

DATA() { EXPOSED;

DATA EXPOSED SPECIAL

Around the Clock with Azure SQL and Azure Data Factory

Americas

February 3, 2021

09:00 - 17:00 PT

Asia

February 4, 2021

09:00 - 17:00 SGT

16 Sessions | 2 Ask the Expert Panels | 1 Hackathon



HOSTED BY

Wee Hyong Tok & Anna Hoffman

DATA() {
EXPOSED;

Data security best practices for data engineers using Data Factory

Abhishek Narain
Senior Program Manager
@narainabhishek

Security considerations

1. Credential management

- Store in Azure Key Vault and reference from ADF during runtime
- Store in ADF service (always encrypted)

2. Authentication

- Windows, Basic, Service Principal, Managed identity

3. Data in transit

- Data encrypted in transit through HTTPS/TLS

4. Data at rest

- Leverage data store's encryption-at-rest capability: SQL TDE, Azure Blob/Table Storage SSE, etc

5. Network

- Managed Virtual Network – compute network isolation (data plane)
- ADF Private link – connect securely to ADF service using private endpoint (control plane)

6. Meta-data encryption (Meta-data at rest)

- ADF managed key encryption
- Customer managed key (CMK) encryption (double encryption)

Build password-free data pipelines

Preferred authentication mechanism –

If Data store == 'MSI' auth

Then Use MSI auth (AAD-based)

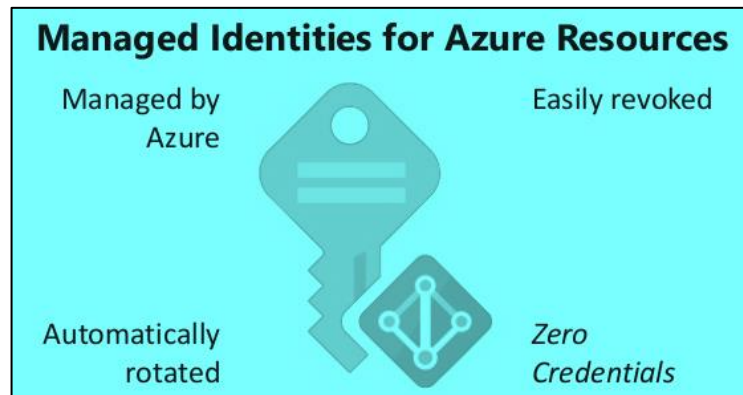
Else Use Key-Vault to store credentials & reference it in the Linked service

**ADF connects with Key-vault to retrieve the credentials during runtime using 'MSI' authentication*



Benefits of using MSI (Managed identity) authentication:

