Data Gathering and Data Injection- Security

Document version: Internal Draft

Background:

In our experience, very rarely do Data Scientists have a direct feed from production data to the systems/services/tools used to develop Machine Learning Models. Most Machine Learning models are developed on a subset (of a copy) that resembles the data (distributions) that the model will see in production. Typically, that subset of the data comes in a flat file format (.csv, xlsx, .json,.ndjson).

Security Preamble:

Before starting any project in Azure, it is important to understand the shared responsibility model and which security tasks are handled by the cloud provider and which tasks are handled by the customer. The workload responsibilities vary depending on whether the workload is hosted on Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS), or in an on-premises datacenter.

Division of responsibility

The following diagram illustrates the areas of responsibility between you and Microsoft, according to the type of deployment of your stack.

Figure 1 Share Responsibility Model

Regardless of the type of deployment, the following responsibilities are always retained by the customer:

- Data
- Endpoints
- Account
- Access management

Data protection

Data segregation: Azure is a multi-tenant service, which means that multiple customer deployments might be are stored on the same physical hardware. Azure uses logical isolation to segregate each customer's data from the data of others. Segregation provides the scale and economic benefits of multi-tenant services while rigorously preventing customers from accessing one another's data.

Customers delivering data science enabled capabilities to their customers might be required to build in data segregation into their applications/services.

At-rest data protection: Customers are responsible for ensuring that data stored in Azure is encrypted in accordance with their standards. Azure offers a wide range of encryption capabilities, giving customers the flexibility to choose the solution that best meets their needs. For example,

- Azure Key Vault helps customers easily maintain control of keys that are used by cloud applications and services to encrypt data.
- Azure Disk Encryption enables customers to encrypt Azure Virtual Machines.
- Azure Storage Service Encryption makes it possible to encrypt all data that's placed into a customer's storage account.

In-transit data protection: Customers can enable encryption for traffic between their own Azure services/VMs and end users.

Azure protects data in transit to or from outside components and data in transit internally, such as between two virtual networks. Azure uses the industry-standard Transport Layer Security (TLS) 1.2 or later protocol with 2,048-bit RSA/SHA256 encryption keys, to encrypt communications between:

- The customer and the cloud.
- Internally between Azure systems and datacenters.

Data destruction: When customers delete data or leave Azure, Microsoft follows strict standards for overwriting storage resources before their reuse, as well as the physical destruction of decommissioned hardware. Microsoft executes a complete deletion of data on customer request and on contract termination.

Data Science Sample Architecture

The diagram below is a representation of the recommended high-level architecture used to for Data Science projects in Azure.

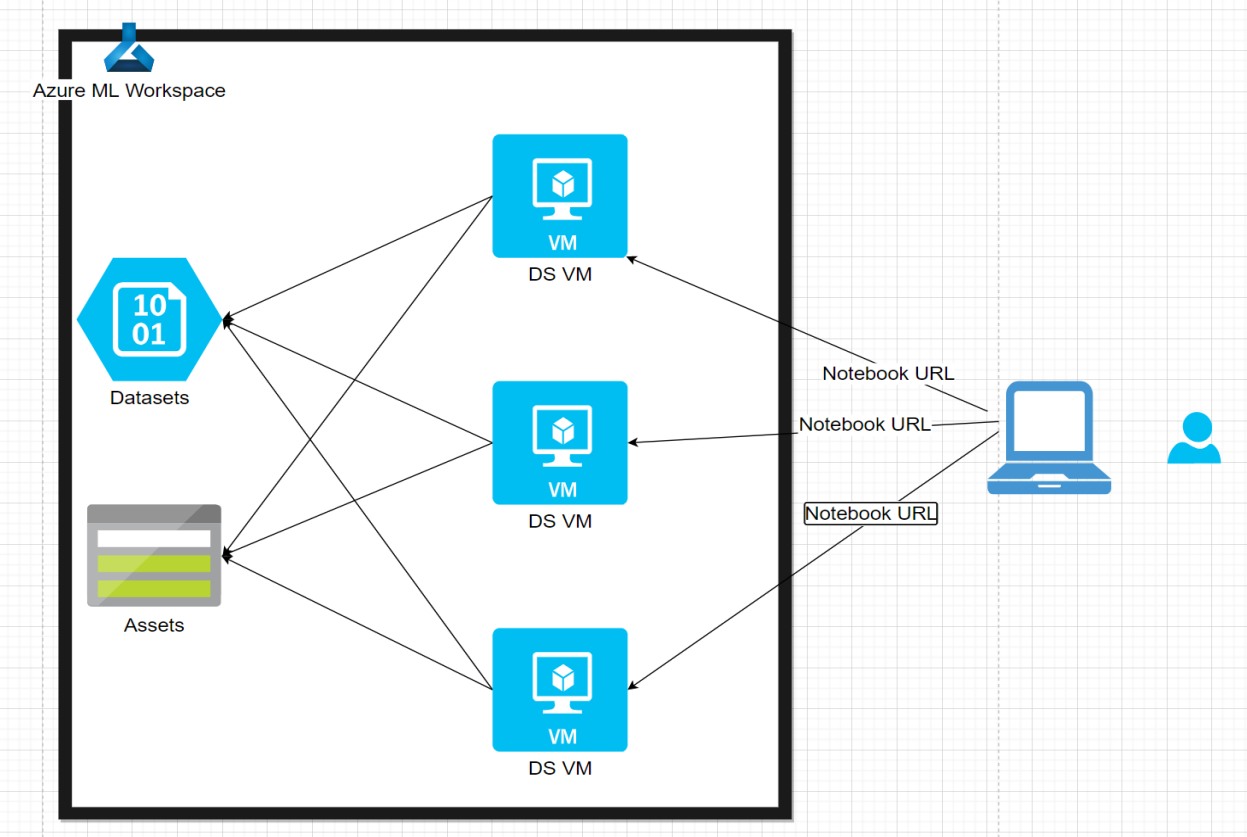
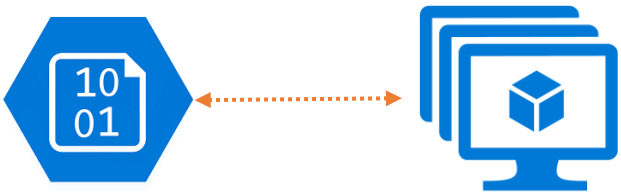


