DATAZEN LAB

SIMPLE AWS 2 AZURE SQL SERVER CDC REPLICATION – ~30 minutes

This document provides an overview of how to implement a secured Change Data Capture (CDC) SQL Server database replication topology from AWS into Azure using Enzo DataZen as the technology platform, using a single DataZen instance in AWS.

Pre-Requisites

To successfully complete this lab, you will need the following:

- An AWS account
- A Microsoft Azure account

Overview

In this lab we will configure two SQL Server databases (one in AWS and another in Azure using the Azure SQL Database service), configure Change Data Capture (CDC) in AWS, configure the target database, and observe data flowing between the two cloud platforms.

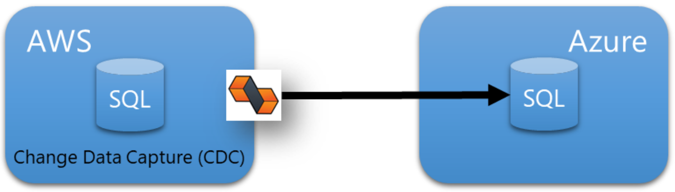


Figure 1 - Logical Replication Topology using DataZen between AWS and Azure

NOTE: Some of the technology choices were made to simplify the lab, including the versions of the source and target databases. Additionally, DataZen can forward changes to any target system or systems; this lab uses SQL Server as the target database for simplicity. In addition, this lab uses the Synthetic Change Data Capture logic from DataZen instead of SQL Server's built-in CDC because SQL Server CDC is not support by SQL Server Express edition used in this lab.

Technologies Used

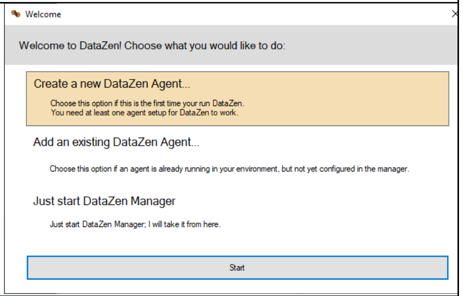
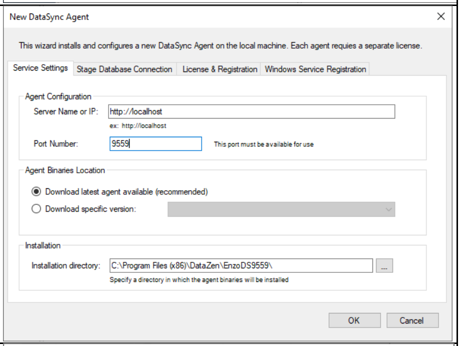
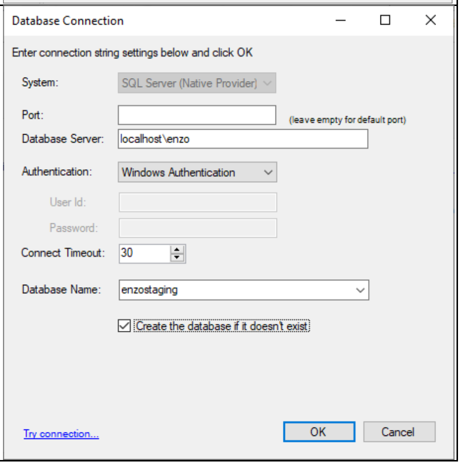
The following technologies are used in this lab:

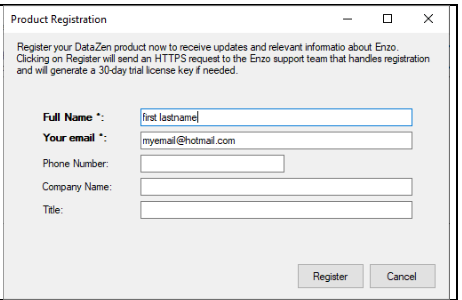
Cloud	Technology	Notes
AWS	EC2 Instance	Use the AWS Marketplace to create an Enzo Server EC2 instance that contains DataZen
	SQL Server	The EC2 image will contain a SQL Server database
	DataZen	DataZen will be installed on the same EC2 instance
Azure	Azure SQL Database	Use an Azure SQL Database instance to push changes detected in AWS

