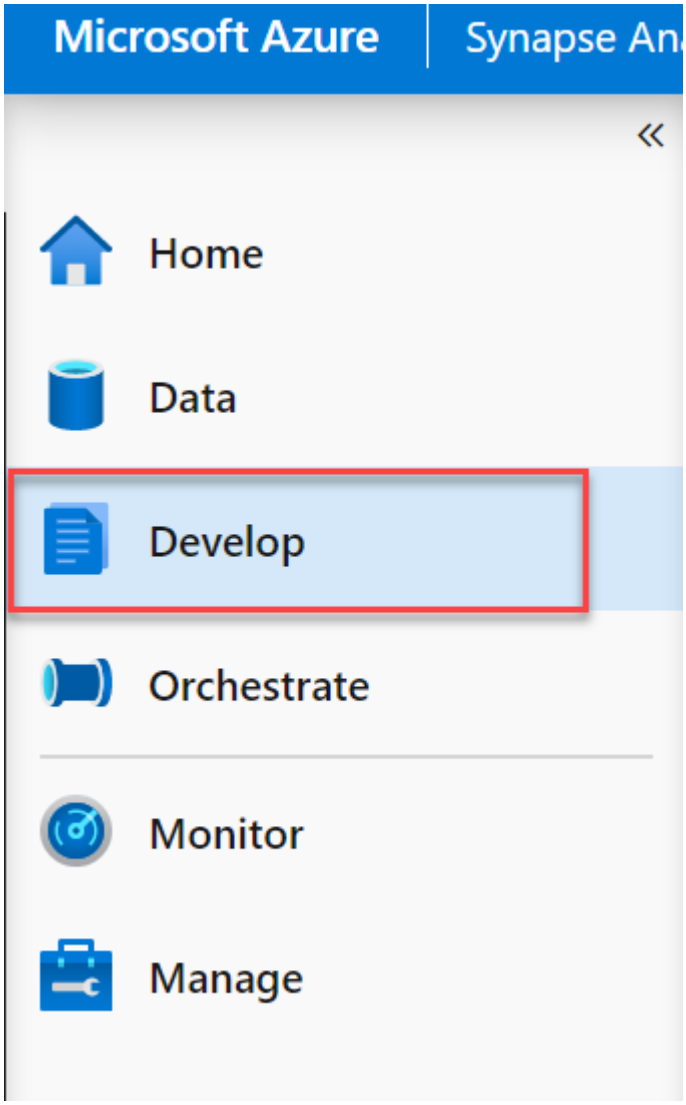
4. In the query window, **run each step individually** by highlighting the statement(s) for the step in the query window, and selecting the **Run** button from the toolbar.



