

8. Data Security

Azure Key Vault Service

We had used this earlier to store the storage account access keys. We created a databricks scope where we were storing a secret using this.

So, what is the purpose of Azure key vault service? Is a managed service that is used for storing and managing the lifecycle of your *certificates*, of your *encryption keys*, and of your *secrets*.

An example of where secrets can be used – Normally when an application is trying to connect to a database, the application needs to establish the connection using the database password. In this scenario, the database password can be stored as a secret in the key vault and then when the application wants to connect to the application it would make a call to the key vault service, fetch the password and then connect to the database.

Azure Data Factory – Encryption

ADF already encrypts data at rest which also include entity definitions and any data that is cached. Done with the help of Microsoft managed keys but can also define your own keys using the azure key vault services.

For the key vault you have to ensure that the Soft delete is enabled and the setting of Do Not Purge is also enabled. Also grant ADF the key permission of Get, Unwrap Key and Wrap Key.

To have encryption in your ***data factory using customer managed keys*** first you have to make sure there are no resources defined in data factory. So create a new data factory resource in the same region as your key vault.

- Open your data factory studio. From the left-hand section select Manage > Customer Managed Key
- Before we can add a key, we have to go to the key vault and give the required permissions to your new data factory.
 - Go to your key vault > Access policies from the left > Add Access Policy
 - Select Get, Unwrap and Wrap permission in Key Permissions
 - Select your data factory in your principal
 - Add the access policy
 - Save
- Now in the key vault, go to the keys menu from the left. Click generate
 - Name it and hit create
 - Go to the key, click on the current version and copy the key identifier url
- In your data factory, click add key in the customer managed key window and past the url and save
- Now the encryption is based on the customer managed key defined in the key vault service

Azure Synapse – Encryption

Using customer managed keys in Azure Synapse is possible when you create Azure Synapse itself. Begin by going to your key vault and creating a key.

- Create a new Synapse resource, give your resource name, storage account name
- In security there is an option Workspace Encryption at the bottom.
- Enable it and in select the select a key option.
- Select your vault and your newly created key
- Then you can go ahead with the creation of the workspace

Enabling transparent data encryption in the dedicated sql pool

Used to ensure that the data that is in the warehouse is encrypted. You can enable this by going to your dedicated sql pool and from the left hand menu go to the transparent data encryption and turn on data encryption and save.

Data Masking in Azure Synapse

If you want to hide data in a particular column from your users, we can use data masking. A rule would be created and that rule would decide how much data would be exposed to the user. There are different masking rules –

- Credit Card masking rule – used to mask columns that contain credit card details, only the last four digits are exposed
- Email – first letter of the email address is exposed
- Custom text – we can decide which characters to expose for a field
- Random number – we generate a random number for a field

For this purpose, we would copy the contents of an email address table from our adventureworks database onto your synapse sql pool.

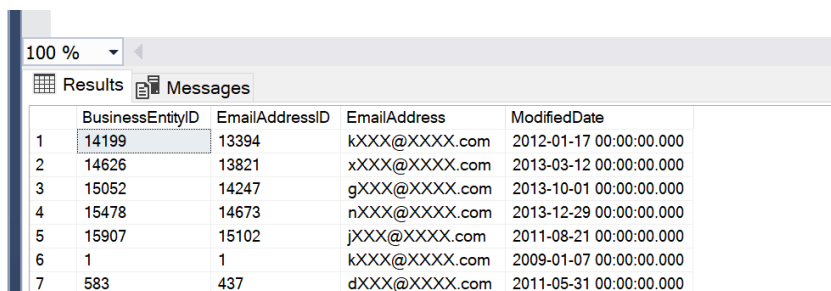
- Go to your Synapse studio, and select the copy data tool
- Select the adventureworks connection as the source and select person.EmailAddress as the table. Next
- In the target select the AzureSynapse connection and it would automatically create the target table
- Remove the rowguid from the mapping in the next page
- Disable staging and select bulk insert
- Once the pipeline has finished running
- Go to your dedicated pool in the SSMS and you can see the table
- **To enable email address masking**, go to your synapse workspace and from the left hand select sql pool. Select your dedicated sql pool.
 - On the left hand section of your dedicated sql pool in the workspace, select dynamic data masking
 - You would see it is already recommending certain masking operations to take place
 - Click on Add mask at the top
 - Select Person schema with the EmailAddress table, EmailAddress as the column and Email as the masking field format. Click Add
- Now go to your SSMS and execute the following command –

```
CREATE USER UserA WITHOUT LOGIN;

GRANT SELECT ON [Person].[EmailAddress] TO UserA;

EXECUTE AS USER = 'UserA';
SELECT * FROM [Person].[EmailAddress];
REVERT;
```

- You would be able to see that the email address field is being masked



The screenshot shows a SQL Server Enterprise Manager interface. At the top, there's a zoom level of 100% and tabs for 'Results' and 'Messages'. Below the tabs is a table with the following columns: BusinessEntityID, EmailAddressID, EmailAddress, and ModifiedDate. The table contains 7 rows of data. The EmailAddress column shows masked values, where only the first letter of the domain is visible (e.g., kXXX@XXXX.com, xXXX@XXXX.com, gXXX@XXXX.com, nXXX@XXXX.com, jXXX@XXXX.com, kXXX@XXXX.com, dXXX@XXXX.com).

