Data Engineering on Azure - The Setup - 1/4

qimia.io/en/blog/data-engineering-microsoft-azure-set-up

The Setup Introduction

In recent years, Machine Learning has received a lot of attention due to the progress made in research. Many companies would like to leverage the opportunities offered by these techniques or perform Business Analytics but struggle to do so, because their data is locked up in silos like in OLTP databases containing operational data. Historically, there have been two main approaches to utilize data for analytical and Machine Learning purposes:

1. With a Data Warehouse
2. With a Data Lake

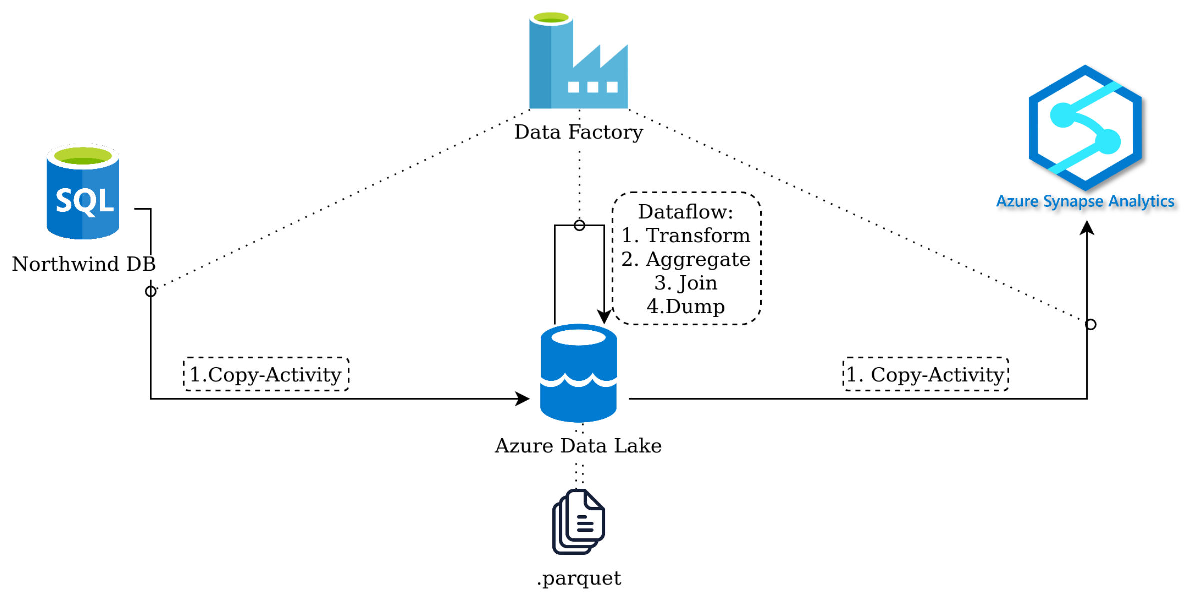
Access to the data is granted by utilizing ETL (Data Warehouse) or ELT (Data Lake) processes. While the concept of a Data Lake is mostly about avoiding to transfer data and keeping it where it is, Data Warehouses focus on transforming and storing data in a way that enables lightning-fast analytical queries. Setting up the appropriate Data Architecture is absolutely crucial to the success of not only classical analytical solutions but also Machine Learning Algorithms. With this short introduction, we would like to welcome you to our tutorial series about Data Engineering on MS Azure!

Target

In this tutorial, we are going to teach you how to:

- Set up the appropriate resources on Azure and manage their access to each other
- Build pipelines in Data Factory that can read, transform, partition, and write data
- Set up appropriate star schemas in the Azure Data Warehouse Solution 'Synapse'

The series will eventually lead us to the following setup:



Data Engineering on Azure Schema
The *tutorial* will consist of *four parts*, building on each other:

1. Setup and Introduction to the project
2. Load the Northwind Database, batch and dump the data to a Data Lake
3. Design the appropriate star schema for our data based on assumed access patterns
4. Transform, aggregate, join and dump the data to the Data Lake & copy the data to Synapse on a daily basis

This is the first part of our series and the focus of it will be the setup of our starting resources including their most important configuration options. We will prepare for our first pipeline that will transfer data from our SQL database to the Azure Data Lake Storage. We are going to use the well-known Northwind Database.