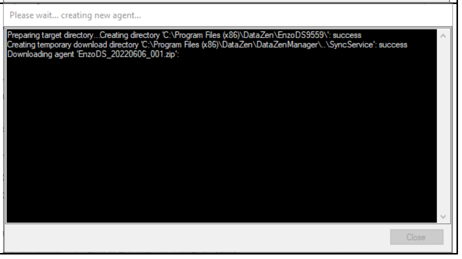
Configure AWS

Install DataZen on EC2 Server

Follow these steps to create a new EC2 instance from the AWS Marketplace.

Step	Description	Details
Login to the AWS portal	Navigate to your AWS portal and login using a privilege account	
Create an EC2 image from the Enzo Server Marketplace	Navigate to https://aws.amazon.com/marketplace/pp/prodview-mdxtczpoykjyk *It is highly recommended to select a Medium instance size or higher.* *Make sure you enable a security rule that allows RDP connections from your computer.*	
Follow instructions to connect to the Instance	Connect to the EC2 Instance that was just created using an RDP connection. Once logged on, the VM may need a few minutes to download a few packages automatically. *Once you see Enzo icons on the desktop, you are ready to proceed to the next step.*	
Starts DataZen and choose Create New Agent	To start DataZen, right-click on the DataZen icon and choose **Run as Administrator**.	 Welcome Welcome to DataZen! Choose what you would like to do: Create a new DataZen Agent... Choose this option if this is the first time your run DataZen. You need at least one agent setup for DataZen to work. Add an existing DataZen Agent... Choose this option if an agent is already running in your environment, but not yet configured in the manager. Just start DataZen Manager Just start DataZen Manager; I will take it from here. Start
Verify the information on the General Settings tab	You can leave all the defaults provided on the General Tab. By default, a new DataZen Agent will be started on the local machine and will listen on port 9559; this is the port DataZen manager uses to control the agent.	 New DataSync Agent This wizard installs and configures a new DataSync Agent on the local machine. Each agent requies a separate license. Service Settings | Stage Database Connection | License & Registration | Windows Service Registration Agent Configuration Server Name or IP: http://localhost ex: http://localhost Port Number: 9559 This port must be available for use Agent Binaries Location ● Download latest agent available (recommended) ○ Download specific version: Installation Installation directory: C:\Program Files (x86)\DataZen\EnzoDS9559\ Specify a directory in which the agent binaries will be installed OK Cancel
Click on Stage Database Connection	Enter the following information, then click OK: Database Server: **localhost\enzo** Authentication: **Windows Auth** Database Name: **enzostaging** Check the **Create Database if it doesn't exist** option	 Database Connection Enter connection string settings below and click OK System: SQL Server (Native Provider) Port: (leave empty for default port) Database Server: localhost\enzo Authentication: Windows Authentication User Id: Password: Connect Timeout: 30 Database Name: enzostaging ☑ Create the database if it doesn't exist Try connection... OK Cancel

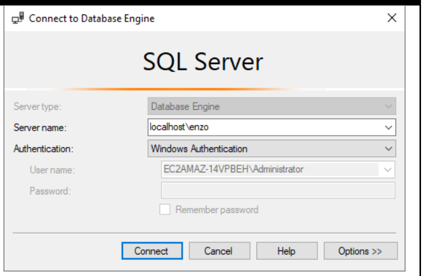
Click on the License tab and enter your license key	Click on the **Generate Free 30-day License** button. Enter your contact information in the **Product Registration** window and click **Register**.	 Product Registration Register your DataZen product now to receive updates and relevant informatio about Enzo. Clicking on Register will send an HTTPS request to the Enzo support team that handles registration and will generate a 30-day trial license key if needed. **Full Name ***: first lastname **Your email ***: myemail@hotmail.com Phone Number: Company Name: Title: Register Cancel

WAIT A FEW MINUTES	The agent installation and configuration will start automatically as a separate window.	
		Please wait... creating new agent...
		`Preparing target directory...Creating directory 'C:\Program Files (x86)\DataZen\EnzoDS9559\': success`
		`Creating temporary download directory 'C:\Program Files (x86)\DataZen\DataZenManager\..\SyncService': success`
		`Downloading agent 'EnzoDS_20220606_001.zip':`
		Close

At this point, DataZen has been installed on the AWS EC2 Server and ready to be configured for replication.