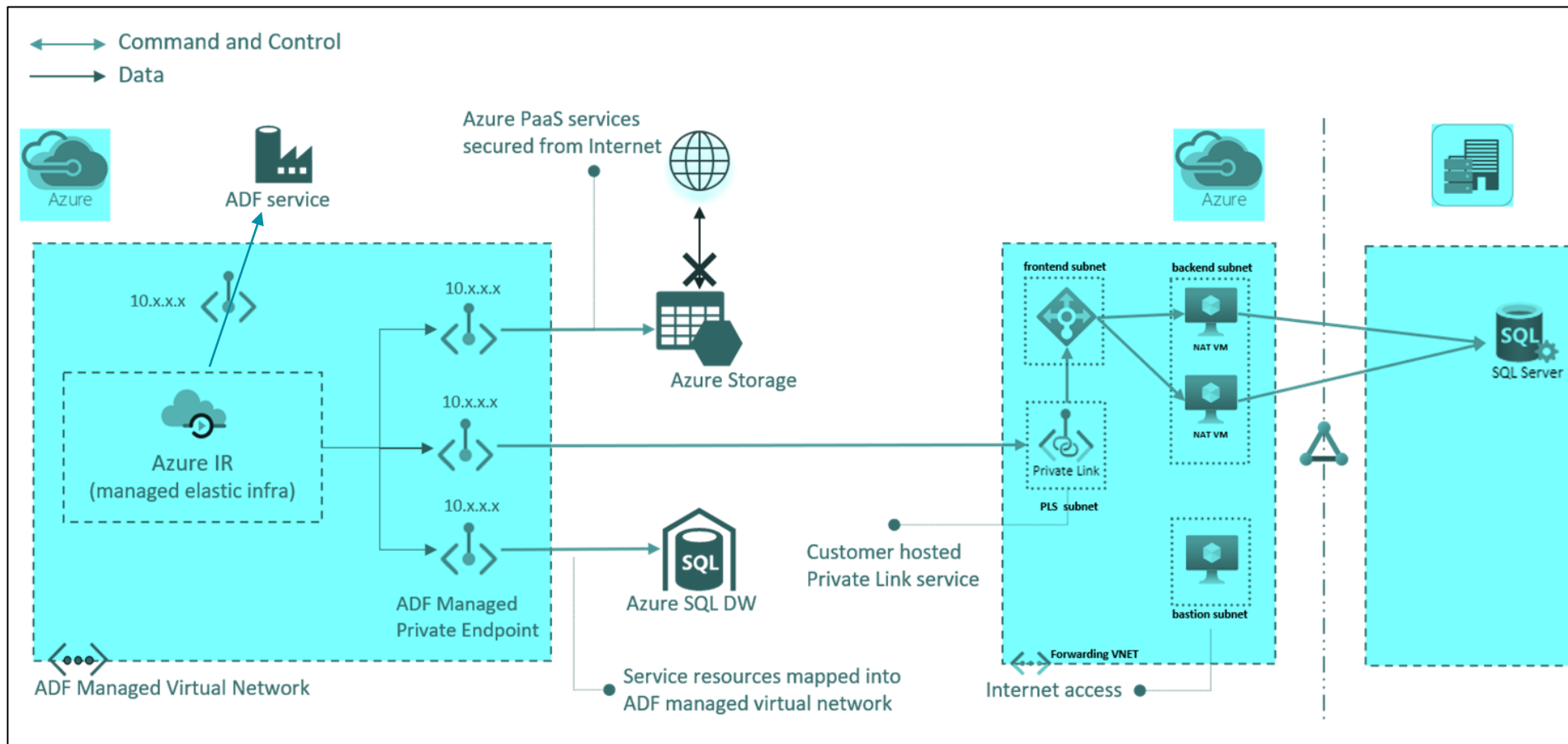
- Managed identities eliminate the need for data engineers having to manage credentials by providing an identity for the Azure resource in Azure AD and using it to obtain Azure Active Directory (Azure AD) tokens. In our case, Data Factory obtains the tokens using its Managed Identity and accesses the respective Data Store or Key Vault.
- It lets you provide fine-grained access control to a particular Data Factory instances using Azure AD.
- Since MSI is unique per Data Factory and is system-assigned there is no way someone can impersonate or use the resolved tokens causing security breaches.