Figure 2 Data Science Reference Architecture

In the Data Science Lifecycle, there are several patterns for data be ingestion or access. Below are some examples of the same, and recommendations on how those interactions should be secured for the most frequently used data access pattern (Blob Storage + Virtual Machine). It also includes the relevant Azure Policy that IT Operations and Governance teams can leverage to enforce security controls across the organizations Azure estate:

1. Azure Storage account + Virtual Machine:

In this pattern, flat files are generally stored in a Blob Storage account or in a Data Lake, then data is accessed from the VM, where a local copy is created and used.



Security recommendations for Storage account:

Data protection

a. **Enable the Secure transfer required option on all of your storage accounts**

When you enable the **Secure transfer required** option, all requests made against the storage account must take place over secure connections. Any requests made over HTTP will fail. For more information, see Require secure transfer in Azure Storage.

b. **Enable advanced threat protection for all your storage accounts**

Advanced threat protection for Azure Storage provides an additional layer of security intelligence that detects unusual and potentially harmful attempts to access or exploit storage accounts. Security alerts are triggered in Azure Security Center when anomalies in activity occur and are also sent via email to subscription administrators, with details of suspicious activity and recommendations on how to investigate and remediate threats. For more information, see Advanced threat protection for Azure Storage.

c. **Limit shared access signature (SAS) tokens to HTTPS connections only**

Requiring HTTPS when a client uses a SAS token to access blob data helps to minimize the risk of eavesdropping. For more information, see Grant limited access to Azure Storage resources using shared access signatures (SAS).

d. **Turn on soft delete for blob data**

Soft delete enables you to recover blob data after it has been deleted. For more information on soft delete, see Soft delete for Azure Storage blobs.

e. **Where required by corporate data security policies or by regulatory requirements, Store business-critical data in immutable blobs**

Configure legal holds and time-based retention policies to store blob data in a WORM (Write Once, Read Many) state. Blobs stored immutably can be read but cannot be modified or deleted for the duration of the retention interval. For more information, see Store business-critical blob data with immutable storage.

Encryption

By default, Azure Storage encryption is enabled for all storage accounts Microsoft-managed keys. Based on your security policy or compliance requirements, you may rely on Microsoft-managed keys for the encryption of your data, or you can manage encryption with your own keys. For more information, see

https://docs.microsoft.com/en-us/azure/storage/common/storage-service-encryption?toc=/azure/storage/blobs/toc.json.

Access Management

f. Use Azure Active Directory (Azure AD) to authorize access to blob data

Azure AD provides superior security and ease of use over Shared Key for authorizing requests to Blob storage. For more information, see Authorize access to Azure blobs and queues using Azure Active Directory.

g. Ensure anonymous read access to containers and blobs is disabled

By default, a container, and any blobs within it may be accessed only by a user that has been given appropriate permissions. Ensure that Public read access is best for scenarios where you want certain blobs to always be available for anonymous read access and may not be appropriate for data science projects. For more information, see https://docs.microsoft.com/en-us/azure/storage/blobs/storage-manage-access-to-resources?tabs=dotnet.

h. Keep in mind the principal of least privilege when assigning permissions to an Azure AD security principal via RBAC

When assigning a role to a user, group, or application, grant that security principal only those permissions that are necessary for them to perform their tasks. Limiting access to resources helps prevent both unintentional and malicious misuse of your data.

i. Use a user delegation SAS to grant limited access to blob data to clients

A user delegation SAS is secured with Azure Active Directory (Azure AD) credentials and by the permissions specified for the SAS. A user delegation SAS is analogous to a service SAS in terms of its scope and function but offers security benefits over the service SAS. For more information, see Grant limited access to Azure Storage resources using shared access signatures (SAS).

j. Regenerate your account keys periodically

Rotating the account keys periodically reduces the risk of exposing your data to malicious actors.