Create a Test Database

Let's create a test database that will be used for replication.

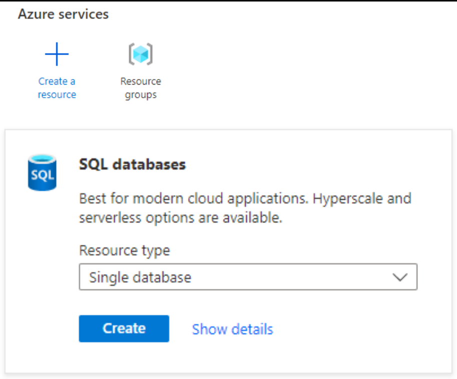
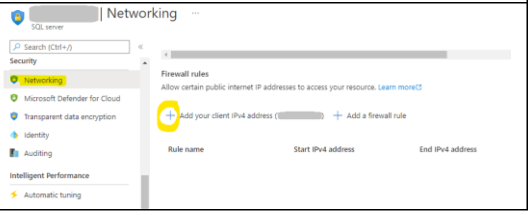
Step	Description	Details
Start SSMS	On the EC2 Instance, double-click on the SQL Server Management Studio (SSMS) shortcut and connect to the SQL Server. Server Name: **localhost\enzo** Authentication: **Windows Authentication** *It may take a few minutes for SSMS to start depending on the Instance size* Once logged on, click on New Query.	 Connect to Database Engine SQL Server Server type: Database Engine Server name: localhost\enzo Authentication: Windows Authentication User name: EC2AMAZ-14VPBEH\Administrator Password: Remember password Connect Cancel Help Options >>

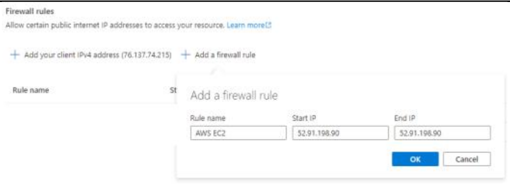
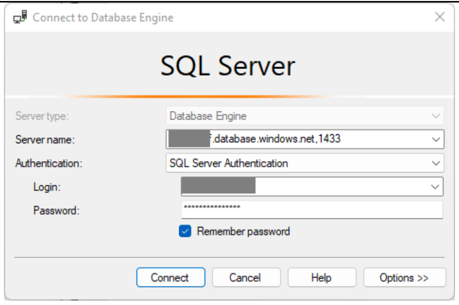
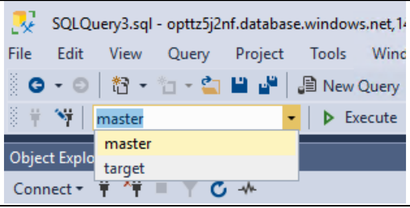
Create a Test database and table	Run the following SQL command to create the test database and table:
	
	`CREATE DATABASE test`
	`GO`
	
	`CREATE TABLE test.dbo.source (`
	`         id int identity(1,1) PRIMARY KEY,`
	`         stateCode nvarchar(2) NOT NULL,`
	`         stateDesc nvarchar(100) NOT NULL`
	`         )`
	`GO`	

Insert a few records	Let's add three initial records to see the [source] table with initial data:
	
	`INSERT INTO test.dbo.source`
	`VALUES ('FL', 'Florida'), ('CA', 'California'), ('NY', 'New York')`	

Create an Azure SQL Database

Next, let's create a new Azure SQL Database in Microsoft Azure and configure its firewall settings.

