# Program Connections and Gateways

Creating connections programmatically is required in scenarios which involve OneLake Shortcuts, Data Pipelines and Semantic Models.

Before we dive into a discussion of connections, let’s take a moment to distinguish between **Inbound Security** versus **Outbound Security**. Inbound security is involved when a custom application executes API calls on Fabric REST API endpoints. A key point is that the custom application runs outside the Fabric environment. Before the application can call to the Fabric REST API, it must first authenticate with the Entra Id Service in order to acquire access tokens. It must then transmit an access token in each and every API call to Fabric REST API endpoints. This is a topic that has already been covered earlier in this guidance document.

Outbound security is different because it involves a scenario where you’ve created some type of workspace item inside the Fabric environment which must connect to an external datasource. For example, you can create a semantic model which connects to an Azure SQL database. In another example, you can create a OneLake shortcut which connects to an ADLS storage container. With outbound security, you can create and bind connections using the Fabric REST API as shown in the following diagram.

A close-up of a screen

Description automatically generated

Here is an important factor to keep in mind. When you create a connection with code, you are not really establishing a connection across the network at that point in time. Instead, you are really just creating a persistent Fabric object with metadata for creating a connection at some point in the future. This metadata includes the datasource type and path as well as security credentials. It is not until the connection is actually used by something in Fabric such as a semantic model or a OneLake shortcut when the Fabric Service reads this metadata and uses it to establish a connection to the datasource across the network.

Connections in Fabric are scoped at level of Entra Id tenant. That means that Fabric connection can be shared across workspaces. Of course, just because you can doesn’t mean that you should.

You can see all the connections you have access to using the Manage Connections and Gateways page in the Fabric Service.

A screenshot of a computer

Description automatically generated

Fabric connections support four different connectivity types.

1. Personal Cloud Connections (PCCs)
2. Sharable Cloud Connections (SCCs)
3. On-prem Gateway Connections
4. Virtual Gateway Connections

Personal Cloud Connections (PCCs) have been used in Power BI for years, However, they are limited because they cannot be shared. Each PCC is owned and exlussively used by a single user or service principal. There is no way to share a PCC.

Sharable Cloud Connections (SCCs) are new and serve as the strategic replacement for PCCs. Once a SCC has been created, it can be shared with other users or service principals. The creator of a Connection is automatically configured with connection Role Assignment of Owner. Other users and service principals can be added to the SCC membership with a Role Assignments of User, UserWithReshare or Owner.

## Create an Anonymous Web Connection

Content to come

## Create a Azure SQL Connectiong using Basic Credentials

Content to come

## Create an Azure Storage Connection using Service Principal Credentials

Content to come