	BusinessEntityID	EmailAddressID	EmailAddress	ModifiedDate
1	14199	13394	kXXX@XXXX.com	2012-01-17 00:00:00.000
2	14626	13821	xXXX@XXXX.com	2013-03-12 00:00:00.000
3	15052	14247	gXXX@XXXX.com	2013-10-01 00:00:00.000
4	15478	14673	nXXX@XXXX.com	2013-12-29 00:00:00.000
5	15907	15102	jXXX@XXXX.com	2011-08-21 00:00:00.000
6	1	1	kXXX@XXXX.com	2009-01-07 00:00:00.000
7	583	437	dXXX@XXXX.com	2011-05-31 00:00:00.000

Auditing in Azure Synapse

This is used to track database events and write them to an audit log. These logs in turn can be stored in an Azure Storage account, a log analytics workspace and azure event hubs. This helps us gain insights on any anomalies when it comes to database activities.

For this part we begin by creating a log analytics workspace resource from the create a resource section. Simple select the resource group, region and create.

- In Synapse workspace there is Azure sql auditing on the left hand side. The same auditing feature is available for the dedicated sql pool page as well. So we can enable auditing at the synapse workspace level or at the dedicated sql pool level.

- So for synapse workspace, enable sql auditing, select log analytics as the destination, your subscription, and the log analytics that you just created and save.
- Now data would start to get sent onto the the analytics workspace. It would take 10-15 minutes for the data to show up.
- In the log analytics workspace, go to the log section from the left hand menu
- You would see a SQLSecurityAuditEvents table that has appeared.
- So you can query the logs here and also create different kind of alerts based on the data that is coming in the logs

Data Discovery and Classification in Synapse

Provides capabilities to discover, classify, label and report the sensitive data in your databases. The data discovery feature can also scan the database and identify columns that contain sensitive data. You can then view and apply the recommendations accordingly.

We can also apply sensitivity labels to the column. This helps to define the sensitivity level of the data stored in the column.

Go to the page of your dedicated sql pool from your azure synapse workspace.

- Upon opening, it would give the message that it has found five columns (or a different number based on your tables) with classification recommendations
- Upon clicking and scrolling down, it shows the tables and the columns that might contain potentially sensitive info
- If you want you can also add a classification manually
- So we choose the Person Schema, EmailAddress table, BuisnessEntityID and select an information type and give a sensitivity label and hit add classification.
- On the previous screen select all the recommendations and hit Accept Selected Recommendations and save.
- Now in the overview you can see the classification of your sensitive data and your regular data

Azure Active Directory Authentication

It is the identity store in Azure. You can define users, groups, applications, etc. So you can define the users in your organization and based on that you can give them role based access control.

We have been using Synapse using the sql credentials but we can also log in to synapse using the active directory authentication. So you don't need separate users and then give them access to the sql pool. Instead if you have users in your Active directory, you could give them sql pool access through AD.

Setting the admin in Active Directory

From your homepage, go to Azure Active Directory. Upon going to Users from the left-hand menu, you can define all the users here. By default you would see your own account.

In your synapse workspace, on the left, there's SQL Active Directory admin. You can see yourself as the admin for the workspace and can also set another user as the admin for the workspace.

To add a new user to your Synapse dedicated sql pool, you can –

- Create a new user in Azure active directory if you haven't already
- Now log out of your SSMS or create a new connection and while logging in, Give Azure Active Directory – Universal with MFA as the authentication method. Give your Azure email and log in with the help of the pop up window to enter your password.
- Open up the SSMS and then use the following query –

```
-- Lab - Azure Synapse - Azure AD Authentication - Creating a user
```

```
CREATE USER [newsq1@techsup1000gmail.onmicrosoft.com]
FROM EXTERNAL PROVIDER
WITH DEFAULT_SCHEMA = dbo;
```

```
CREATE ROLE [readrole]
GRANT SELECT ON SCHEMA::[dbo] TO [readrole]
EXEC sp_addrolemember N'readrole', N'newsq1@techsup1000gmail.onmicrosoft.com'
```

- With the above query we are creating a new user in the pool from an external provider meaning the log in details would be coming from an external provider. We are then assigning a schema to the user as in what data they'd be able to access lastly the kind of role.
- Run the create user query, create the role with the schema and readrole, lastly give the permissions to the user
- Now log in as this user with the same type of authentication as above, an in the additional connection parameters mention the database name. For me it would look like - database=dp203
- Now we can execute any table that is part of the dbo schema design.

```
SELECT * FROM [dbo].[DimCustomer]
```

Row level security in Azure Synapse

The basic idea for this is only the rows meant for a particular user should be viewable by that user and all other rows show not be shown.