Networking

k. Access to storage accounts should be restricted with firewall and virtual network configurations

It is a best practice to audit unrestricted network access in your storage account firewall settings. Configure network rules so only applications from allowed networks can access the storage account.

- To allow connections from specific Internet or on-premises clients, you can grant access to traffic from specific Azure virtual networks or to public Internet IP address ranges.
- To allow trusted Microsoft services to access the storage account, firewall rules for your storage account blocks incoming requests for data by default, unless the requests originate from a service operating within an Azure Virtual Network (VNet) or from allowed public IP addresses.
  Related Azure Policy: Audit unrestricted network access to storage accounts.

l. Use private endpoints

Azure Private Link enables you to access Azure PaaS Services (for example, Azure Storage and SQL Database) and Azure hosted customer-owned/partner services over a private endpoint in your virtual network. Traffic between your virtual network and the service travels the Microsoft backbone network eliminating the need for your service to be exposed to the public internet. Azure Private Endpoint is a network interface that connects you privately and securely to a service powered by Azure Private Link. Private Endpoint uses a private IP address from your VNet, effectively bringing the service into your VNet. For more information about private endpoints, see Connect privately to a storage account using Azure Private Endpoint.

m. Use VNet service tags

A service tag represents a group of IP address prefixes from a given Azure service. You can use service tags to achieve network isolation and protect your Azure resources from the general Internet while accessing Azure services that have public endpoints. For more information on how to get started with VNet service tags, see https://docs.microsoft.com/en-us/azure/virtual-network/service-tags-overview.

Logging and Monitoring

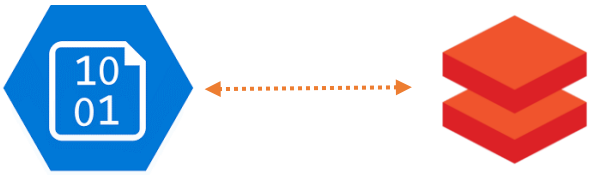
n. Enable Azure Storage logging

Enable Azure Storage logging to track how each request made against Azure Storage was authorized. These logs help track access to Blob storage and whether a request was made anonymously, by using an OAuth 2.0 token, by using Shared Key, or by using a shared access signature (SAS). This is key in identifying malicious access attempts.

Additional data access patterns

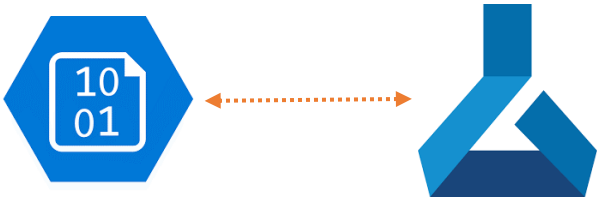
2. Blob Storage + Azure Databricks:

In this patter, flat files are stored in a Blob Storage account or in a Data Lake, then data is accessed from an Azure Databricks Workspace. In here the file is not permanently copied, you can create a mount point and access the data as if it located on the local storage, this is enabled by the use of pointers.



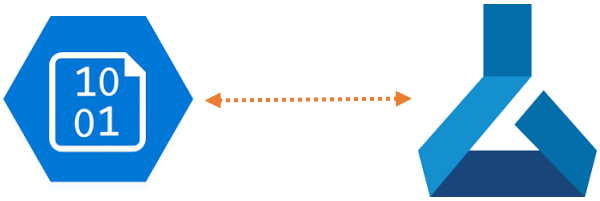
3. Blob Storage + Azure Machine Learning Service Workspace (As File):

In this pattern, flat files stored in a Blob Storage account or in a Data Lake, then the data is registered and uploaded as a dataset into Azure Machine Learning Services. This data will later be accessible from notebook VMs and pipelines created in the same workspace. In this method, the same file will be present in 3 different locations. One advantage is that this allows to have data versioning.



4. Blob Storage + Azure Machine Learning Service Workspace (As Table):

In this pattern, flat files stored in a Blob Storage account or in a Data Lake, then the data is registered and uploaded as a dataset into Azure Machine Learning Services. **If the file has tabular data on it, the data can be defined as table and would be accessed as it was a SQL table**. This data will later be accessible from notebook VMs and pipelines created in the same workspace. In this method, the same file will be present in **2** different locations. One advantage is that this allows to have data versioning.



5. Azure Machine Learning Service Workspace (As Table): (blob storage)

In this pattern, data can be uploaded directly to the Azure Machine Learning workspace. The number of copies on different systems, will depend on whether you register the data as a table or as a file. One advantage is that this allows to have data versioning.

6. Direct Database or Data Warehouse

In some cases, there will be an olap database or data warehouse designed for analytics workflows, in which case teams, might be connecting directly to it. This workflow also supports AzureML Datastores if it is an Azure SQL or Azure Synapse Analytics source. Otherwise, the best practice would be copying the data as a file into Data Lake via Data Factory. Depending on the needs, this can be done as a one-time copy, batched, or streamed changes.

*We advise strongly against having data scientists developing against production databases. Data should be integrated and ingested into the enterprise data lake and data scientists should work from there. If you don’t have a data lake, then making copies of the data and putting them into a storage account for your data science teams is a good interim workflow.