If you are not interested in setting up the components by yourself, check out our Azure CLI script on Github. Be aware that you will still have to put the Storage Account Access Key into the Key Vault as a Secret.

Setting up the Resources

Before we get started with our first pipeline, we will set up these few things:

1. Resource Group
2. Azure Data Lake Storage Gen 2 (ADLS G2)
3. SQL Database and Database

4. Key Vault

1. Resource Group

In Azure, resources are organized into Resource Groups. We will set up a new resource group, where we will store all our resources. To do so, go to your Azure Portal and search for Resource Groups and add a new one.

We will call ours 'qde_rg' for Qimia Data Engineering resource group. We are going to use that prefix throughout the whole tutorial. Make sure to use a region that contains all the services that we need. We are going to use **US East**.

2. Data Lake Storage

Next, we will create the Data Lake Storage Gen 2. In Azure, this storage utilizes Blob Storage which runs on storage accounts, so we need to create a new storage account first. From your resource group, add a new resource and search for the storage account. When creating a blob storage, you will be confronted with configuration options.

2.1. Read/Write Performance

To reduce the costs, we will go for the **Standard Performance tier**, **General purpose V2 storage**, **LRS,** and **Cool access tier**. Be aware that you can opt for the Premium Performance tier, blob storage, and hot access tier if you want to speed up reading and writing.

