**SCENARIO**

Customer is looking for an Azure solution which can function like a traditional FTP Server. The customer (and some of their own vendors) need a place to store .CSV files. These .CSV files will then be accessed by a SaaS application, uploaded into that system, and processed. The customer has the following additional requirements/asks:

* 60 Days of file retention
* .CSV files must be encrypted at rest and in transit
* Solution needs to be robust (highly available)
* How to backup?
* Auditing access to files (users, source IP address, name of files)

**SCENARIO ARCHITECTURE**

* Solution: Utilize Azure Blob Storage to store the .CSV Files. Utilize Azure Logic Apps or Azure Automation Runbook to automatically delete files after the 60-day period. Recommended to use Azure Logic Apps. Lifecycle Management, a built-in service to provide this functionality is currently in preview.
* How does this architecture meet the customer requirements?
  + Azure Blob Storage is ideal for storing files for distributed access
  + Azure Storage Accounts can be configured to require HTTPS connections
  + Azure Storage Accounts use Storage Service Encryption to automatically encrypt all data before persisting it. Encryption utilizes 256-bit AES Encryption and the entire process is transparent to all users and requires no additional setup or configuration.
  + Azure Storage Accounts support several different replication policies to ensure high-availability. Listed lowest availability and cost to highest.
    - Locally Redundant – Stores 3 copies of the data in a single storage cluster in a single datacenter. Offers 11 9’s of availability over the course of year.
    - Zone Redundant – Stores 3 copies of the data across different storage clusters in different Availability Zones in a single region. Offers 12 9’s of availability over the course of the year.
    - Geo Redundant – Data is replicated to a secondary region hundreds of miles away from the primary region. Offers 16 9’s of availability over the course of the year.
      * Offers an option for Read-Access which always makes the secondary site available to read from
  + Azure Storage Soft Delete enables you to save and recover your data after it was deleted (intentionally or unintentionally). In this scenario this would be set for 60 days.
  + Users or client applications can access Azure Blog Storage via HTTPS from anywhere in the world via URLs, PowerShell, Azure CLI and the Azure Storage Client Libraries for various programming languages
  + Supports Identity and Access Management via Azure AD
  + All auditing capabilities can be met by enabling Azure Storage Analytics Logs and Azure Log Analytics
  + Azure Storage Explorer can be utilized to upload and download files as well as manage the storage account from within the Azure portal or via the stand-alone Azure Storage Explorer
* What are some limitations of this proposed architecture?
  + Azure Storage Accounts don’t support the ability to specify multiple logins to different containers or to the same container
  + Unable to map a drive to the Azure Storage Account or the Azure Blob Container where the files are stored