**Scenario 1: Using two Azure regions for business continuity with minimal downtime**

1. **Using Azure SQL Database for business continuity**

Auto-failover groups feature in Azure SQL provides a powerful abstraction of active geo-replication by supporting group level replication and automatic failover. It removes the necessity to change the SQL connection string after failover by providing the additional listener end-points. More details about Active geo-replication and auto-failover groups can found [here](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-geo-replication-overview).

1. **How it works**

We will be using two Azure regions for business continuity with minimal downtime in this approach. When setup Auto-failover groups, Azure SQL will have following endpoints.

* + 1. <failover-group-name>.database.windows.net (Primary region)
    2. < failover-group-name>.secondary.database.windows.net (Secondary region)

After an outage in the primary region, Azure SQL detects that the primary database is not accessible and triggers failover to the secondary region. All transactions committed after the failover are lost during the reconnection of application to Azure SQL. After the failover is completed, the application can connect and restart processing the application requests. When the failed region is restored and back online, it reconnects as a new secondary region.

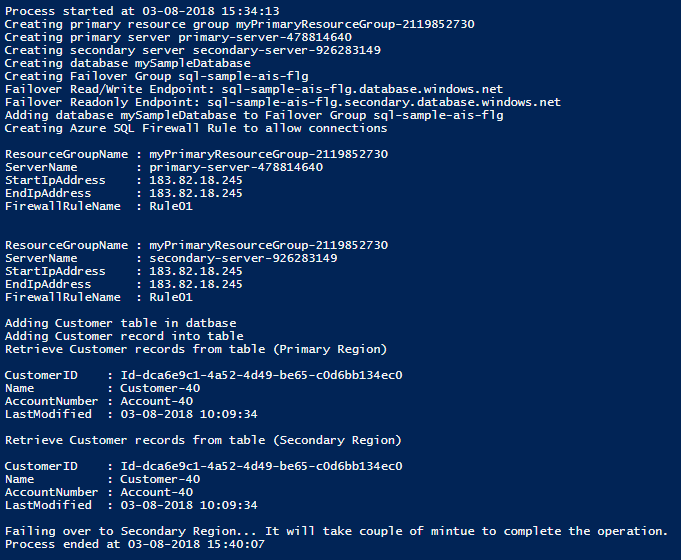
If an outage happens in secondary region, the replication process between the primary and the secondary database gets suspended. Once the outage is mitigated, the secondary database automatically resynchronizes with the primary. During synchronization, performance of the primary can be impacted

1. **How to run the sample**

The sample contains two PowerShell scripts

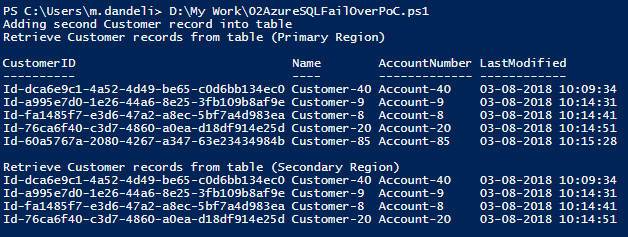
* **01AzureSQLFailOverPoC.ps1**

This script will create all the required infrastructure in Azure including SQL Servers and Failover Groups and inserts a new row in sample table. After inserting sample data, it will do a failover to secondary region. Following image shows sample output of the script execution.



* **02AzureSQLFailOverPoC.ps1**

This script need to be executed once failover operation is completed. It also inserts data rows into sample table and retrieves data from both regions. Please note, this script uses same Failover Read/Write Endpoint which first script uses, which will demo that applications don’t have change data connection strings after failover operation. Following image shows sample output of the script execution.



1. **More information**
   * 1. [Designing globally available services using Azure SQL Database](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-designing-cloud-solutions-for-disaster-recovery)
     2. [Overview: Active geo-replication and auto-failover groups](https://docs.microsoft.com/en-us/azure/sql-database/sql-database-geo-replication-overview)