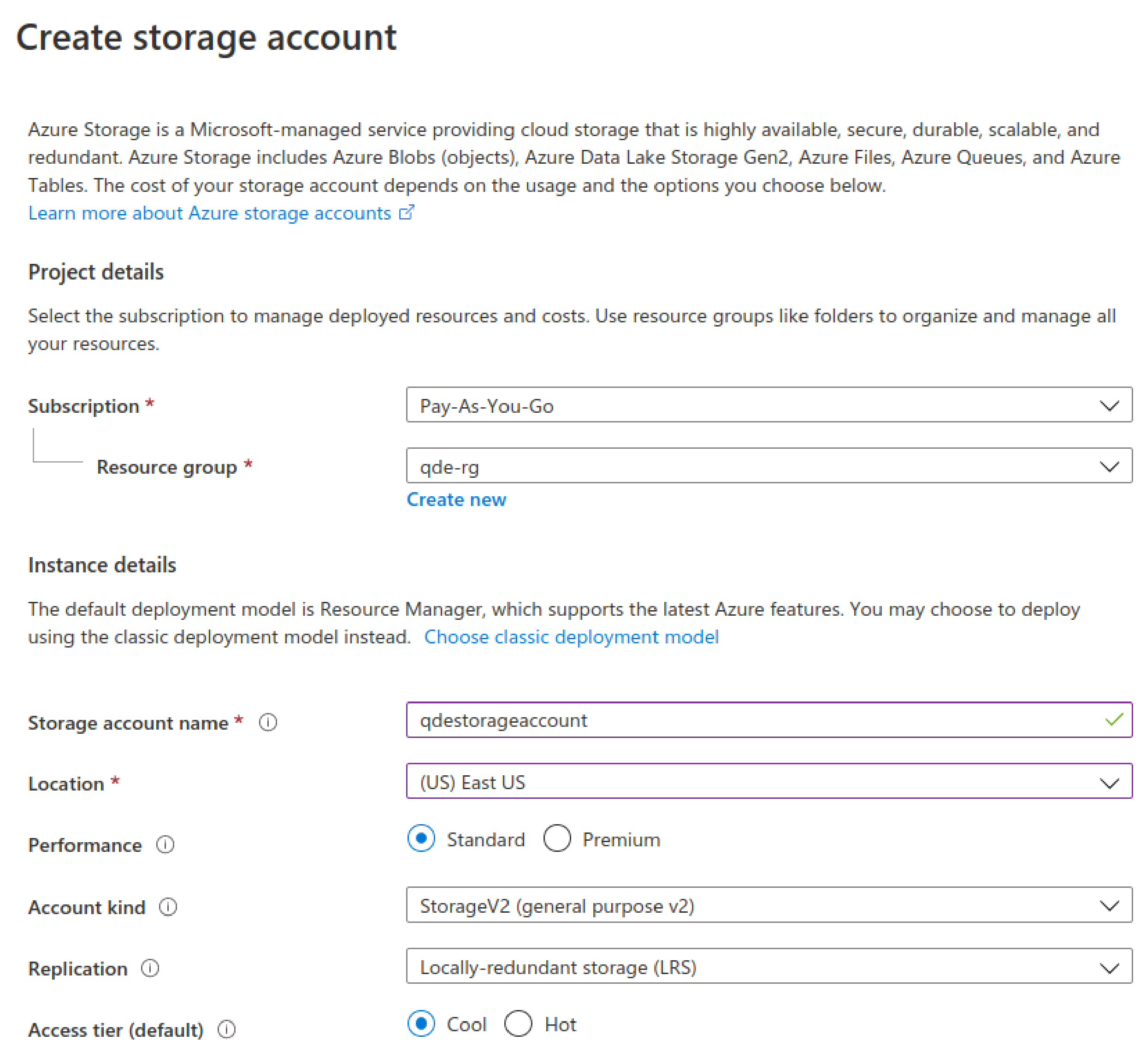
2.2. Durability & Security

Choosing another replication option can be useful to improve on the durability of your storage in cases of disasters. When opting for replication to other regions, be aware that reading from the asynchronous replicas is disabled by default. Enabling soft delete is another option to improve the durability in the case of unintentionally deleted files. We can go without it for our case.

2.3. Hierarchical Namespaces

We will require secure transfer and disable public access to blobs for security reasons. Make sure to check the option for hierarchical namespaces, which will turn this storage account into the Data Lake storage.

Apart from some additional options for Access Management, the hierarchical namespace is the main advantage of the Data Lake Storage over casual Blob Storage.



Create storage account

Azure Storage is a Microsoft-managed service providing cloud storage that is highly available, secure, durable, scalable, and redundant. Azure Storage includes Azure Blobs (objects), Azure Data Lake Storage Gen2, Azure Files, Azure Queues, and Azure Tables. The cost of your storage account depends on the usage and the options you choose below.
Learn more about Azure storage accounts

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *	Pay-As-You-Go
Resource group *	qde-rg Create new

Instance details

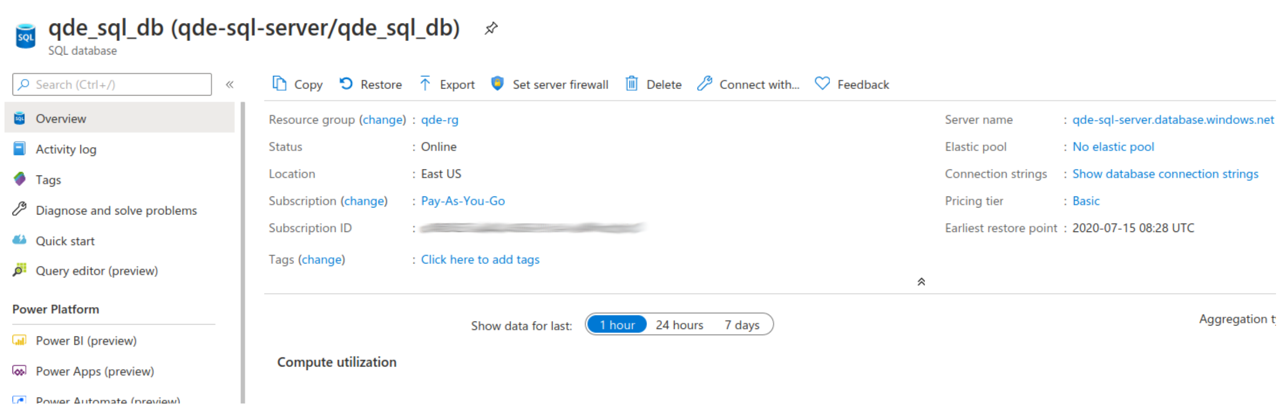
The default deployment model is Resource Manager, which supports the latest Azure features. You may choose to deploy using the classic deployment model instead. Choose classic deployment model

Storage account name *	qdestorageaccount
Location *	(US) East US
Performance	[✓ Standard] Premium
Account kind	StorageV2 (general purpose v2)
Replication	Locally-redundant storage (LRS)
Access tier (default)	[✓ Cool] Hot

Azure Data Lake Storage - Create Storage Account

3. SQL Server and Database

Our tutorial will start with data stored in a database on an SQL Server. Go to your resource group and add the SQL Database to it. You will need the password, so remember it or write it down. In the configuration, create a new SQL Server for the database to run on. A basic database with 2GB storage will be enough for our use case.