Begin by creating a table in your dedicated sql pool logged in as the administrator and use the following query –

```
CREATE TABLE [dbo].[Orders]
(
    OrderID int,    Agent varchar(50),    Course varchar(50), Quantity int
);
```

- Insert data using the query –

```
INSERT INTO [dbo].[Orders] VALUES(1, 'AgentA', 'AZ-900', 5);
INSERT INTO [dbo].[Orders] VALUES(1, 'AgentA', 'DP-203', 4);
INSERT INTO [dbo].[Orders] VALUES(1, 'AgentB', 'AZ-104', 5);
INSERT INTO [dbo].[Orders] VALUES(1, 'AgentB', 'AZ-303', 6);
INSERT INTO [dbo].[Orders] VALUES(1, 'AgentA', 'AZ-304', 7);
INSERT INTO [dbo].[Orders] VALUES(1, 'AgentB', 'DP-900', 8);
```

- Next we create three users with no logins. One is a supervisor which should see all the data, the other two are agents who would see their individual data. Next we grant the select permission to the three users.

```
CREATE USER Supervisor WITHOUT LOGIN;
CREATE USER AgentA WITHOUT LOGIN;
CREATE USER AgentB WITHOUT LOGIN;

GRANT SELECT ON [dbo].[Orders] TO Supervisor;
GRANT SELECT ON [dbo].[Orders] TO AgentA;
GRANT SELECT ON [dbo].[Orders] TO AgentB;
```

- Next we write a function at would return a value for each row based on who is executing the query. So if it is AgentA and the row has AgentA in it the value would be 1.

```
CREATE SCHEMA Security;
CREATE FUNCTION Security.securitypredicate(@Agent AS nvarchar(50))
    RETURNS TABLE
WITH SCHEMABINDING
AS
    RETURN SELECT 1 AS securitypredicate_result
WHERE @Agent = USER_NAME() OR USER_NAME() = 'Supervisor';
```

- Next we create a security policy where we pass the function as a means to filter out the data

```
CREATE SECURITY POLICY Filter
ADD FILTER PREDICATE Security.securitypredicate(Agent)
ON [dbo].[Orders]
WITH (STATE = ON);
GO
```

- Next we grant the users access to the function itself –

```
GRANT SELECT ON Security.securitypredicate TO Supervisor;
GRANT SELECT ON Security.securitypredicate TO AgentA;
GRANT SELECT ON Security.securitypredicate TO AgentB;
```

- Now execute the following commands one by one and see the difference in the results –

```
EXECUTE AS USER = 'AgentA';
SELECT * FROM [dbo].[Orders];
REVERT;

EXECUTE AS USER = 'AgentB';
SELECT * FROM [dbo].[Orders];
REVERT;

EXECUTE AS USER = 'Supervisor';
SELECT * FROM [dbo].[Orders];
REVERT;
```

- AgentA and AgentB are only seeing rows with AgentA and AgentB in it where as the supervisor is seeing all the rows.
- Clean up with the following query –

```
DROP USER Supervisor;
DROP USER AgentA;
DROP USER AgentB;

DROP SECURITY POLICY Filter;
DROP TABLE [dbo].[Orders];
DROP FUNCTION Security.securitypredicate;
DROP SCHEMA Security;
```

Column level security in Synapse

- We go through the same process as above of –
 - Creating a table
 - Creating supervisor and UserA
- Next we want to grant access to the tables for the users. This is the part which is different for the supervisor and the UserA compared to Row based security. Execute the following script –

```
GRANT SELECT ON [dbo].[Orders] TO Supervisor;
GRANT SELECT ON [dbo].[Orders](OrderID,Course,Quantity) TO UserA;
```

- Now execute the table as the supervisor and the user and observe that the columns that are returned are different

```
EXECUTE AS USER = 'UserA';
SELECT * FROM [dbo].[Orders];
SELECT OrderID,Course,Quantity FROM [dbo].[Orders];
REVERT;
```

- Also, the select * statement would not work for the UserA as it does not have access to all the columns of the table.

Role based access control in Data lake

When you add a new user, they can log in but would not be able to access any resources that are part of your subscription. You would need to give them control via role based access control. There are four basic roles that can be assigned – contributor, owner, reader and user access administrator.