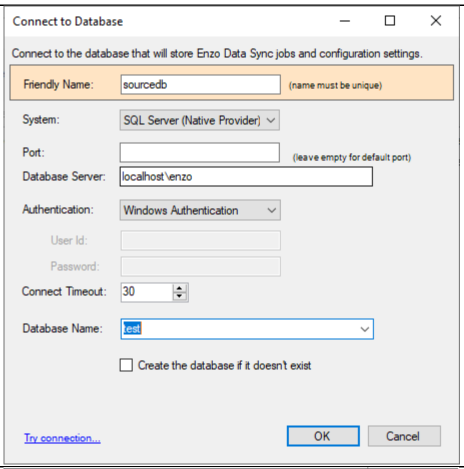
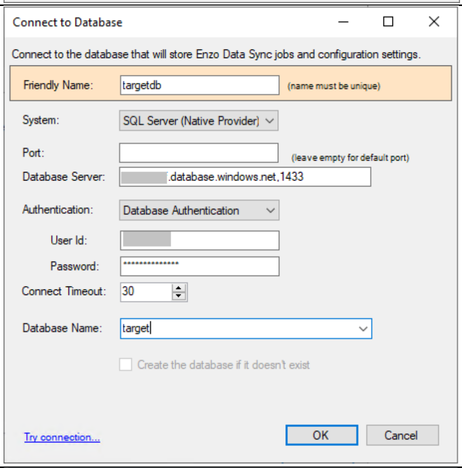
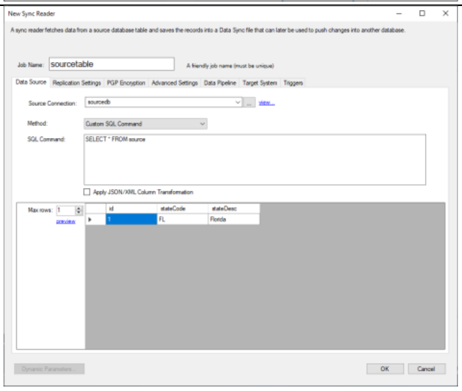
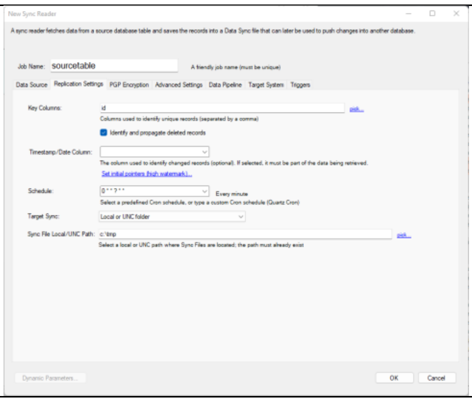
Step	Description	Details
Login to Azure	Open a browser and navigate to the Azure portal, then login. https://portal.azure.com	
Create an Azure SQL Database	From the Azure Portal, click on Create a Resource. For complete instructions on how to create a Single Azure SQL Database, follow Microsoft's documentation here: https://docs.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart?view=azuresql&tabs=azure-portal Search for Azure SQL, and click on it, then choose **Create**. Then, choose **SQL Database** / **Single Database** and click **Create**. **Enter the following information:** **Resource Group**: (leave default) **Database Name**: target **Server**: (create new if needed) **Workload Env**: development **On the Networking Tab:** Add current IP Address: **Yes** **On the Security Tab:** Enable Microsoft Defender: **Not now** **Click on Create**	 Azure services Create a resource Resource groups SQL databases Best for modern cloud applications. Hyperscale and serverless options are available. Resource type: Single database
Find you Database Server connection string	Once the database has been created, navigate to the Overview tab, and click on the server name. The server name should look something like this: *abcdefghi*.database.windows.net,1433 You are now on the Database Server page.	
Locate Server Firewall Rules	Find the Fire Rule options, located under the Networking tab. If you already see a rule that allows connection from the EC2 server's IP Address, skip the next step.	[redacted] | Networking SQL server Search (Ctrl+/) Security Networking Microsoft Defender for Cloud Transparent data encryption Identity Auditing Intelligent Performance Automatic tuning Firewall rules Allow certain public internet IP addresses to access your resource. Learn more Add your client IPv4 address ([redacted]) Add a firewall rule Rule name | Start IPv4 address | End IPv4 address