qde_sql_db (qde-sql-server/qde_sql_db)
SQL database

Resource group (change)	: qde-rg	Server name	: qde-sql-server.database.windows.net
Status	: Online	Elastic pool	: No elastic pool
Location	: East US	Connection strings	: Show database connection strings
Subscription (change)	: Pay-As-You-Go	Pricing tier	: Basic
Subscription ID	: [redacted]	Earliest restore point	: 2020-07-15 08:28 UTC
Tags (change)	: Click here to add tags		

Show data for last: [✓ 1 hour] 24 hours 7 days

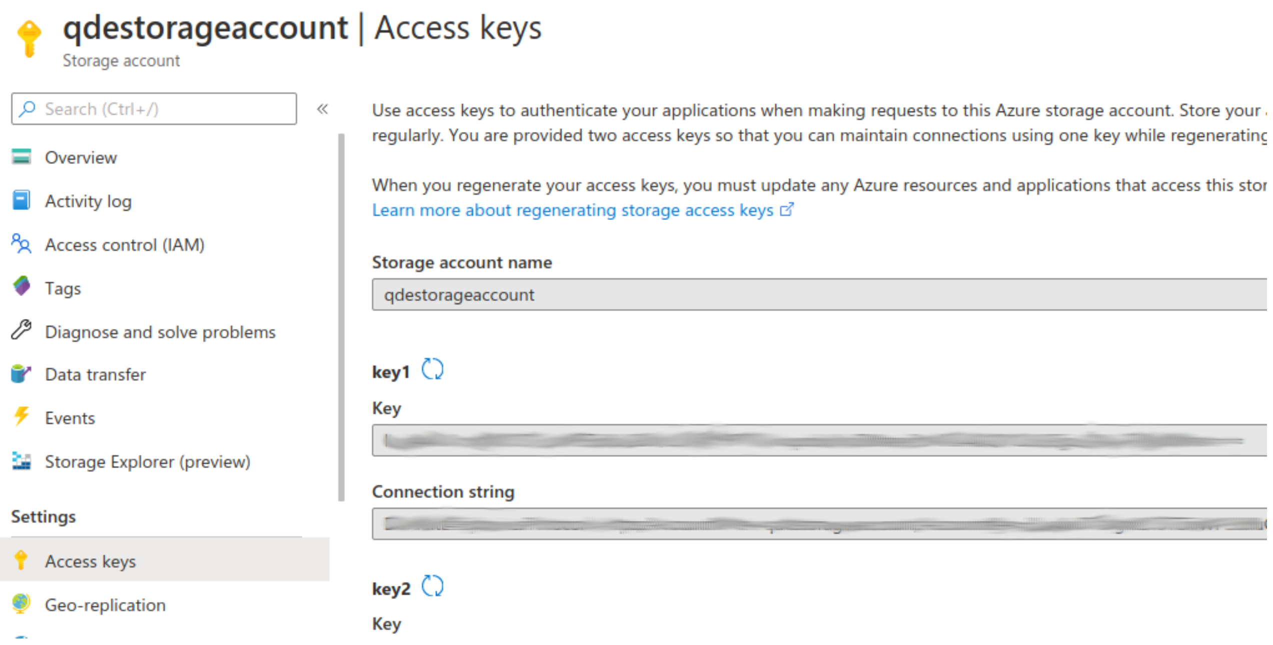
Aggregation t

Compute utilization

Data Engineering on Azure - SQL Server and Database

4. Key Vault

There are a few options to control access to blob storage and databases. For storage, you can use Access Keys, Shared Access Signatures, and the Azure Active Directory. Shared Access Signatures are a good option if you want to restrict access to individual objects or to a specific time period. Access with Azure AD works in combination with Role-Based Access Control (RBAC) and is evaluated when you are personally trying to access the storage. Also, it can be a good tool to grant access to applications over a service principal. For our case, we will grant access with the access keys. Those are basically master keys to the storage account, that will enable you to perform nearly all operations on the storage account, except special things like mounting data to databricks. Be aware that RBAC trumps the rights granted by Access Keys.



qdestorageaccount | Access keys
Storage account

Use access keys to authenticate your applications when making requests to this Azure storage account. Store your
regularly. You are provided two access keys so that you can maintain connections using one key while regeneratinc

When you regenerate your access keys, you must update any Azure resources and applications that access this stor
Learn more about regenerating storage access keys