More information about the roles is given on the site - [Azure built-in roles - Azure RBAC | Microsoft Docs](#)

If you give the *reader* role to a user, they would be able to view the resources but not be able to make any changes to the resource. If the *contributor* access is given, the user would be able to manage the resources but cannot give access to other users. *Owner* role gives full access privileges to that resource.

For storage roles, we have storage blob data contributor. This gives specific permissions to the user to add blobs in containers in that storage account. So the access is being given to the data and not just the resource.

To see an example of this,

- Go to azure active directory and create a new user
- Now in a private window, sign in as that user
- So here we would not be able to see any resources, we would only be able to log in because there were no roles assigned to the user
- Now in your admin account, go to your data lake and from the left select access control
- Select Add > Add role assignment > reader > newly created user above > review and assign
- You would see this user appear in the role assignments tab of the access control section
- Now as the new user go to all resources and you would see the data lake show up

Access Control List in Data Lake

Right now when we go to the data inside the containers of the data lake as the newly created user, we would not be able to see any data. This is because the user has only been granted access to the data lake and not the data that is inside the data lake. For that we would need to make use of ACL to grant access on to your files and directory.

In access control of your data lake, add a new role assignment > Storage Blob Data Reader > Select the above new user > review and assign

- Now open your azure storage explorer. With your admin user, right click on any container that you want to give access to to the new user and select Access Control List. Click Add and search for the new user > Select Access check box and Read check box.
- Now right click on the directory inside the container add user. Same process as above for access control list, add user and select access and read check boxes.
- Right click again on the folder and select propagate access control list.
- Now you should be able to see the data in your new user's view of the data lake.

Azure Synapse – External table authorization via managed identity

In Azure, for certain resources, you can enable managed identity. When enabled, an identity gets created in the Azure active directory.

- So, begin by going to your data lake access control screen. Add role Reader and search for your Synapse workspace name. Review and assign.
- Also give the storage blob data reader role to the same synapse workspace name
- From your Azure storage explorer, give the access control list permissions to the container and the directory where the log.csv file is
- Propagate the access control list
- Open your SSMS. First we create the database scoped credential

```
CREATE DATABASE SCOPED CREDENTIAL AzureManaged
WITH IDENTITY = 'Managed Identity'
```

- Next we create a data source pointing to our data lake gen2 account

```
CREATE EXTERNAL DATA SOURCE log_data_managed
WITH ( LOCATION = 'abfss://data@datalake2000.dfs.core.windows.net',
        CREDENTIAL = AzureManaged,
        TYPE = HADOOP )
```

- We create the external file format –

```
CREATE EXTERNAL FILE FORMAT TextFileFormatManaged WITH (
    FORMAT_TYPE = DELIMITEDTEXT,
    FORMAT_OPTIONS (
        FIELD_TERMINATOR = ',',
        FIRST_ROW = 2))
```

- Next we create our external table making use of the log.csv file in the raw directory

```
CREATE EXTERNAL TABLE logdatamanaged
(
    [Id] [int] NULL,
    [Correlationid] [varchar](200) NULL,
    [Operationname] [varchar](200) NULL,
    [Status] [varchar](100) NULL,
    [Eventcategory] [varchar](100) NULL,
    [Level] [varchar](100) NULL,
    [Time] [datetime] NULL,
    [Subscription] [varchar](200) NULL,
    [Eventinitiatedby] [varchar](1000) NULL,
    [Resourcetype] [varchar](1000) NULL,
    [Resourcegroup] [varchar](1000) NULL
)
WITH (
    LOCATION = 'raw/Log.csv',
    DATA_SOURCE = log_data_managed,
    FILE_FORMAT = TextFileFormatManaged
)
```

- Upon doing a select * from logdatamanaged we can see the data is populated.

With this example we can see how we didn't have to provide credentials in the script. The access control to the data lake was given to Synapse by the admin and data is accessible by the synapse pools easily.

Firewalls in Azure Synapse (renamed to Networking)

This feature can be accessed from the left-hand menu of your synapse workspace from the Networking option.

In the Networking screen you would see your ip address and one rule that says allowAll and a range of ip addresses that can access the synapse workspace. By default, the start ip address is 0.0.0.0 and end ip address is 255.255.255.255 meaning anyone with an internet connection can access this synapse workspace.

If you delete this rule and try to connect to it from your SSMS, it would give an error that your client ip address doesn't have access to the server.

If you want to be able to access your synapse from your ip, click the Add client ip button and a new rule would be added with just the client ip address that is allowed to access.

But let's say the only rule that is there is of the client ip address, then no other service would be able to connect to the Synapse dedicated sql pool as well. For example if we create a connection to it from our data factory to our dedicated sql pool, then we would get an error.

To work around this problem is to check the box where it says allow azure services and resources to access this workspace.

Skipping a part about creating a secure connection between a virtual machine and a data lake. But here's more documentation
 - [Azure virtual network service endpoints | Microsoft Docs](#)