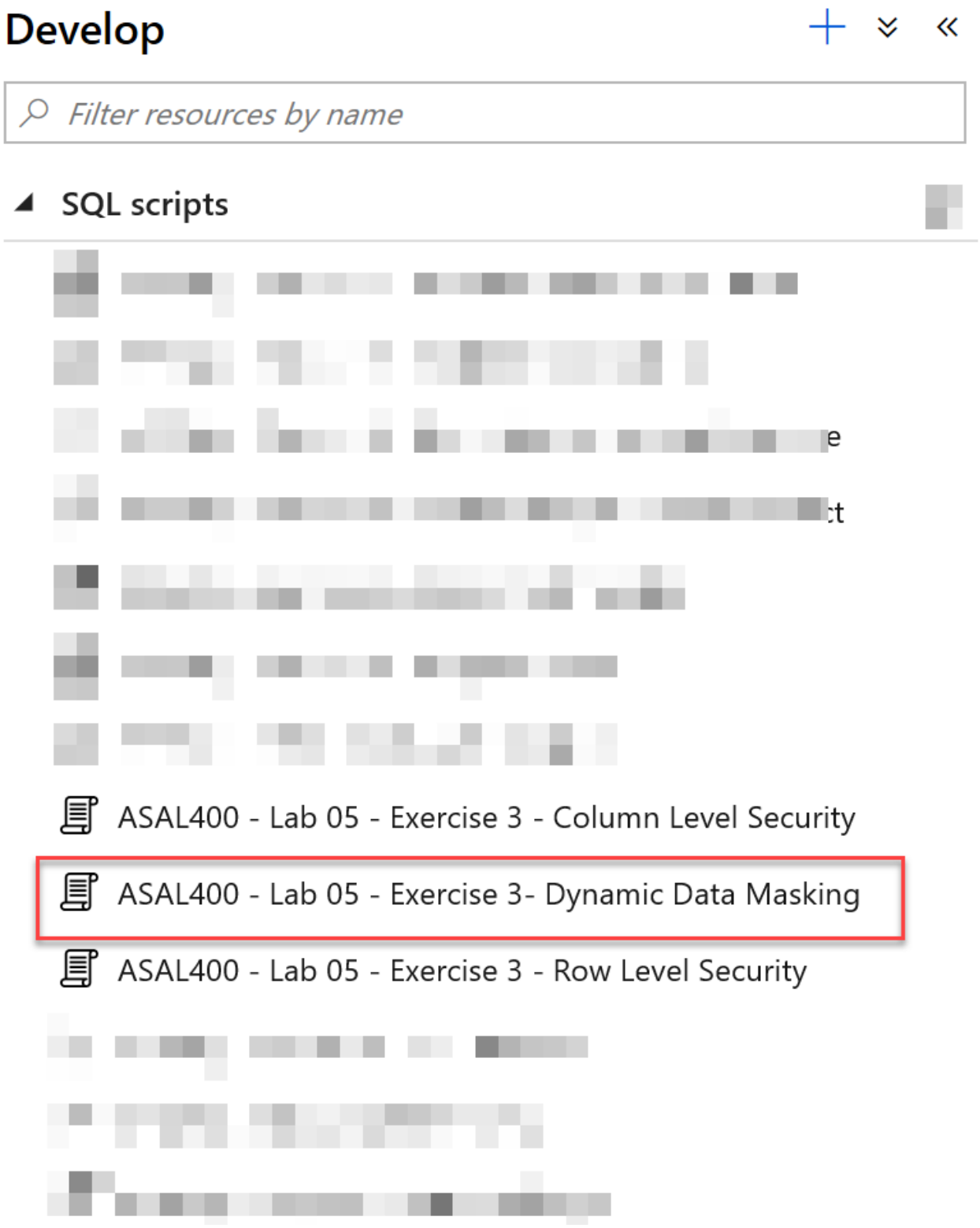
5. You may now close the script tab, when prompted choose to **Discard all changes**.

Task 3 - Dynamic data masking

1. In **Azure Synapse Studio**, select **Develop** from the left menu.

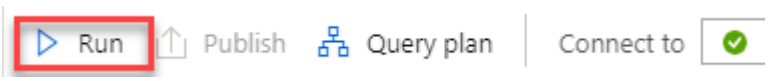


2. From the **Develop** menu, expand the **SQL scripts** section, and select **Lab05 - Exercise 3 - Dynamic Data Masking**.



3. In the toolbar menu, connect to the database on which you want to execute the query, `SQLPool01`.

4. In the query window, **run each step individually** by highlighting the statement(s) for the step in the query window, and selecting the **Run** button from the toolbar.



5. You may now close the script tab, when prompted choose to **Discard all changes**.

Reference

- IP Firewalls
- Synapse Workspace Managed Identity
- Synapse Managed VNet
- Synapse Managed Private Endpoints
- Setting up Access Control
- Connect to Synapse Workspace using Private Endpoints
- Create Managed Private Endpoints
- Granting Permissions to Workspace Managed Identity

Other Resources

- Managing access to workspaces, data and pipelines
- Easily read and write safely with Spark into/from SQL Pool
- Connect SQL Serverless with Power BI desktop
- Control storage account access for SQL Analytics Serverless