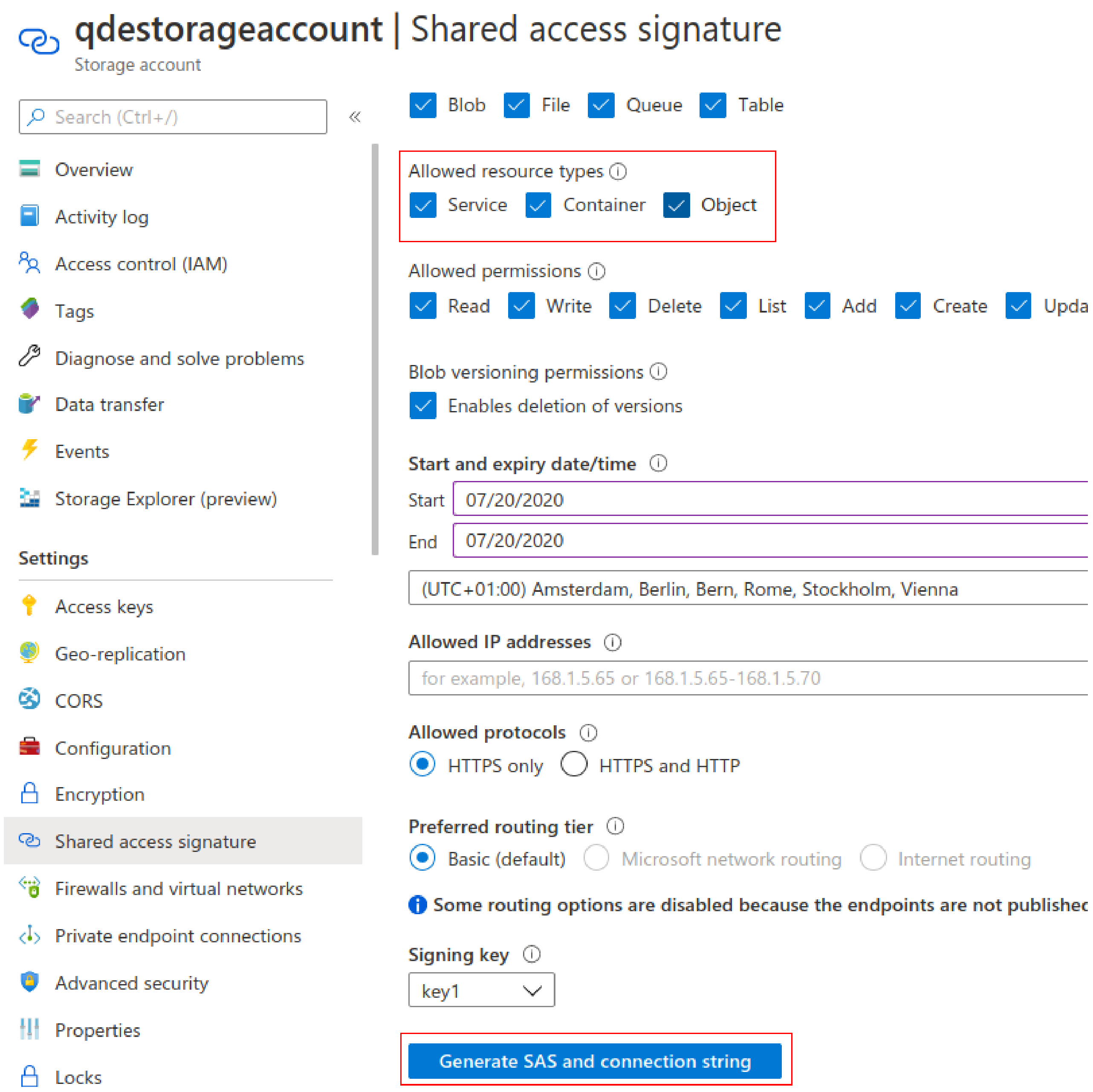
Storage account name: qdestorageaccount

key1
Key: [redacted]
Connection string: [redacted]

key2
Key

Data Engineering on Azure - Setting up the Resources - Access Keys

As a more secure alternative, you can also generate a Shared Access Signature on the whole storage account or for a subset of components. That signature can include different permissions like 'Read' or 'Write' and the validity can be restricted to a timeframe.