Secure pipelines by running in virtual/ private network

Azure IR in Managed Virtual Network

- Private connection to data stores using Managed Private Network

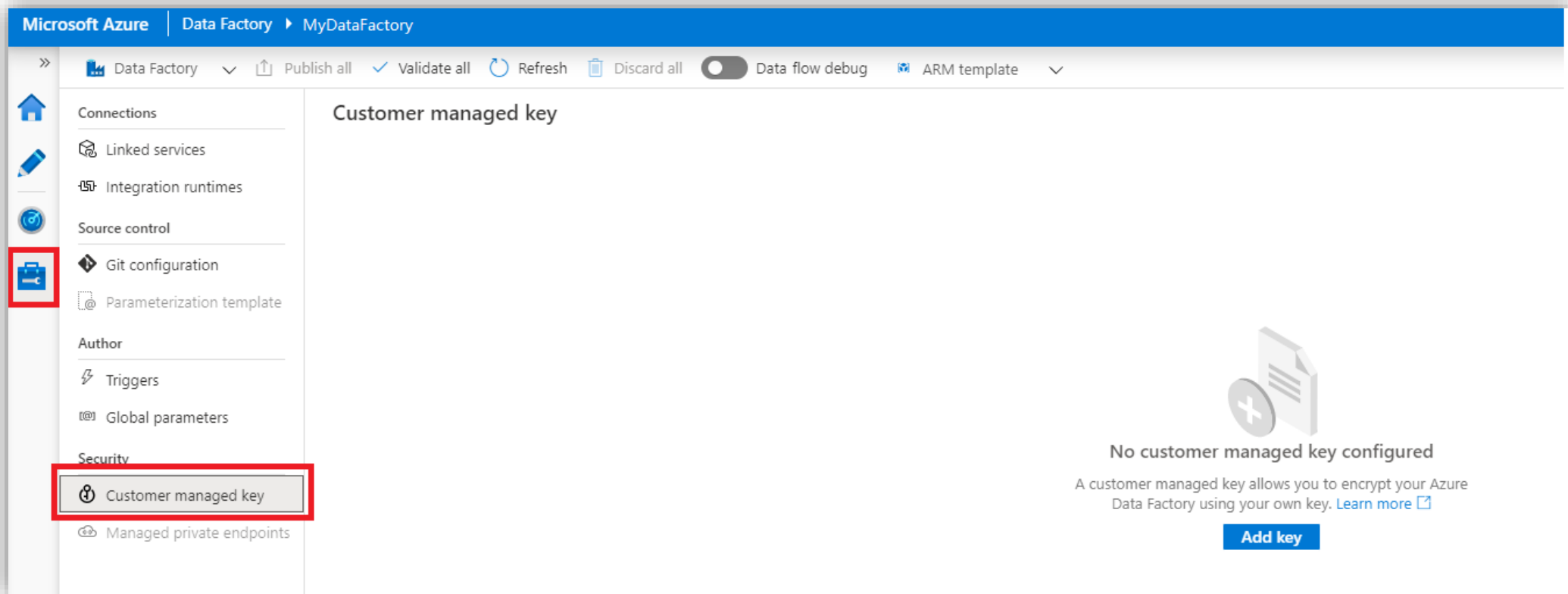


Encrypt ADF metadata using your key

Microsoft encrypts the meta-data using Microsoft managed keys

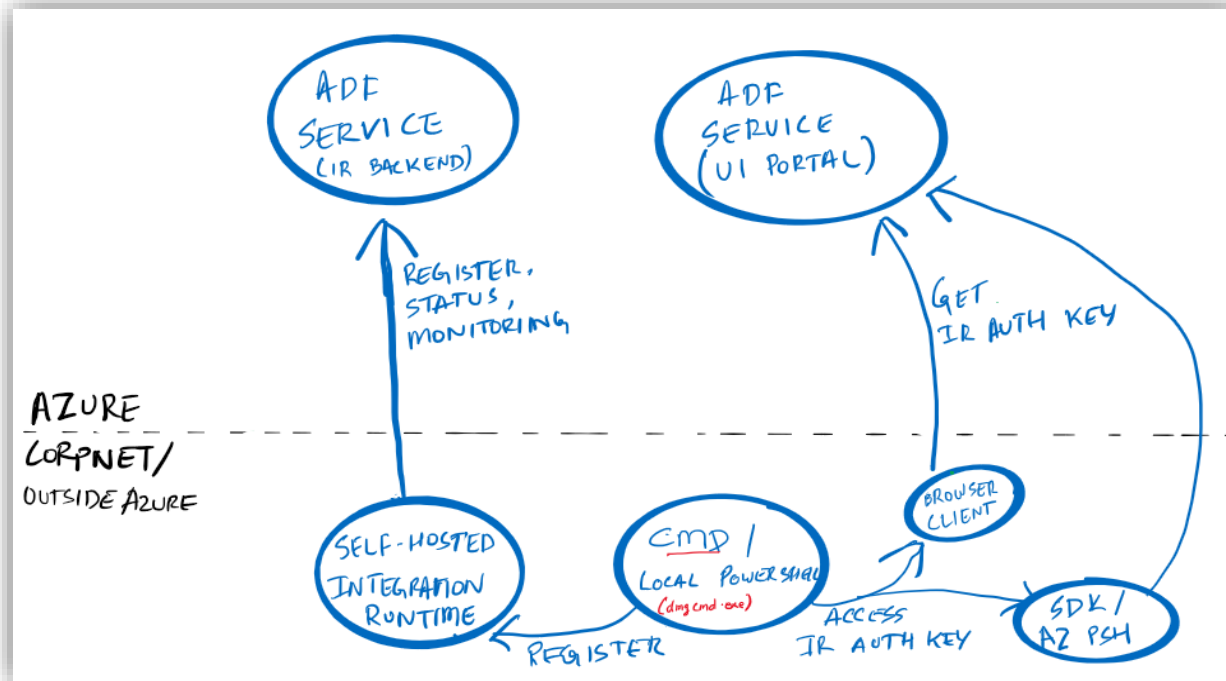
Customer can additionally encrypt the metadata with their own-keys (CMK)