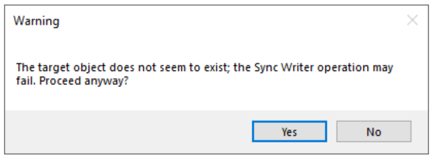
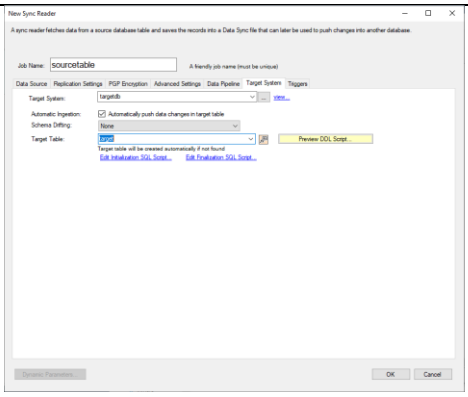
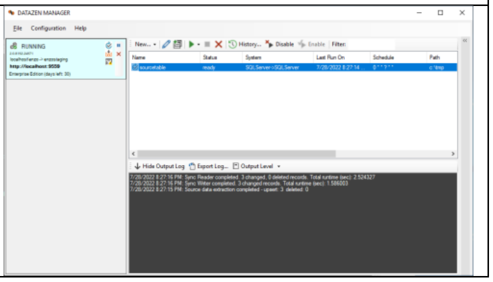
Add EC2 Firewall Rule	Add a firewall rule manually on the SQL Database Server itself so that your EC2 instance can connect to Azure SQL.	 Firewall rules Allow certain public internet IP addresses to access your resource. Learn more Add your client IPv4 address (76.137.74.215) Add a firewall rule Rule name: AWS EC2 — Start IP: 52.91.198.90 — End IP: 52.91.198.90
Connect using SSMS	Now that your Azure SQL Database is running and configured, connect to it using SSMS from your EC2 Instance. Click on New Query, then right-click inside the query window, and select **Connection -> Change Connection**. **Server Name**: The Azure server name **Authentication**: SQL Server Authentication **Login**: Your Azure SQL Server login name **Password**: Your Azure SQL password *IF YOUR EC2 Instance PUBLIC IP ADDRESS WAS NOT ADDED TO THE FIREWALL RULES IN PRIOR STEPS YOU WILL GET AN ERROR.*	 Connect to Database Engine SQL Server Server type: Database Engine Server name: [redacted].database.windows.net,1433 Authentication: SQL Server Authentication Login: [redacted] Password: **************** ☑ Remember password
Select the target database	Once connected, select the **target** database from the database dropdown. You are now connected to the target database from the EC2 Instance.	 SQLQuery3.sql - opttz5j2nf.database.windows.net,1 master master target

Configure DataZen for Replication

To finalize the source configuration, the next step is to configure DataZen to capture the changes identified in the source table, and create a Change Log file that will be stored in an Azure Blob (note other target stores are possible, including AWS Bucket, and FTP locations).

Step	Description	Details
Configure Connections in DataZen	Next, we need to configure DataZen's database connections. On the AWS EC2 instance, using DataZen Manager, click on the DataZen agent on the left and choose **Configuration -> Central Connection Strings**.	