✓ Blob ✓ File ✓ Queue ✓ Table

Allowed resource types
✓ Service ✓ Container ✓ Object

Allowed permissions
✓ Read ✓ Write ✓ Delete ✓ List ✓ Add ✓ Create ✓ Upda

Blob versioning permissions
✓ Enables deletion of versions

Start and expiry date/time
Start 07/20/2020
End 07/20/2020
(UTC+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna

Allowed IP addresses
for example, 168.1.5.65 or 168.1.5.65-168.1.5.70

Allowed protocols
[✓ HTTPS only] HTTPS and HTTP

Preferred routing tier
[✓ Basic (default)] Microsoft network routing Internet routing

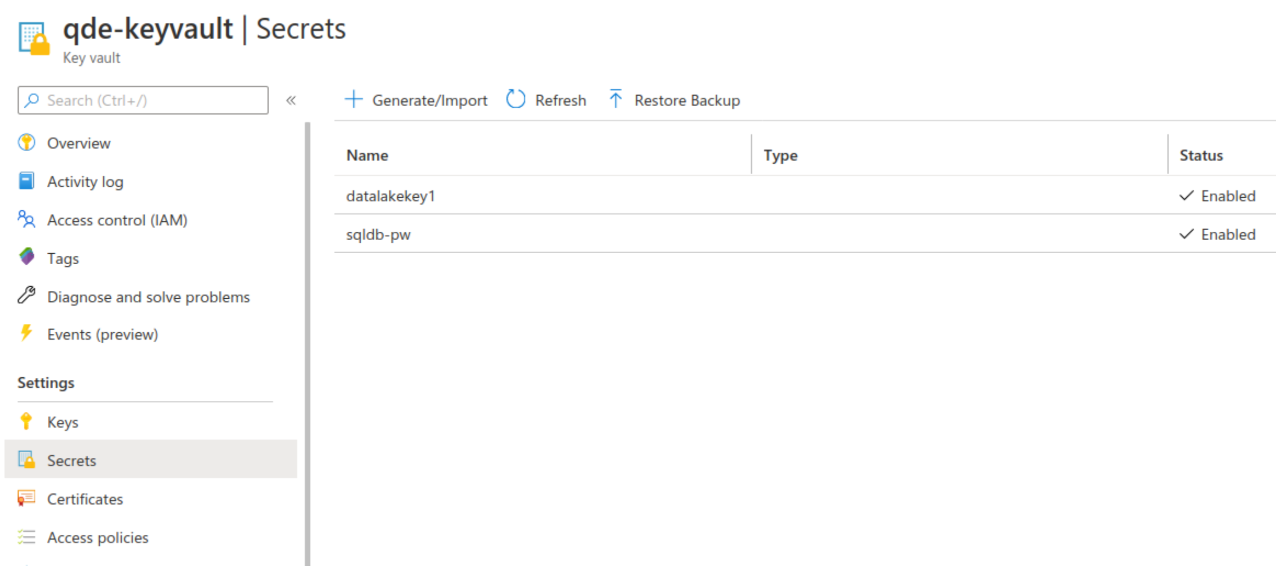
Some routing options are disabled because the endpoints are not publishec

Signing key
key1

Data Engineering on Microsoft Azure - Shared Access Signature
Access to the database can be granted for Azure AD identities and database users. We will use the database user credentials, more specifically the password. Since we do not want to enter our raw access data to connect to the resources, we will make use of a key vault. Go to your resource group and add a key vault to it. Once the key vault is up and running, open it and switch to the secrets tab. Generate/Import the two Secrets:

1. Containing the Access Key of your Storage Account

2. Containing your database user password



Name	Type	Status
datalakekey1		✓ Enabled
sqldb-pw		✓ Enabled

Data Engineering on Microsoft Azure - Key Vault - Secrets

Wrapping It Up

Now that we have setup the necessary resources, we will be able to build a pipeline that will transfer data from our SQL DB to the Data Lake storage. This concludes part 1 of our tutorial. See you on part 2 Basic ETL Processing with Azure Data Factory.

Sources

- Azure Docs: Resource Groups
- Azure Docs: Introduction to ADLS G2
- Azure Docs: Introduction to Blob Storage
- Azure Docs: Authorizing Access to Blob Storage
- Azure Docs: Key Vault Concepts
- Azure Docs: Create Azure SQL Database

For the next part of the tutorial click here.