[\[Encrypt Azure Data Factory with customer-managed key - Azure Data Factory | Microsoft Docs\]](#)



Self-hosted Integration runtime

Registration flow



Creation on Linked Service on SHIR

Creating new linked service (using self-hosted IR) and credential encryption flow

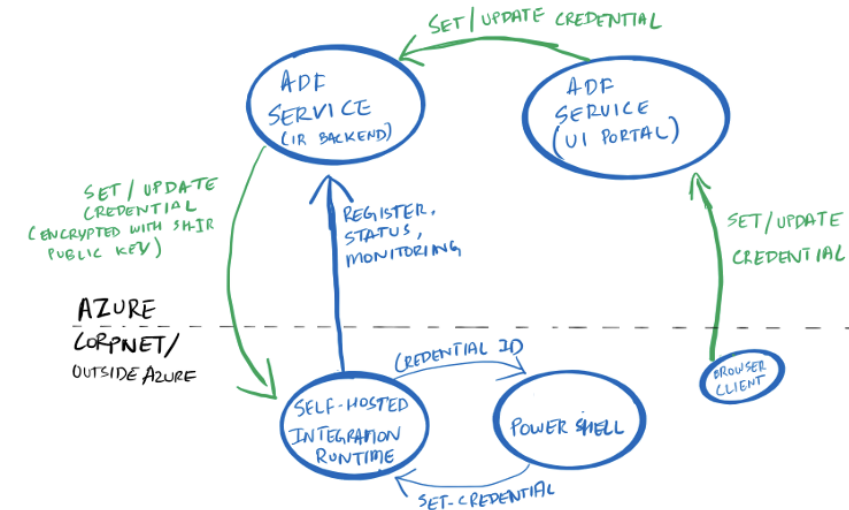


Fig 1.4, shows credential encryption workflow when using a self-hosted [IR](#)

Linked service payload encryption

	Payload before encryption (explicit encryption)	Payload after encryption (explicit encryption)
Dataset example	<pre>{ "properties": { "type": "AzureSqlTable", "typeProperties": { "tableName": { "type": "SecureString", "value": "dbo.SourceTable" } }, "linkedServiceName": { "referenceName": "SqlAzureLinkedService", "type": "LinkedServiceReference" } }, "name": "SqlAzureDataset" }</pre>	<pre>{ "properties": { "type": "AzureSqlTable", "typeProperties": { "tableName": { "type": "SecureString", "value": "xxxxxxxxxxxxxxxxxxxx", //Base64String } }, "linkedServiceName": { "referenceName": "SqlAzureLinkedService", "type": "LinkedServiceReference" } }, "name": "SqlAzureDataset" }</pre>

- You can additionally declare properties as secure string to encrypt the values

DATA() {
EXPOSED;

Learn with us!

View our on-demand playlist:
aka.ms/azuresqlandadf

@AzureSQL
@AzDataFactory