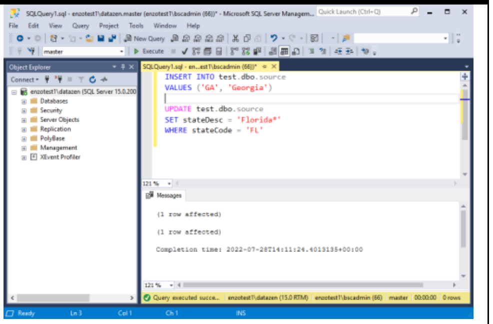
Create the source SQL Server Connection	From the Central Connection Strings window, choose **New -> Database Connection.** Enter a name for this connection (ex: **sourcedb**) and fill out the rest of the information needed to connect to SQL Server: Source System: **SQL Server** Database Server: **localhost\enzo** Authentication: **Windows Authentication** Database Name: **test** Click OK.	 Connect to Database Connect to the database that will store Enzo Data Sync jobs and configuration settings. Friendly Name: sourcedb (name must be unique) System: SQL Server (Native Provider) Port: (leave empty for default port) Database Server: localhost\enzo Authentication: Windows Authentication User Id: Password: Connect Timeout: 30 Database Name: test ☐ Create the database if it doesn't exist Try connection... OK Cancel
Create the target SQL Server Connection	Choose **New -> Database Connection** again. Enter a name for this connection (ex: **targetdb**) and fill out the rest of the information needed to connect to SQL Server: Source System: **SQL Server** Database Server: **the Azure SQL Server** Authentication: **Database Authentication** User Id: **You Azure SQL login id** Password: **Your Azure SQL Login password** Database Name: **target** Click OK, and close the Central Connection Strings window.	 Connect to Database Connect to the database that will store Enzo Data Sync jobs and configuration settings. Friendly Name: targetdb (name must be unique) System: SQL Server (Native Provider) Port: (leave empty for default port) Database Server: [redacted].database.windows.net,1433 Authentication: Database Authentication User Id: [redacted] Password: ************** Connect Timeout: 30 Database Name: target ☐ Create the database if it doesn't exist Try connection... OK Cancel
Create a new Sync Job	Click on **New -> New Data Sync.** Enter the following information: Job Name: **sourcetable** Source Connection: **sourcedb** Method: **Custom SQL Command** SQL Command: **SELECT * FROM test.dbo.source** **Click Preview…**	 New Sync Reader A sync reader fetches data from a source database table and saves the records into a Data Sync file that can later be used to push changes into another database. Job Name: sourcetable A friendly job name (must be unique) Data Source | Replication Settings | PGP Encryption | Advanced Settings | Data Pipeline | Target System | Triggers Source Connection: sourcedb view... Method: Custom SQL Command SQL Command: SELECT * FROM source ☐ Apply JSON/XML Column Transformation Max rows: 1 preview id | stateCode | stateDesc 1 | FL | Florida Dynamic Parameters... OK Cancel
Select the Replication Settings Tab	Enter the following information: Key Columns: **id** Schedule: **0 * * ? * *** Identify & Propagate Deleted Records: **Yes** Target Sync: **Local or UNC Folder** Sync File: **c:\tmp** *NOTE: In production environments, when large volumes of data changes are expected, it is best to use Change Data Capture (CDC) or Change Tracking for improved performance.*	 New Sync Reader A sync reader fetches data from a source database table and saves the records into a Data Sync file that can later be used to push changes into another database. Job Name: sourcetable A friendly job name (must be unique) Data Source | Replication Settings | PGP Encryption | Advanced Settings | Data Pipeline | Target System | Triggers Key Columns: id pick... Columns used to identify unique records (separated by a comma) ☑ Identify and propagate deleted records Timestamp/Date Column: The column used to identify changed records (optional). If selected, it must be part of the data being retrieved. Set initial pointers (high watermark)... Schedule: 0 * * ? * * Every minute Select a predefined Cron schedule, or type a custom Cron schedule (Quartz Cron) Target Sync: Local or UNC folder Sync File Local/UNC Path: c:\tmp pick... Select a local or UNC path where Sync Files are located; the path must already exist Dynamic Parameters... OK Cancel

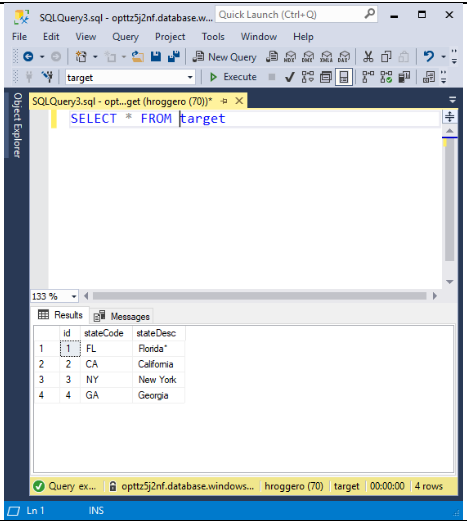
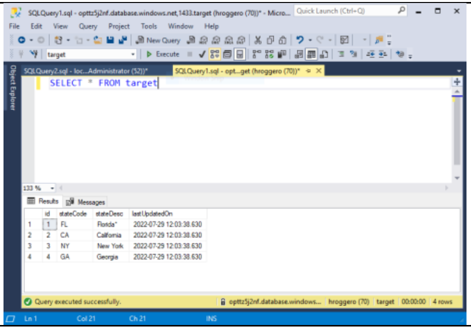
Select the Target System tab	Enter the following information: Target System: **targetdb** Schema Drifting: **Auto add and update columns** Target Table: **target** Click OK. The job will start immediately and will run every minute to automatically detect new/updated records. Note: A warning may come up indicating the target table doesn't exist; click Yes. The table will be created automatically.  Warning The target object does not seem to exist; the Sync Writer operation may fail. Proceed anyway? 	 New Sync Reader A sync reader fetches data from a source database table and saves the records into a Data Sync file that can later be used to push changes into another database. Job Name: sourcetable — A friendly job name (must be unique) Data Source Replication Settings PGP Encryption Advanced Settings Data Pipeline Target System Triggers Target System: targetdb Automatic Ingestion: ☑ Automatically push data changes in target table Schema Drifting: None Target Table: target Target table will be created automatically if not found Edit Initialization SQL Script... Edit Finalization SQL Script... Preview DDL Script... Dynamic Parameters... OK Cancel
Verify Sync Job completed	Once the job is completed, within a few seconds, you can inspect the output window of the job itself. Click on the job, and expand the output window at the bottom. At this point, the target table contains a copy of the source table, along with its records.	 DATAZEN MANAGER File Configuration Help RUNNING New... History... Disable Enable Filter: Hide Output Log Export Log... Output Level localhost\enzo -> enzostaging http://localhost:9559 Enterprise Edition (days left: 30) Name: sourcetable — Status: ready — System: SQLServer->SQLServer — Last Run On: 7/28/2022 8:27:14 ... — Schedule: 0 * * ? * * — Path: c:\tmp 7/28/2022 8:27:16 PM: Sync Reader completed. 3 changed, 0 deleted records. Total runtime (sec): 2.524327 7/28/2022 8:27:16 PM: Sync Writer completed. 3 changed records. Total runtime (sec): 1.586003 7/28/2022 8:27:15 PM: Source data extraction completed - upsert: 3 deleted: 0

Test Cross-Cloud Replication

Now that the replication topology is in place, let's test ongoing database replication between AWS and Azure.

Step	Description	Details

Add/Modify source data	Bring up SSMS, and make sure to select the Query window that is connected to the local SQL Server database.
	
	To add records to the **source** table, run this SQL batch:
	
	`INSERT INTO test.dbo.source`
	`VALUES ('GA', 'Georgia')`
	
	`UPDATE test.dbo.source`
	`SET stateDesc = 'Florida*'`
	`WHERE stateCode = 'FL'`
	
	Within a minute the changes will be detected and replicated directly to the Azure SQL Database table.
		
		SQLQuery1.sql - enzotest1\datazen.master (enzotest1\bscadmin (66))* - Microsoft SQL Server Managem...
		 File
		Edit
		View
		Query
		Project
		Tools
		Window
		Help
		New Query
		Execute
		master
		Object Explorer
		Connect
		Databases
		Security
		Server Objects
		Replication
		PolyBase
		Management
		XEvent Profiler
		Messages
		enzotest1\datazen (SQL Server 15.0.200
		`INSERT INTO test.dbo.source`
		`VALUES ('GA', 'Georgia')`
		`UPDATE test.dbo.source`
		`SET stateDesc = 'Florida*'`
		`WHERE stateCode = 'FL'`
		`(1 row affected)`
		`(1 row affected)`
		`Completion time: 2022-07-28T14:11:24.4013135+00:00`
		Query executed succe... — enzotest1\datazen (15.0 RTM) — enzotest1\bscadmin (66) — master — 00:00:00 — 0 rows
		Ready — Ln 3 — Col 1 — Ch 1 — INS

Verify the data in the target table	Switch the query window to the target database, and run this SQL command to visualize the records (note, it may take up to 1 minute before the changes take place). `SELECT * FROM target`	
Propagate Schema Changes	On the source database, run the following command; a new datetime field will be added: `ALTER TABLE test.dbo.source ADD lastUpdatedOn DATETIME NOT NULL DEFAULT(GETUTCDATE())`	
Inspect target table	A minute later, the new column will be added to the target table. Run this command on the target Azure SQL database: `SELECT * FROM target` NOTE: We explicitly selected the Schema Drifting option to 'Auto add and update columns'. This option automatically propagates new columns and changes the target data types.	
Delete a record	On the source database, run the following command to delete a record: `DELETE FROM test.dbo.source WHERE id=1`	
Inspect target table	A minute later, run this command on the target Azure SQL database: `SELECT * FROM target`	

```
SELECT * FROM target
```

	id	stateCode	stateDesc
1	1	FL	Florida*
2	2	CA	California
3	3	NY	New York
4	4	GA	Georgia

Query ex... | opttz5j2nf.database.windows... | hroggero (70) | target | 00:00:00 | 4 rows

```
SELECT * FROM target
```

	id	stateCode	stateDesc	lastUpdatedOn
1	1	FL	Florida*	2022-07-29 12:03:38.630
2	2	CA	California	2022-07-29 12:03:38.630
3	3	NY	New York	2022-07-29 12:03:38.630
4	4	GA	Georgia	2022-07-29 12:03:38.630

Query executed successfully. | opttz5j2nf.database.windows... | hroggero (70) | target | 00:00:00 | 4 rows

How to Restart the DataZen Agent

Under production deployments, the DataZen Agent runs as a Windows Service and restarts automatically upon reboots. However, in this lab, the agent runs as a console application. As a result, if the agent stops, or after a reboot, you will need to restart the agent manually.

To start the agent manually from the EC2 Instance, open Windows Explorer, navigate to **C:\Program Files (x86)\DataZen\EnzoDS9559**, and start **EnzoDS.exe** in administrative mode.

Further Considerations

This lab was designed to introduce you to the capabilities of DataZen replication and its automated Synthetic Change Data Capture capabilities. Several choices were made to simplify this lab so that the core capabilities could be easily demonstrated, including the ability to established a fully decoupled replication topology between cloud providers and database systems.

Additionally, the following considerations apply for more complex scenarios and production environments:

- **Cross Database Replication**
  The target system can be any SQL Server edition (including Azure SQL Database), any other database server (such as Oracle) or platform (including Azure Event Hub or ADLS Parquet files).

- **Native SQL Server CDC and Change Tracking Support**
  In this lab, we selected DataZen's automated Synthetic CDC capabilities; however, you can also select SQL Server CDC or Change Tracking. CDC and Change Tracking allow you to leverage SQL Server's built-in data capture to detect inserts, updates, and delete operations. This is the preferred option when dealing with tables that experience large data changes.

- **Decoupled Replication**
  We chose to install and configure a single VM (in AWS) to quickly demonstrate how to replicate data from a single server. However, in some instances where security is important, it may be necessary to install a second VM in Azure, and use a cloud drive to push the change log between AWS and Azure.

- **Multicast**
  Because DataZen uses Change Logs to capture changes, you can replay the same logs to any number of target systems, or stand up a new target system at any time in the future.

- **PGP Encryption**
  Change Logs can also be encrypted using PGP; this can be useful if you do not trust the intermediate storage system.

- **Any Source**
  We chose SQL Server Express as the source system for this lab; however, virtually any source system could be used, including other database platforms, messaging systems, files or HTTP data sources.

For more information about DataZen and its capabilities, please contact info@enzounified.com or visit https://www.enzounified.com